Administrator's Guide

WebtoB 6

Copyright

Company Information

TmaxSoft Co., Ltd.

TmaxTower 8-9F, 29, Hwangsaeul-ro 258 beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

Website: https://www.tmaxsoft.com/en/

Restricted Rights Legend

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

Trademarks

Open Source Software Notice

Document History

Product Version	Guide Version	Date	Remarks
WebtoB 6	3.1.2	2025-01-22	-

Contents

1. Getting Started

1.1. License Specific Functions

WebtoB provides the following functions according to the license type.

License Type	Description
TRIAL	Trial license. Any WebtoB installed using the installer contains a default trial license. Supports a single HTH, and a maximum of 5 users.
BASE	For use as embedded in JEUS. Only supports a single HTH. In a UNIX/Linux environment, JSVPort is not used since only named pipes can be used to connect to JEUS.
STANDARD	One or more HTHs are supported. Can connect to JEUS without limitations and supports most WebtoB functions.
ENTERPRISE	Supports the following functions in addition to all of those provided in the STANDARD version. ◦ Supports JSP/Servlet engine of the embedded engine. Only a single JEUS server (DAS) can be used in this case. ◦ Supports Reverse Proxy Group function for multi-configuration when connecting to another web application server through reverse proxy. ◦ Supports HTTP Methods (MKCOL, COPY, MOVE, and PORPFIND) for WebDAV. ◦ Supports FILTERS process for filter processing. Notably, SiteMinder Filter (wbSmISAPI) support for CA SSO. ◦ Provides WebAdmin.

1.2. Quick Start

The following are simple steps to quickly verify WebtoB's normal operation of after installation:

1. Use **configValidator**, WebtoB's configuration file validation tool, to verify that the configuration file is properly set up.

```
$ configValidator
```

The following is an example of configuration of the WEBTOB6_HOME_PATH\config\webtob-config.json file.

```
{
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads": 8,
    "hth_schedule": "RR"
  },
  "server": {
    "http": {
      "common_config": {
        "doc_root": "docs",
        "service_order": "uri,ext",
        "access_log": "access_log1"
      },
      "http_servers": [
        {
          "name": "http1",
          "port": 8080,
          "enable_ssl": true,
          "ssl_name": [
            "ssl1"
          ]
        }
      ]
    },
    "wjp": {
      "port": 9900,
      "wjp_servers": [
        {
          "name": "MyGroup1",
          "svr_chk_time": 60
        }
      ]
    }
  },
  "logging": {
    "access_log": [
      {
        "name": "access_log1",
        "level": "INFO",
        "format": "DEFAULT",
        "handlers": {
          "file_handler": {
            "file_name": "logs/access.log"
          },
          "enable_console_handler": false
        }
      }
    ],
    "error_log": {
      "level": "INFO",
      "handlers": {
        "file_handler": {
          "file_name": "logs/errors.log"
        },
        "enable_console_handler": false
      }
    },
    "system_log": [
      {
        "name": "webtob",
        "level": "INFO",
        "handlers": {
          "file_handler": {
            "file_name": "logs/webtob.log"
          },
          "enable_console_handler": true
        }
      }
    ]
  },
  "destination": {
    "jeus": [
      {
        "name": "MyGroup1",
        "server_schedule": "RR",
        "connection_schedule": "RR"
      }
    ],
    "reverse_proxy": {
      "reverse_proxy_group": [
      {
        "name": "rproxyGroup1",
        "reverse_proxy_server": [
          {
            "address": "internal.server:80"
          }
        ]
      }
    ]},
    "htmls": [
      {
        "name": "htmls1"
      }
    ]
  },
  "service": {
    "uri": [
      {
        "name": "uri1",
        "target_http_servers": [
          "http1"
        ],
        "match": {
          "type": "prefix",
          "target": "/rproxy",
          "rewrite": "/"
        },
        "destination": {
          "type": "REVERSE_PROXY",
          "target": "rproxyGroup1"
        }
      }
    ],
    "ext": [
      {
        "name": "ext1",
        "target_http_servers": [
          "*"
        ],
        "match": {
          "type": "exact",
          "target": "text/html"
        },
        "destination": {
          "type": "JEUS",
          "target": "MyGroup1"
        }
      }
    ]
  },
  "ssl": {
    "ssl_configs": [
      {
        "name": "ssl1",
        "webtob_certificate_file": "server.crt",
        "webtob_certificate_key_file": "key.crt"
      }
    ]
  }
}
```

2. Enter the command to start WebtoB in the command prompt.

```
$ wsboot
```

After starting WebtoB, you can check the names of the running threads using the 'ps -p [WebtoB 6 PID] -L' or 'top -H -p [WebtoB 6 PID]' command.

The following describes the roles of each thread.

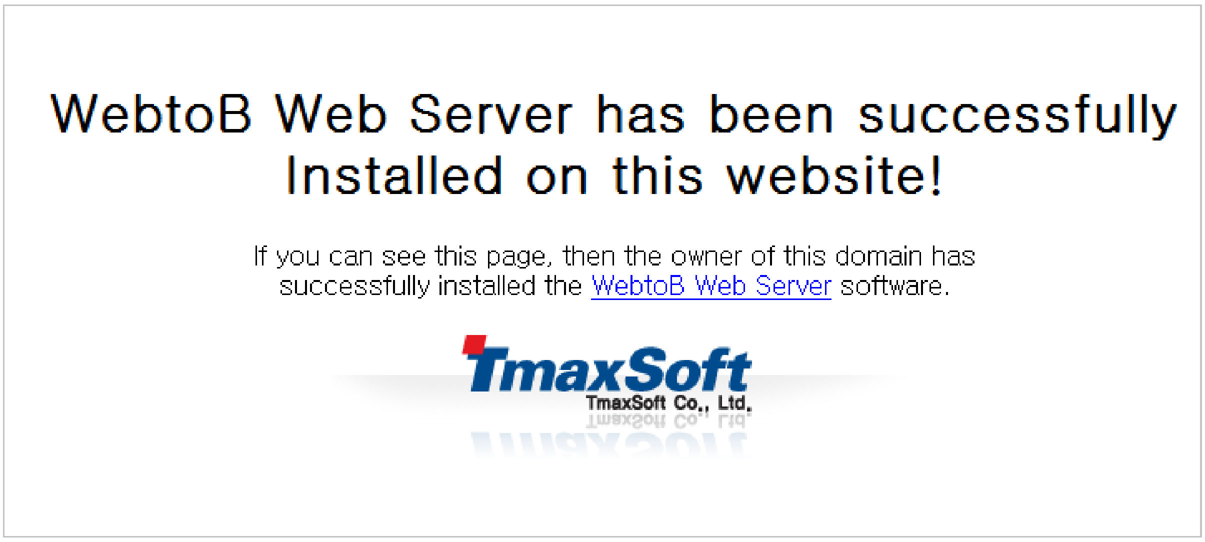
Thread	Role
BOOT	The main thread that runs when WebtoB starts. It handles the overall server booting process, including loading configurations, creating and starting the server, etc.
ADMIN	Admin server thread. It performs WebtoB status monitoring or commands according to the client’s REST API request.
HTL	The thread that accepts client connections.
HTH	The thread that handles client requests. It performs parsing of requests/responses, network I/O, and delegates specific tasks to the WORKER thread.

Thread	Role
WORKER	The thread that processes commands from the HTH thread. Each WORKER is created and assigned by a specific HTH to handle tasks such as static content (HTMLS) processing. (To be expanded in the future)
LOGGING	The thread responsible for handling all log messages output from WebtoB.

3. Open a browser and enter the following URL.

```
http://<IP Address>:<8080 or a user-specified port number>/
```

If WebtoB starts without any error, the following page appears.



WebtoB Web Server has been successfully Installed on this website!

If you can see this page, then the owner of this domain has successfully installed the WebtoB Web Server software.

4. Enter the WebtoB shutdown command in the command prompt.

```
$ wsdown
```

2. Configuration

2.1. Basic Structure

WebtoB environment configuration is set for the operation of a WebtoB node and the web server.

WebtoB’s environment configuration file supports YAML and JSON formats. It consists of 'section name', 'item name', and 'setting value'.

The following is the configuration format in JSON.

```
"Section Name": {
  "Item" : Setting Value,
  ...,
  "Item" : Setting Value
}
```

The start of each section is followed by the section name and then configuration items.

Configuration must be set according to the rules of JSON or YAML format, depending on the chosen format.

The following is an example of configuration in JSON format.

```
{
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads": 8
  },
  "server": {
    "http": {
      "common_config": {
        "doc_root": "docs",
        "access_log": "access_log1"
      },
      "http_servers": [
        {
          "name": "http1",
          "port": 8080,
          "enable_ssl": true,
          "ssl_name": [
            "ssl1"
          ]
        }
      ]
    },
    "wjp": {
      "port": 9900,
      "wjp_servers": [
        {
          "name": "MyGroup1"
        }
      ]
    }
  },
  "logging": {
    "access_log": [
      {
        "name": "access_log1",
        "level": "INFO",
        "format": "DEFAULT",
        "handlers": {
          "file_handler": {
            "file_name": "logs/access.log"
          }
        }
      }
    ],
    "error_log": {
      "level": "INFO",
      "handlers": {
        "file_handler": {
          "file_name": "logs/errors.log"
        }
      }
    },
    "system_log": [
      {
        "name": "webtob",
        "level": "INFO",
        "handlers": {
          "file_handler": {
            "file_name": "logs/webtob.log"
        }
      }
    ]
  },
  "destination": {
    "jeus": [
      {
        "name": "MyGroup1"
      }
    ],
    "reverse_proxy": {
      "reverse_proxy_group": [
      {
        "name": "rproxyg1",
        "reverse_proxy_server": [
          {
            "address": "internal.server:80",
            "name": "rproxy1"
          }
        ]
      }
    ]},
    "htmls": [
      {
        "name": "htmls1"
      }
    ]
  },
  "service": {
    "uri": [
      {
        "name": "uri1",
        "target_http_servers": [
          "http1"
        ],
        "match": {
          "type": "prefix",
          "target": "/rproxy",
          "rewrite": "/"
        },
        "destination": {
          "type": "REVERSE_PROXY",
          "target": "rproxyGroup1"
        }
      }
    ],
    "ext": [
      {
        "name": "ext1",
        "target_http_servers": [
          "*"
        ],
        "match": {
          "type": "exact",
          "target": "text/html"
        },
        "destination": {
          "type": "JEUS",
          "target": "MyGroup1"
        }
      }
    ]
  },
  "ssl": {
    "ssl_configs": [
      {
        "name": "ssl1",
        "certificate_file": "server.crt",
        "certificate_key_file": "key.crt"
      }
    ]
  }
}
```

2.1.1. Types of Configurable Sections

The following describes each section that can be configured in WebtoB.

Section	Description
ACCESS	Client access control based on IP address, network/netmask, and header information.
ALIAS	Alias for the physical server path and URI.
DESTINATION	The internal server that will actually handle client requests.
ERRORDOCUMENT	Error response actions.
FILTER	Section for using the Filter module.
HEADERS	Settings for controlling HTTP request and response headers.
LOGGING	Logging format for client requests.
NODE	Node settings.
SERVER	Settings for the information that clients can access.
SERVICE	Settings for internal server to handle HTTP requests based on the request's URI and EXT.
SSL	SSL to use on WebtoB.

The DESTINATION, SERVER, and SERVICE sections are required.

2.1.2. Types of Setting Values and Configuration Methods

The values of each item are set in one of the following types: object, integer, string, boolean, or array. The following describes each value type.

Type	Description
object	Used when there are multiple child settings. ◦ JSON: Enclosed with curly braces ({ }) ◦ YAML: Indent one level after a line break
integer	Numeric values.
string	String values.
boolean	Sets to either true or false.
array	Sets multiple settings of the same type. Enclose with square brackets ([ ]) and separate each setting with a comma (,), or use hyphen (-) after a line break.

To fully understand the configuration items described in this guide, you must first be familiar with the basic rules as follows:

- An item starting with the hash symbol (#) before its name is an optional item.
- If a format is enclosed in square brackets ([ ]), it indicates an array format

composed of specific types.

- If a default value exists, specify the value after the hash symbol.
- For a integer variable, use parentheses to display the valid range.
- If the maximum limit does not exist, omit the range. The maximum value must be less than or equal to INT_MAX (2147483647).
- "$ENV" means the environment variable can be referenced.
- "R.PATH" means that a relative path, if specified, is relative to $WEBTOB6_HOME_PATH.
- For object format, internal settings are expressed in curly brackets ({ }).
- For object format, if the same format exists in a parent setting, the description is omitted with "{...}".
- "COMMON" indicates that if not specified, it follows the parent setting.

2.2. ACCESS Section

The ACCESS section configures client accesses based on IP address, network/netmask, header information, and the order of applying configurations.

This section can be applied to the SERVICE and SERVER sections. The defined resources from each of these sections are accepted or denied.

2.2.1. Configuration Items

The following is the configuration format of the ACCESS section.

```
#"access": {
    "access_list": [
        {
            "name": string,
            "policy": string,
            #"allow_network": [string],
            #"deny_network": [string],
            #"allow_header": [string],
            #"deny_header": [string],
            #"method_whitelist": [string],
            #"method_blacklist": [string]
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the ACCESS section configuration items.

2.2.1.1. access_list

A list of access control configuration objects for the ACCESS section.

Item	Description
Data Type	Array (object)

2.2.1.2. access_list/name (Required)

Sets the ACCESS section name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.2.1.3. access_list/policy (Required)

Sets the order of application for method, allow_network, allow_header, deny_network, and deny_header.

Item	Description
Data Type	String
Values	"blacklist" | "whitelist" | "method_only"

2.2.1.4. access_list/allow_network

Sets the IP addresses or networks/netmasks that are allowed for requests. The special value 'Allow = "all"' means all IP addresses.

Item	Description
Data Type	Array (string)
Range	Up to 256 items (within 255 characters)

2.2.1.5. access_list/deny_network

Sets the IP addresses or networks/netmasks that are denied for requests.

Item	Description
Data Type	Array (string)
Range	Up to 256 items (within 255 characters)

2.2.1.6. access_list/allow_header

Sets the headers that are allowed in the request. The value of the <header field name> in the request matches the pattern of <regular expression>.

Item	Description
Data Type	Array (string)
Range	Up to 256 items (within 255 characters)
Format	<header field name> <regular expression>

2.2.1.7. access_list/deny_header

Sets the headers that are denied in the request. The value of the <header field name> in the request matches the pattern of <regular expression>.

Item	Description
Data Type	Array (string)
Range	Up to 256 items (within 255 characters)
Format	<header field name> <regular expression>

2.2.1.8. access_list/method_whitelist

Sets the HTTP method to use. However, this cannot be set simultaneously with method_blacklist.

Item	Description
Data Type	Array (string)
Values	"GET", "POST", "PUT", "HEAD", "DELETE", "CONNECT", "OPTIONS", "TRACE", "PATCH", "PROPFIND", "PROPPATCH", "MKCOL", "COPY", "MOVE", "LOCK", "UNLOCK"

2.2.1.9. access_list/method_blacklist

Sets HTTP methods to be excluded. However, this cannot be set simultaneously with method_whitelist.

Item	Description
Data Type	Array (string)
Values	"GET", "POST", "PUT", "HEAD", "DELETE", "CONNECT", "OPTIONS", "TRACE", "PATCH", "PROPFIND", "PROPPATCH", "MKCOL", "COPY", "MOVE", "LOCK", "UNLOCK"

2.2.2. Example

The following is an example configuration of the ACCESS section.

```
{
    "access": {
        "access_list": [
            {
                "name": "access1",
                "policy": "blacklist",
                "deny_network": [ "192.168.1.43/255.255.255.0" ]
            }
        ]
    }
}
```

2.3. ALIAS Section

The ALIAS section maps a URI to a physical directory path on the server regardless of the document root. Aliasing can map a directory regardless of the Document Root, which is convenient from a management perspective.

2.3.1. Configuration Items

The following is the environment configuration format of the ALIAS section.

```
#"alias": {
    #"alias_list": [
        {
            "name": string,
            "url": string,
            "real_path": string
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the ALIAS section configuration items.

2.3.1.1. alias_list

A list of ALIAS configuration.

Item	Description
Data Type	Array (object)

2.3.1.2. alias_list/name (Required)

The name of the ALIAS configuration. This 'name' must be set when using the functionality of the ALIAS section in other sections.

Item	Description
Data Type	String
Range	Up to 31 characters

2.3.1.3. alias_list/url (Required)

Sets the URI to be specified as an alias. Matching is checked by comparing the request URL with the configured URI, starting from the first byte, up to the length of the URI.

Item	Description
Data Type	String
Range	Up to 255 characters
Format	Must start and end with a slash (/).

2.3.1.4. alias_list/real_path (Required)

Sets the physical directory path on the server. Relative paths (paths that do not start with '/') are automatically replaced with "$WEBTOB6_HOME_PATH/relative path".

Item	Description
Data Type	String
Range	Up to 255 characters
Format	Must end with a slash (/).

2.3.2. Example

The following is an example configuration of the ALIAS section.

```
{
    "alias": {
        "alias_list": [
            {
                "name" : "alias1",
                "url": "/external/path/",
                "real_path": "/internal/real/path/"
            }
        ]
    }
}
```

2.4. DESTINATION Section

The DESINTATION section configures the backend server to be used for processing requests and receiving responses. It must include one of **'jeus'**, **'reverse proxy'**, or **'htmls'**.

2.4.1. Configuration Items

The following is the environment configuration format of the DESTINATION section.

```
"destination": {
    #"jeus": [
        {
            "name": string,
            #"server_schedule": string,                                 # "RR"
            #"connection_schedule": string,                             # "RR"
            #"enable_flexible_sticky_session_routing": boolean,         # false
            #"enable_request_level_ping": boolean,                      # false
            #"request_level_ping_timeout": integer                      # 3 (0-INT_MAX)
            #"request_level_ping_retry_count": integer                  # 0 (0-INT_MAX)
            #"headers": [string],
            #"session_id_cookie_key": string,                           # "JSESSIONID"
            #"rewrite_cookie_domain": string,
            #"rewrite_cookie_path": {
                "from": string,
                "to": string
            },
            #"enable_cache": boolean,                                   # false
            #"cache_refresh": integer,                                  # 3600 (1-INT_MAX)
            #"max_queue_count": integer,                                # 0 (0-INT_MAX)
            #"max_queue_url": string,
            #"max_queue_status_code": integer,                          # 503
            #"queue_timeout": integer,                                  # 0 (0-INT_MAX)
            #"backup_server": string
        }
    ],
    #"reverse_proxy": {
        #"common_config": {                                             # COMMON
            #"headers": [string],
            #"compression": [string],
            #"compression_min_size": integer,                           # 0 (0-INT_MAX)
            #"rewrite_redirect": [
                {
                    "original_uri": string,
                    "redirect_path": string
                }
            ],
            #"rewrite_cookie_domain": string,
            #"rewrite_cookie_path": {
                "from": string,
                "to": string,
            },
            #"max_queue_count": integer,                                # 0 (0-INT_MAX)
            #"max_queue_url": string,
            #"max_queue_status_code": integer,                          # 503
            #"websocket_connections_max": integer,                      # 0 (0-INT_MAX)
            #"websocket_session_timeout": integer,                      # 0 (0-INT_MAX)
            #"set_host_header": string,
            #"queue_timeout": integer,                                  # 0 (0-INT_MAX)
            #"name_resolution_interval": integer,                       # 0 (0-INT_MAX)
            #"server_health_check": {
                #"retry_count": integer,                                # 3 (0-INT_MAX)
                #"failback_interval": integer,                          # 60 (0-INT_MAX)
                #"http_failure": [string],                              # []
                #"connection_timeout": integer,                         # 5 (0-INT_MAX)
                #"enable_name_resolution_on_fail": boolean              # true
            },
            #"rewrite_html_url": [
                {
                    "tag": string,
                    "attribute": [string],
                    "from": string,
                    "to": string
                }
            ],
            #"rewrite_html_max_size": integer                           # 10240 (1-INT_MAX)
        },
        "reverse_proxy_group": [
            {
                "name": string,
                "reverse_proxy_server": [
                    {
                        "name": string,
                        "address": string,
                        #"enable_proxy_ssl": boolean,                   # false
                        #"proxy_ssl_name": [string],
                        #"persistent_server_connections_min": integer,  # 0 (0-INT_MAX)
                        #"persistent_server_connections_max": integer,  # 0 (0-INT_MAX)
                        #"persistent_server_check_time": integer,       # 30 (0-INT_MAX)
                        #"persistent_server_check_url": string,
                        #"persistent_server_timeout": integer,          # 300 (0-INT_MAX)
                        #"enable_cache": boolean,                       # false
                        #"cache_refresh": integer,                      # 3600 (1-INT_MAX)
                        #"sticky_session_routing_id": string,
                        #"is_backup_server": boolean,                   # false
                        #"load_balancing_factor": integer,              # 1 (1-INT_MAX)
                        #"common_config": {...}                         # COMMON
                    }
                ],
                #"server_schedule": string,                             # "RR"
                #"sticky_session_routing": {
                    #"policy": string,                                  # "UseOriginalCookie"
                    #"session_id_cookie_key": string,                   # "JSESSIONID"
                    #"enable_flexible_sticky_session_routing": boolean  # false
                },
                #"common_config": {...}                                 # COMMON
            }
        ]
    },
    #"htmls":[
        {
            "name": string,
            #"headers": [string],
            #"compression": [string],
            #"compression_min_size": integer,                           # 0 (0-INT_MAX)
            #"enable_cache": boolean,                                 # false
            #"cache_refresh": integer,                                # 3600 (1-INT_MAX)
            #"queue_timeout": integer,                                # 0 (0-INT_MAX)
            #"enable_sendfile": boolean,                              # false
            #"sendfile_min_size": integer,                            # 0 (0-INT_MAX)
            #"enable_etag": boolean                                   # true
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the DESTINATION section configuration items.

2.4.1.1. jeus

Set this when performing services using WJP by integrating WebtoB with JEUS.

Item	Description
Data Type	Array (object)

2.4.1.2. jeus/name (Required)

Sets the name of JEUS. The specified name must match the name set in wjp/wjp_servers/name.

Item	Description
Data Type	String
Range	Up to 31 characters

2.4.1.3. jeus/server_schedule

Sets the method for specifying a server to process requests when multiple servers are defined in wjp/wjp_servers.

Item	Description
Data Type	String
Range	"RR"
Default Value	"RR"

The following describes each configuration value.

Value	Description
RR	Assigns requests sequentially to JEUS using the round robin method.

2.4.1.4. jeus/connection_schedule

Sets the method for specifying a connection to process requests when multiple connections are established to the JEUS server.

Item	Description
Data Type	String
Range	"RR"
Default Value	"RR"

The following describes each configuration value.

Value	Description
RR	Assigns requests to connections sequentially using the round robin method.

2.4.1.5. jeus/enable_flexible_sticky_session_routing

Sets whether to enable flexible routing during session routing.

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
True	If connections up to persistent_server_connections_max of the internal server are in the RUN state when using the sticky_session_routing_id for session routing, requests are routed to another internal server connection with a different sticky_session_routing_id instead of being queued.
False	Uses the sticky session routing by default. Requests are routed only to internal server connections with the same sticky_session_routing_id, and requests are queued if connections up to persistent_server_connections_max are all in the RUN state.

If you are using flexible routing, requests from different clients with the same JSESSIONID may be routed to different servers. (**Default setting is recommended.**)

2.4.1.6. jeus/enable_request_level_ping

Sets whether to send a PING to the server before forwarding a request. This setting is useful when multiple JEUS servers are connected and a server may have difficulty responding to requests due to Out Of Memory (OOM).

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
True	Sends a PING for every request forwarded to the JEUS server and only forwards the request if a response is received.
False	Directly forwards the request without sending a PING for all requests forwarded to the JEUS server.

This may result in performance degradation as it requires sending a PING and waiting for a response for every request. (**Default setting is recommended.**)

2.4.1.7. jeus/request_level_ping_timeout

Sets the time to wait for a response after sending a PING. If no response is received within the specified time, the connection is terminated, and another JEUS server is rescheduled to send a PING.

Item	Description
Data Type	Integer
Unit	seconds
Range	0 ~ INT_MAX
Default Value	3

2.4.1.8. jeus/request_level_ping_retry_count

Sets the number of retry attempts if sending a PING to all Jengines within the JEUS server fails.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.9. jeus/headers

Sets the name of the HEADERS section to apply.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

2.4.1.10. jeus/session_id_cookie_key

Sets the Key name of the HTTP cookie used for session routing.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"JSESSIONID"

2.4.1.11. jeus/rewrite_cookie_domain

Modifies the "domain=" value of the Cookie header field of the response from the internal server. If the domain string specified in "domain=" matches the string specified in RewriteCookieDomain, the "domain=" value is replaced with the domain of the user request.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"JSESSIONID"

Using the following configuration, if the WebtoB domain is webtob, the cookie of the original response "jsessionid=abc, domain=internal.server.com" is changed to "jsessionid=abc, domain=webtob".

```
"rewrite_cookie_domain": "internal.server.com"
```

2.4.1.12. jeus/rewrite_cookie_path

Modifies the "path=" value of the cookie header field of the internal server response.

Item	Description
Data Type	Object

2.4.1.13. jeus/rewrite_cookie_path/from

Specifies the string to be modified in the "path=" value.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.14. jeus/rewrite_cookie_path/to

Specifies the string to replace the "path=" value.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.15. jeus/enable_cache

Sets whether the content is cached.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.16. jeus/cache_refresh

Sets the value for calculating the validity period of cached responses when caching responses received from JEUS.

When WebtoB responds to a client’s request with cached content, it checks whether the cached content is valid. The validity is determined based on the set value (valid time).

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	3600

Item	Description
Priority	The priority of the setting is as follows: 1. Cache-Control:max-age: If the Cache-Control:max-age value is in the Response Header when the header is being cached, the header is specified to be valid for the max-age value (second). 2. Expires: If the Expires value is in the Response Header when the header is being cached, the header is specified to be valid until the Expires value (hour). 3. cache_refresh: If there are no 'Cache-Control:max-age' or 'Expires' values in the response header, the response received from JEUS is valid for cache_refresh (in seconds) after being cached.

2.4.1.17. jeus/max_queue_count

When a surge in client requests overloads a server and the server is no longer able to respond to new requests, it is necessary to ignore them.

Once the number of client requests in the queue reaches a certain limit, any further incoming requests will not be queued. Instead, the server will respond to the client with an error.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.18. jeus/max_queue_url

Sets the page to be served instead when the server queue is full.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.19. jeus/max_queue_status_code

Sets the status code to be returned when the server queue is full.

Item	Description
Data Type	String
Range	301 | 302 | 303 | 307 | 308 | 410 | 503
Default Value	503

2.4.1.20. jeus/queue_timeout

Specifies the timeout for user requests in a server queue.

When a request cannot be processed due to a large volume of user requests, the request waits in the server queue until a server process becomes available.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	There is no time limit for waiting in the server queue.
Positive Integer	Requests waiting longer than the specified time will be removed from the queue and respond with "503 Service Unavailable".

The following is an example of setting max_queue:

```
...
"destination": {
  "jeus": {
    "max_queue_count": 3,
    "max_queue_url":"/jsvtest/common/test.html",
    "max_queue_status_code":302,
    "queue_timeout":100
  }
},
...
```

With the above settings, when there are 2 JEUS connections and 10 calls are made, the response will include 2 page calls, 3 queue page calls, and 5 calls to the max_queue_url.

2.4.1.21. jeus/backup_server

Sets the server to be used as a backup when all servers are not ready.

Item	Description
Data Type	String
Range	Up to 31 characters

2.4.1.22. reverse_proxy

This is a type of HTTP proxy used to forward requests to internal servers and can be used for web application server (WAS) integration.

Item	Description
Data Type	Object

2.4.1.23. reverse_proxy/common_config

A common setting for reverse proxies. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. reverse_proxy_server 2. reverse_proxy_group 3. reverse_proxy

2.4.1.24. reverse_proxy/common_config/headers

Sets the name of the HEADERS section to apply.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

2.4.1.25. reverse_proxy/common_config/compression

Sets the target for response compression. Specifies the MIME-type (content-type of the response) to be compressed. The target response is compressed with GZIP, then sent to the client.

Item	Description
Data Type	Array (string)
Range	Up to 32 items (within 255 characters)

> Compression can reduce the network traffic, but it may degrade server performance.
>
> Files with low compression ratio, such as zip or jpeg, should not be compressed if

possible to avoid server overhead.

Compression can only be used for requests with Accept-Encoding set to GZIP or deflate in the HTTP request header.

2.4.1.26. reverse_proxy/common_config/compression_min_size

Specifies the minimum size of the response to compress.

If the value of the Content-Length response header is greater than the specified size, the response is compressed. However, this does not apply to chunked responses, as it is difficult to determine the size of the response.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.27. reverse_proxy/common_config/rewrite_redirect

Modifies the values in the "Location" and "Content-Location" headers in the response from the internal server. If that field starts with "original_uri", replace the part that matches "original_uri" with "redirect_path".

Item	Description
Data Type	Array (object)
Range	Up to 16 items (each original_uri and redirect_path must be within 255 characters)

In the following example, if the Location of the original response is "http://internal.server.com:80/docs_kr/abc.html", it will be changed to "/internal_kr/abc.html".

```
"rewrite_redirect": [
    {
        "original_uri": "http://internal.server.com:80/docs/",
        "redirect_path": "/internal/"
    },
    {
        "original_uri": "http://internal.server.com:80/docs_kr",
        "redirect_path": "/internal_kr/"
    },
    {
        "original_uri": "http://internal.server.com:80/docs_ch/",
        "redirect_path": "/internal_ch/"
    }
]
```

2.4.1.28. reverse_proxy/common_config/rewrite_cookie_domain

Modifies the "domain=" value of the Cookie header field of the response from the internal server. If the domain string specified in "domain=" matches the string specified in rewrite_cookie_domain, the "domain=" value is replaced with the domain of the user request.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	0

Using the following configuration, if the WebtoB domain is webtob, the cookie of the original response "jsessionid=abc, domain=internal.server.com" is changed to "jsessionid=abc, domain=webtob".

```
"rewrite_cookie_domain": "internal.server.com"
```

2.4.1.29. reverse_proxy/common_config/rewrite_cookie_path

Modifies the "path=" value of the cookie header field of the internal server response.

Item	Description
Data Type	Object

Using the following configuration, the original response cookie "jsessionid=abc, path=/jeus/application" is changed to "jsessionid=abc, path=/jeus_proxy/application".

```
"rewrite_cookie_path": {
    "from": "jeus",
    "to": "/jeus_proxy"
}
```

2.4.1.30. reverse_proxy/common_config/rewrite_cookie_path/from (Required)

Specifies the string to be modified in the "path=" value.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.31. reverse_proxy/common_config/rewrite_cookie_path/to (Required)

Specifies the string to replace the "path=" value.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.32. reverse_proxy/common_config/max_queue_count

When a surge in client requests overloads a server and the server is no longer able to respond to new requests, it is necessary to ignore them.

Once the number of client requests in the queue reaches a certain limit, any further incoming requests will not be queued. Instead, the server will respond to the client with an error.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.33. reverse_proxy/common_config/max_queue_url

Sets the page to be served instead when the server queue is full.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.34. reverse_proxy/common_config/max_queue_status_code

Sets the status code to be returned when the server queue is full.

Item	Description
Data Type	Integer
Range	301 | 302 | 303 | 307 | 308 | 410 | 503
Default Value	503

2.4.1.35. reverse_proxy/common_config/websocket_connections_max

Specifies the maximum number of connections when connection is upgraded from HTTP to WebSocket protocol.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	No limit is set on the number of WebSocket connections.

2.4.1.36. reverse_proxy/common_config/websocket_session_timeout

Specifies the timeout for WebSocket connection.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	Timeout check is not performed.

It is recommended to set this value based on the timeout setting of the WebSocket session on the internal server.

2.4.1.37. reverse_proxy/common_config/set_host_header

Specifies the Host header when forwarding the request to the internal server using reverse proxy.

If not set, the value specified in ServerAddress is used. If set to "$BypassHostHeader", the client Host header is used without modification.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	0

2.4.1.38. reverse_proxy/common_config/queue_timeout

Specifies the timeout for user requests in a server queue.

When a request cannot be processed due to a large volume of user requests, the request waits in the server queue until a server process becomes available.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	There is no time limit for waiting in the server queue.
Positive Integer	Requests waiting longer than the specified time will be removed from the queue and respond with "503 Service Unavailable".

2.4.1.39. reverse_proxy/common_config/name_resolution_interval

Sets the frequency at which Hostname Resolution is performed.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	Follows the operating system’s Time-To-Live (TTL) settings.

2.4.1.40. reverse_proxy/common_config/server_health_check

Configures health check information for reverse proxy groups.

Item	Description
Data Type	Object

The following is an example of using the server_health_check:

```
  "destination": {
    "reverse_proxy ": {
      "common_config": {
        "server_health_check":{
            "retry_count":4,
            "failback_interval":5,
            "http_failure":["http_invalid", "http_4xx"],
            "connection_timeout": 5,
            "enable_name_resolution_on_fail":false
        }
```

2.4.1.41. reverse_proxy/common_config/server_health_check/retry_count

Sets the number of retry attempts to determine failover.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	3

2.4.1.42. reverse_proxy/common_config/server_health_check/failback_interval

Sets the health check interval to attempt a failback after a failover.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	60

The following describes each configuration value.

Value	Description
0	Does not perform failover or failback.

2.4.1.43. reverse_proxy/common_config/server_health_check/http_failure

Sets the criteria for determining HTTP response failures.

Item	Description
Data Type	Array (string)

Item	Description
Allowed Options	The following options can treat certain types of HTTP responses as failures: ◦ http_invalid: Considers an invalid HTTP message as a failure. ◦ http_4xx: Considers 400 HTTP responses as failures. ◦ http_5xx: Considers 500 HTTP responses as failures. ◦ http_{3-digit number}: Considers HTTP responses corresponding to the specified {3-digit number} as failures. Allowed numbers are 403, 404, 429, 500, 502, 503, and 504.

2.4.1.44. reverse_proxy/common_config/server_health_check/connection_timeout

Specifies the retry interval for TCP connection with the internal server.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX (seconds)
Default Value	5 (seconds)

2.4.1.45. reverse_proxy/common_config/server_health_check/enable_name_resolution_on_fail

Sets whether to attempt Hostname Resolution again when a failure is determined.

Item	Description
Data Type	Boolean
Default Value	True

2.4.1.46. reverse_proxy/common_config/rewrite_html_url

Used to replace the URL included in the HTML page response body. If the URL includes a host, the host is replaced with the WebtoB server address used in the request.

Item	Description
Data Type	Array (object)
Range	Up to 64 items

The following is an example of changing the URL from http://test2:80 to /proxy/ for the src, longdesc, and usemap attributes of the img tag.

```
"rewrite_html_url": [
    {
        "tag": "img",
        "attribute": ["src", "longdesc"],
        "from": "http://test2:80",
        "to": "/proxy/"
    }
]
```

2.4.1.47. reverse_proxy/common_config/rewrite_html_url/tag (Required)

Specifies the tag name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.48. reverse_proxy/common_config/rewrite_html_url/attribute (Required)

Specifies the attribute name.

Item	Description
Data Type	Array (string)
Range	Up to 64 items (within 255 characters)

2.4.1.49. reverse_proxy/common_config/rewrite_html_url/from (Required)

Specifies the string to be modified in the URL.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.50. reverse_proxy/common_config/rewrite_html_url/to (Required)

Specifies the string to which the URL will be modified.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.51. reverse_proxy/common_config/rewrite_html_max_size

If the response is an HTML page (Content-Type: text/html), specific tag values inside the page can be modified. This value sets the maximum internal buffer size that can be used. If the response size is greater than the specified value, the original response is sent to the client.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	10240

2.4.1.52. reverse_proxy/reverse_proxy_group (Required)

You can manage the reverse_proxy settings as a group and configure multiple servers.

By setting up multiple reverse proxy (internal) servers, you can group them for load balancing and sticky session routing, and integrate them with Web Application Server (WAS).

Item	Description
Data Type	Array (object)

2.4.1.53. reverse_proxy/reverse_proxy_group/name (Required)

Specifies the reverse_proxy_group name.

Item	Description
Data Type	String
Range	Up to 31 characters

2.4.1.54. reverse_proxy/reverse_proxy_group/reverse_proxy_server (Required)

You can manage different configurations for each ip:port in one reverse_proxy_group.

Item	Description
Data Type	Array (object)
Range	Up to 32 items

2.4.1.55. reverse_proxy/reverse_proxy_group/reverse_proxy_server/name (Required)

Specifies the name of the reverse proxy server.

Item	Description
Data Type	String
Range	Up to 31 characters

2.4.1.56. reverse_proxy/reverse_proxy_group/reverse_proxy_server/address (Required)

Specifies the internal server address to which the request is sent.

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.57. reverse_proxy/reverse_proxy_group/reverse_proxy_server/enable_proxy_ssl

Specifies whether to use SSL/TLS to connect to the internal server. If set, the PROXY_SSL section item can be set in proxy_ssl_name.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.58. reverse_proxy/reverse_proxy_group/reverse_proxy_server/proxy_ssl_name

Specifies the name of the ssl/proxy_ssl_configs section to be used. This setting is applied only when enable_proxy_ssl is set to 'true'.

Item	Description
Data Type	Array (string)
Range	Up to 100 items (within 31 characters)

The following is an example of using proxy_ssl_name:

```
"destination": {
  "reverse_proxy":{
    "reverse_proxy_server":[
      {
        "address":"internal.server.com:80",
        "name":"rproxy1",
        "enable_proxy_ssl": true,
        "proxy_ssl_name":["ssl1"]
      }
    ]
  }
}
```

2.4.1.59. reverse_proxy/reverse_proxy_group/reverse_proxy_server/persistent_server_connections_min

Specifies the minimum number of connections required to maintain connection with an internal server after request processing is complete.

A new connection with an internal server is created when there is a new request, and as long as the internal server doesn’t terminate the connection it is maintained and reused for subsequent requests.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.60. reverse_proxy/reverse_proxy_group/reverse_proxy_server/persistent_server_connections_max

Specifies the maximum number of connections required to maintain connection with an internal server after request processing is complete.

When there is a new request, a new connection is created if all connections are currently processing other requests and the request is queued if the number of connections has reached the specified maximum limit.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.4.1.61. reverse_proxy/reverse_proxy_group/reverse_proxy_server/persistent_server_check_url

Specifies to internally use the HTTP HEAD request as the ping message for maintaining internal server connections.

Item	Description
Data Type	String
Range	Up to 255 characters

Item	Description
Default Value	"/"

Connections are maintained by using a ping message to check for the connection state after sending an HTTP request to the internal server and receiving an HTTP response. If the response message is not 200, the connection is terminated. An application must be implemented on the internal server to respond to the ping.

2.4.1.62. reverse_proxy/reverse_proxy_group/reverse_proxy_server/persistent_server_check_time

Specifies the interval for checking connection with an internal server in order to manage internal server connections.

To maintain internal server connections, this must be set to a value less than the keepalive_timeout of the internal server. A single ping is sent to connections in the Ready state, and the connection is disconnected if no reply is received within the persistent_server_check_time period.

Item	Description
Data Type	Integer
Unit	seconds
Range	0 ~ INT_MAX
Default Value	30

The following describes each configuration value.

Value	Description
0	no ping is sent and connections are maintained for the keepalive_timeout period of the external server.

2.4.1.63. reverse_proxy/reverse_proxy_group/reverse_proxy_server/persistent_server_timeout

Specifies the timeout for terminating a connection in the Ready state when the number of internal server connections has reached the persistent_server_connections_min.

Item	Description
Data Type	Integer
Unit	seconds
Range	0 ~ INT_MAX
Default Value	300

The following describes each configuration value.

Value	Description
0	No timeout check for idle connections.

The following is an example of using persistent_server:

```
"destination": {
  "reverse_proxy":{
    "reverse_proxy_group":[
      {
        "name": "rproxyg1",
        ...
        "persistent_server_connections_min":10,
        "persistent_server_connections_max":15,
        "persistent_server_check_time": 25,
        "persistent_server_check_url":"/jsvtest/common/test.html",
        "persistent_server_timeout":3000
      }
    ]
  }
}
```

2.4.1.64. reverse_proxy/reverse_proxy_group/reverse_proxy_server/enable_cache

Sets whether the content is cached.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.65. reverse_proxy/reverse_proxy_group/reverse_proxy_server/cache_refresh

Specifies the value to calculate the time a cached response is valid when a response handled by a reverse proxy is cached.

When WebtoB responds to a client’s request with cached content, it checks whether the cached content is valid. The validity is determined based on the set value (valid time).

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	3600

Item	Description
Priority	The priority of the setting is as follows: 1. Cache-Control:max-age: If the Cache-Control:max-age value is in the Response Header when the header is being cached, the header is specified to be valid for the max-age value (second). 2. Expires: If the Expires value is in the Response Header when the header is being cached, the header is specified to be valid until the Expires value (hour). 3. cache_refresh: If there are no 'Cache-Control:max-age' or 'Expires' values in the response header, the response received from JEUS is valid for cache_refresh (in seconds) after being cached.

2.4.1.66. reverse_proxy/reverse_proxy_group/reverse_proxy_server/sticky_session_routing_id

Used for integration with a specific internal server (web application server) and using sticky session routing.

Use the name of the engine that corresponds to the Sticky Session id (JSESSIONID) value set in the Set-Cookie response header by the internal server (WAS). For example, if the "JSESSIONID" value in the Set-Cookie response header is "Pl1xfBkEVbUu2cj20CUNlHJoWLmU.xxx_servlet_engine1", use the value following the delimiter (period) which is "xxx_servlet_engine1".

Item	Description
Data Type	String
Range	Up to 255 characters

2.4.1.67. reverse_proxy/reverse_proxy_group/reverse_proxy_server/is_backup_server

Sets whether this server should be used as a backup server in case all other reverse proxy servers are not ready.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.68. reverse_proxy/reverse_proxy_group/reverse_proxy_server/load_balancing_factor

Sets the ratio at which requests will be distributed to the corresponding reverse proxy servers in the reverse proxy group.

Requests are distributed in the ratio of "(load_balancing_factor of the reverse proxy server) / (sum of load_balancing_factors of all reverse proxy servers in the group)".

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	1

2.4.1.69. reverse_proxy/reverse_proxy_group/reverse_proxy_server/common_config

A common setting for reverse proxies. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. reverse_proxy_server 2. reverse_proxy_group 3. reverse_proxy

2.4.1.70. reverse_proxy/reverse_proxy_group/server_schedule

Sets the method for specifying the reverse proxy to process requests when multiple reverse proxies are set up.

Item	Description
Data Type	String
Range	"RR"
Default Value	"RR"

The following describes each configuration value.

Value	Description
RR	Assigns requests to reverse proxies sequentially using the round robin method.

2.4.1.71. reverse_proxy/reverse_proxy_group/sticky_session_routing

Settings related to session routing.

Item	Description
Data Type	Object

2.4.1.72. reverse_proxy/reverse_proxy_group/sticky_session_routing/policy

Sets the session routing policy.

Item	Description
Data Type	String
Range	"AddNewCookie" | "ModifyOriginalCookie" | "UseOriginalCookie"
Default Value	"UseOriginalCookie"

2.4.1.73. reverse_proxy/reverse_proxy_group/sticky_session_routing/session_id_cookie_key

Specifies the Key name of the HTTP cookie used for session routing.

Item	Description
Data Type	String
Default Value	"JSESSIONID"

2.4.1.74. reverse_proxy/reverse_proxy_group/sticky_session_routing/enable_flexible_sticky_session_routing

Sets whether to enable flexible routing during session routing.

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
True	If connections up to persistent_server_connections_max of the internal server are in the RUN state when using the sticky_session_routing_id for session routing, requests are routed to another internal server connection with a different sticky_session_routing_id instead of being queued.

Value	Description
False	Uses the sticky session routing by default. Requests are routed only to internal server connections with the same sticky_session_routing_id, and requests are queued if connections up to persistent_server_connections_max are all in the RUN state.

If you are using flexible routing, requests from different clients with the same JSESSIONID may be routed to different servers. (Default setting is recommended.)

2.4.1.75. reverse_proxy/reverse_proxy_group/common_config

A common setting for reverse proxies. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. reverse_proxy_server 2. reverse_proxy_group 3. reverse_proxy

2.4.1.76. htmls

Configures this setting to handle static file requests. Supported HTTP methods are "GET", "POST", and "HEAD", while a "405 Method Not Allowed" response is returned for all other methods.

Item	Description
Data Type	Array (object)

2.4.1.77. htmls/name (Required)

Specifies the name of the htmls.

Item	Description
Data Type	String
Range	Up to 31 characters

2.4.1.78. htmls/headers

Sets the name of the HEADERS section to apply.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

2.4.1.79. htmls/compression

Sets the targets for compression in responses for static files. Specifies the MIME-types (content-types) of files to compress. The corresponding response will be compressed using GZIP before being sent to the client.

Item	Description
Data Type	Array (string)
Range	Up to 32 items (within 255 characters)

Compression can reduce the network traffic, but it may degrade server performance.

Files with low compression ratio, such as zip or jpeg, should not be compressed if possible to avoid server overhead.

Compression can only be used for requests with Accept-Encoding set to GZIP or deflate in the HTTP request header.

2.4.1.80. htmls/compression_min_size

Specifies the minimum size of the response to compress. If the requested file size is greater than the specified size, the response is compressed.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	0

2.4.1.81. htmls/enable_cache

Sets whether the content is cached.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.82. htmls/cache_refresh

Sets the validity period for cached responses whose 'Content-Type' is 'text/html'. That is, a 'text/html' response will only be valid for the set number of seconds after being cached.

Item	Description
Data Type	integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	3600

A conditional-GET request checks to see if the cached response has been modified. If it has been modified, the cache is deleted and updated with the new version.

2.4.1.83. htmls/queue_timeout

When a request cannot be processed due to a large volume of user requests, the request waits in the server queue until a server process becomes available.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	There is no time limit for waiting in the server queue.
Positive Integer	Requests waiting longer than the specified time will be removed from the queue and respond with "503 Service Unavailable".

2.4.1.84. htmls/enable_sendfile

Configures whether to use the sendfile function.

Item	Description
Data Type	Boolean
Default Value	False

2.4.1.85. htmls/sendfile_min_size

If a requested file size is greater than the specified value, the sendfile function is used.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	8192

2.4.1.86. htmls/enable_etag

Configures whether to use ETag.

Item	Description
Data Type	Boolean
Default Value	True

The following describes each configuration value.

Value	Description
False	ETag is not added to HTTP response, and ETag of HTTP request is ignored.

2.4.2. Example

The following is an example of configuring the DESTINATION section:

```
"destination": {
    "jeus": [
        {
            "name": "MyGroup"
        }
    ],
    "reverse_proxy": {
        "common_config": {
                "server_health_check": {
                    "retry_count": 3,
                    "failback_interval": 5,
                    "http_failure": ["http_invalid", "http_4xx"],
                    "connection_timeout": 5
                }
        }
        "reverse_proxy_group": [
            {
                "name": "rproxy1",
                "reverse_proxy_server": [
                    {
                        "address": "192.168.15.114:28080",
                        "enable_proxy_ssl": false
                    }
                ]
            }
        ]
    }
    "htmls": [
        {
            "name": "htmls1"
        }
    ]
}
```

2.5. ERRORDOCUMENT Section

There are four ways to troubleshoot an error in WebtoB:

- Displays the error message defined in the source code.
- Displays the error message defined by the user.
- Redirect to a local URL.
- Redirect to an external URL.

For the second, third and fourth cases, the ERRORDOCUMENT section is set to redirect to a specific page for a particular HTTP response status code. All HTTP status codes, except HTTP 401 status code, can be configured.

2.5.1. Configuration Items

The following is the environment configuration format of the ERRORDOCUMENT section.

```
#"error_document": {
    #"error_document_list": [
        {
            "name": string,
            "status": integer,          # (HTTP status code)
            "url": string
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the ERRORDOCUMENT section configuration items.

2.5.1.1. error_document_list

The following is a list of ERRORDOCUMENT settings.

Item	Description
Data Type	Array (object)

2.5.1.2. error_document_list/name (Required)

The name of the ERRORDOCUMENT setting. This 'name' must be set when using the functionality of the ERRRORDOCUMENT section in other sections.

Item	Description
Data Type	String
Range	Up to 31 characters

2.5.1.3. error_document_list/status (Required)

Specifies the HTTP status code value.

Item	Description
Data Type	Integer
Range	100 ~ 599

2.5.1.4. error_document_list/url (Required)

Specifies a relative path under the doc_root or a full path that the client can interpret.

Item	Description
Data Type	String
Range	Up to 255 characters

2.5.2. Example

The following is an example of configuring the ERRORDOCUMENT section:

```
{
    "error_document": {
        "error_document_list": [
            {
                "name": "forbidden",
                "status": 403,
                "url": "err/403.html"
            },
            {
                "name": "notfound",
                "status": 404,
                "url": "http://tmaxsoft.co.kr/404.html"
            }
        ]
    }
}
```

2.6. FILTER Section

The FILTER section specifies the FILTER section name that will use the Filter module. The Filter item is set in the NODE or SERVER section.

2.6.1. Configuration Items

The following is the configuration format of the FILTER section.

```
#"filter": {
    #"filters": [
        {
            "name": string,
            "path": string,
            #"event": [string]
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the FILTER section configuration items.

2.6.1.1. filters

The following is a list of FILTER settings.

Item	Description
Data Type	Array (object)

2.6.1.2. filters/name (Required)

The name of the FILTER setting. This 'name' must be set when using the functionality of the FILTER section in other sections.

Item	Description
Data Type	String
Range	Up to 31 characters

2.6.1.3. filters/url (Required)

Specifies the path of the Filter module within the physical directory of the server.

Item	Description
Data Type	String
Range	Up to 255 characters

2.6.1.4. filters/event

Configures the timing at which the Filter module operates.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

The following describes the filter events by type. Depending on the filter event type, the location is set in either the NODE section or the SERVER section.

- SYSTEM Filter Event

  Filters containing SYSTEM filter events are configured in the NODE section.

Filter Event	Description
ON_STOP	Operates when the WebtoB engine is terminated.
ON_START	Operates when the WebtoB engine starts.

- HTTP Filter Events

  Filters containing HTTP filter events are configured in the SERVER section.

Filter Event	Description
RECEIVE_REQUEST	Operates when an HTTP request is received from the client.
BEFORE_SEND_REQUEST	Operates before the HTTP request received from the client is forwarded to the internal server.
AFTER_SEND_REQUEST	Operates after the HTTP request received from the client is forwarded to the internal server.
RECEIVE_RESPONSE	Operates when an HTTP response is received from the internal server.
BEFORE_SEND_RESPONSE	Operates before the HTTP response received from the internal server is forwarded to the client.
AFTER_SEND_RESPONSE	Operates after the HTTP response received from the internal server is forwarded to the client.

2.6.2. Example

The following is an example configuration for implementing a Filter.

- Filter Project Structure

  The following shows the structure of a project to implement a Filter. Each file contains the elements required to define and implement a Filter.

  ```
  .
  ├──── CMakeLists.txt
  ├──── SampleSystemFilter.h
  ├──── SampleSystemFilter.cpp
  ├──── SampleHttpFilter.h
  └──── SampleHttpFilter.cpp
  ```

  The configuration examples for each file are as follows:

  - CMakeLists.txt

    ```
    cmake_minimum_required(VERSION 3.15)

    project(sampleFilter)
    set(CMAKE_VERBOSE_MAKEFILE true)
    set(CMAKE_CXX_STANDARD 17)

    message("\t MODULE: ${PROJECT_NAME}")

    # Webtob Filter Include Directory
    include_directories(/WEBTOB_BINARY_PATH/include)

    # Library Add
    add_library(sampleSystemFilter SHARED SampleSystemFilter.cpp)
    add_library(sampleHttpFilter SHARED SampleHttpFilter.cpp)
    ```

- SampleSystemFilter.h

```
/**
 * @File     : SampleFilter.h
 * @Author   : Webtob
 * @Year     : 2024
 */

#ifndef WEBTOB6_SAMPLE_SYSTEM_FILTER_H
#define WEBTOB6_SAMPLE_SYSTEM_FILTER_H

#include <iostream>
#include <string>
#include <sstream>
#include <memory>
#include <vector>
#include "webtob-filter/filter-api/WebtobFilter.h"

namespace webtob {
    class SampleSystemFilter : public WebtobFilter {
    public:
        static std::vector<std::string> calledEvents;

        FilterResult onStart() override;

        FilterResult onStop() override;

        FilterResult onReceiveRequest(const std::shared_ptr<FilterRequest>& request,
                                      const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onReceiveResponse(const std::shared_ptr<FilterRequest>& request,
                                       const std::shared_ptr<FilterResponse>& response)
override;

    private:
        static std::string getStaticVector();

        void printCalledInfo(const char* calledEvent);

        std::string filterName = "SampleSystemFilter";
    };
} // namespace webtob

#endif //WEBTOB6_SAMPLE_SYSTEM_FILTER_H
```

- SampleSystemFilter.cpp

```
#include "SampleSystemFilter.h"

using namespace webtob;

std::vector<std::string> SampleSystemFilter::calledEvents;

// Filter Creator
extern "C" [[maybe_unused]] std::shared_ptr<WebtobFilter> createFilter() {
    return std::make_shared<SampleSystemFilter>();
}

std::string SampleSystemFilter::getStaticVector() {
    std::ostringstream description;

    description << "Sample SystemFilter Event Called Vector= [\n";
    for (const std::string& event : calledEvents) {
        description << "\t" << event << "\n";
    }
    description << "]";
    return description.str();
}

void SampleSystemFilter::printCalledInfo(const char* calledEvent) {
    calledEvents.emplace_back(calledEvent);
    std::ostringstream message;
    message << "\n=============================================";
    message << "\n" << filterName << " : " << calledEvent;
    message << "\n---------------------------------------------";
    message << "\n" << getStaticVector();
    message << "\n=============================================";
    std::cout << message.str() << std::endl;
}

FilterResult SampleSystemFilter::onStart() {
    printCalledInfo(__FUNCTION__);
    return FilterResult::SUCCESS;
}


FilterResult SampleSystemFilter::onStop() {
    printCalledInfo(__FUNCTION__);
    return FilterResult::SUCCESS;
}

FilterResult SampleSystemFilter::onReceiveRequest(const std::shared_ptr<FilterRequest>&
request,
                                                  const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    return FilterResult::SUCCESS;
}


FilterResult SampleSystemFilter::onReceiveResponse(const std::shared_ptr<FilterRequest>&
request,
                                                   const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    return FilterResult::SUCCESS;
}
```

- SampleHttpFilter.h

```
/**
 * @File     : SampleHttpFilter.h
 * @Author   : Webtob
 * @Year     : 2024
 */
#ifndef WEBTOB6_SAMPLE_HTTP_FILTER_H
#define WEBTOB6_SAMPLE_HTTP_FILTER_H

#include <iostream>
#include <sstream>
#include "webtob-filter/filter-api/WebtobFilter.h"

namespace webtob {
    class SampleHttpFilter : public WebtobFilter {
    public:
        FilterResult onReceiveRequest(const std::shared_ptr<FilterRequest>& request,
                                      const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onBeforeSendRequest(const std::shared_ptr<FilterRequest>& request,
                                         const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onAfterSendRequest(const std::shared_ptr<FilterRequest>& request,
                                        const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onReceiveResponse(const std::shared_ptr<FilterRequest>& request,
                                       const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onBeforeSendResponse(const std::shared_ptr<FilterRequest>& request,
                                          const std::shared_ptr<FilterResponse>& response)
override;

        FilterResult onAfterSendResponse(const std::shared_ptr<FilterRequest>& request,
                                         const std::shared_ptr<FilterResponse>& response)
override;

        std::string filterName = "SampleHttpFilter";
    private:
        void printCalledInfo(const char* calledEvent);
    };
} // namespace webtob

#endif //WEBTOB6_SAMPLE_HTTP_FILTER_H
```

- SampleHttpFilter.cpp

```
#include "SampleHttpFilter.h"
#include <string_view>

using namespace webtob;

// Filter Creator
extern "C" [[maybe_unused]] std::shared_ptr<WebtobFilter> createFilter() {
    return std::make_shared<SampleHttpFilter>();
}

```

```
bool startsWith(const std::string_view& str, const std::string_view& prefix) {
    return str.substr(0, prefix.size()) == prefix;
}

void SampleHttpFilter::printCalledInfo(const char* calledEvent) {
    std::ostringstream message;
    message << "\n=============================================";
    message << "\n" << filterName << " : " << calledEvent;
    message << "\n=============================================";
    std::cout << message.str() << std::endl;
}

FilterResult SampleHttpFilter::onReceiveRequest(const std::shared_ptr<FilterRequest>&
request,
                                                const std::shared_ptr<FilterResponse>&
response) {

    printCalledInfo(__FUNCTION__);

    std::string path = request->getPath();
    if (startsWith(path, "/service/")) {
        if (startsWith(path, "/service/A")) {
            response->setStatusCode(200);
            response->setBody("Filter makes service A response");
        } else if (startsWith(path, "/service/B")) {
            response->setStatusCode(200);
            response->setBody("Service B response");
        } else if (startsWith(path, "/service/C")) {
            response->setStatusCode(400);
            response->setBody("BAD Response Makes");
        }
        return FilterResult::TERMINATE;
    }

    auto queryString = request->getQueryString();

    if (queryString == "test=reject1") {
        response->setStatusCode(200);
        response->setBody("I decide to reject");
        response->addHeader("ErrorCheck", "Reject1_Detected");
        return FilterResult::TERMINATE;
    }
    request->addHeader("Filter", "InsertOn");
    request->addCookie("FilterCookieKey", "OK_Inserted");
    return FilterResult::SUCCESS;
}

FilterResult SampleHttpFilter::onBeforeSendRequest(const std::shared_ptr<FilterRequest>&
request,
                                                   const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    auto queryString = request->getQueryString();
    if (queryString == "test=reject2") {
        response->setStatusCode(200);
        response->setBody("I decide to reject onBeforeSendRequest");
        response->addHeader("ErrorCheck", "onBeforeSendRequest");
        return FilterResult::TERMINATE;
    }

    return FilterResult::SUCCESS;
}

FilterResult SampleHttpFilter::onAfterSendRequest(const std::shared_ptr<FilterRequest>&
request,
                                                  const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    auto queryString = request->getQueryString();
    if (queryString == "test=reject3") {
        response->addHeader("ErrorCheck", "onAfterSendRequest");
        return FilterResult::TERMINATE;
    }
    return FilterResult::SUCCESS;
}

FilterResult SampleHttpFilter::onReceiveResponse(const std::shared_ptr<FilterRequest>&
request,
                                                 const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    auto cookieBuilder = response->getCookieBuilder();
    auto cookie = cookieBuilder-
>newCookie().setName("Cookie_ReceiveResponse").setValue("AddFromResponse")
        .setHttpOnly(true).setPath("/").setSameSite(FilterCookie::SameSite::OFF).build();
    response->addCookie(cookie);
    response->addHeader("Filter_response", "OK_ADDED");
    return FilterResult::SUCCESS;
}

FilterResult SampleHttpFilter::onBeforeSendResponse(const std::shared_ptr<FilterRequest>&
request,
                                                    const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    auto cookieBuilder = response->getCookieBuilder();
    auto cookie = cookieBuilder->newCookie().setName("MY_COOKIE").setValue("AddFromResponse")
        .setHttpOnly(true).setPath("/").setSameSite(FilterCookie::SameSite::OFF).build();
    response->addCookie(cookie);

    return FilterResult::SUCCESS;
}

FilterResult SampleHttpFilter::onAfterSendResponse(const std::shared_ptr<FilterRequest>&
request,
                                                   const std::shared_ptr<FilterResponse>&
response) {
    printCalledInfo(__FUNCTION__);
    return FilterResult::SUCCESS;
}
```

- Build

```
~/workspace/myTestFilter> cmake . -B build
-- The C compiler identification is AppleClang 15.0.0.15000309
-- The CXX compiler identification is AppleClang 15.0.0.15000309
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
     MODULE: sampleFilter
-- Configuring done (0.5s)
-- Generating done (0.0s)
-- Build files have been written to: /Users/seungwook/workspace/myTestFilter/build
~/workspace/myTestFilter>

~/workspace/myTestFilter> cd build && make
```

- Configuration

  The following is an example of setting the FILTER section in the environment configuration file.

```
{
    "filter": {
        "filters": [
            {
                "name": "filter1",
                "path": "TargetRealPath/myTestFilter/libsampleSystemFilter.so",
                "event": [ "ON_START", "ON_STOP" ]
            },
            {
                "name": "filter2",
                "path": "TargetRealPath/myTestFilter/libsampleHttpFilter.so",
                "event": [ "RECEIVE_REQUEST", "RECEIVE_RESPONSE" ]
            },
        ]
    }
}
```

2.7. HEADERS Section

The HEADERS section is used when a specific HTTP Header is changed based on user requests or responses. Headers items can be configured in the SERVER or DESTINATION sections.

2.7.1. Configuration Items

The following is the configuration format of the HEADERS section.

```
#"headers": {
    #"headers_list": [
        {
            "name": string,
            "action": string,
            "field_name": string,
            #"field_value": string,
            #"reg_exp": string,
            #"status_code": integer     # 0
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the HEADERS section configuration items.

2.7.1.1. headers_list

The following is a list of HEADERS settings.

Item	Description
Data Type	Array (object)

2.7.1.2. headers_list/name (Required)

The name of the HEADERS setting. This 'name' must be set when using the functionality of the HEADERS section in other sections.

Item	Description
Data Type	String
Range	Up to 255 characters

2.7.1.3. headers_list/action (Required)

Configures the action for handling HTTP headers.

Item	Description
Data Type	String
Range	"AddRequest" | "AddResponse" | "AddIfAbsentRequest" | "AddIfAbsentResponse" | "AppendResponse" | "EchoResponse" | "SetResponse" | "UnsetResponse"

The following describes each action type.

Action	Description
AddRequest	Adds a specified HTTP header in the request. Caution is advised as duplicates may occur if the specified header already exists in the request.
AddResponse	Adds a specified HTTP header in the response. Caution is advised as duplicates may occur if the specified header already exists in the response.
AddIfAbsentRequest	Adds a specified HTTP header when the request contains no HTTP headers.
AddIfAbsentResponse	Adds a specified HTTP header when the response contains no HTTP headers.
AppendResponse	Appends a specified FieldValue at the end of an existing header value. If there is no existing header, no action occurs.
EchoResponse	Adds the same header and value to the response as the request, if the request contains a specified header. This action has no FieldValue.
SetResponse	Adds a specified HTTP header when the response contains no HTTP headers. If the header already exists, the header value is replaced by the specified FieldValue.
UnsetResponse	Deletes a specified header if it already exists in the response. This action has no FieldValue.

Certain Action types have required field_value.

2.7.1.4. headers_list/field_name (Required)

Specifies the name of the HTTP header.

Item	Description
Data Type	String
Range	Up to 255 characters

2.7.1.5. headers_list/field_value

Specifies the value of the HTTP header.

Item	Description
Data Type	String
Range	Up to 255 characters

2.7.1.6. headers_list/reg_exp

Set to control headers when the HTTP request URL matches a Regular expression pattern.

Item	Description
Data Type	String
Range	Up to 511 characters

2.7.1.7. headers_list/status_code

Set to control headers for a specific HTTP status code.

Item	Description
Data Type	Integer
Range	0 ~ 599
Default Value	0

2.7.2. Example

The following is an example of configuring the HEADERS section:

```
{
    "headers": {
    "headers_list": [
      {
        "name": "header1",
        "action": "AppendResponse",
        "field_name": "Test_Header",
        "field_value": "test"
      }
    ]
  }
}
```

2.8. LOGGING Section

The LOGGING section configures client request log. Access log and error log are recorded separately, and their format can be configured. System logs, access logs, and error logs are all set in the LOGGING section.

2.8.1. Configuration Items

The following is the configuration format of the LOGGING section.

```
"logging": {
    "system_log": [
        {
            "name": string,
            #"level": string,                               # "INFO"
            #"dump": [string],
            #"handlers": {
                #"file_handler": {
                    "file_name": string,                    # "webtob_system.log"
                    #"rotate_by_seconds": integer,          # 0
                    #"valid_hours": integer,                # 0
                    #"rotate_by_file_size": integer,        # 0
                    #"archive_file_name": string,
                    #"enable_sync": boolean,                # false
                    #"permission": string                   # "0600"
                },
                #"enable_console_handler": boolean          # false
            }
        }
    ],
    #"access_log": [
        {
            "name": string,
            #"level": string,                               # "INFO"
            #"format": string,                              # "DEFAULT"
            #"exclude_by_ext": string,
            #"handlers": {
                #"file_handler": {
                    "file_name": string,                    # "webtob_access.log"
                    #"rotate_by_seconds": integer,          # 0
                    #"valid_hours": integer,                # 0
                    #"rotate_by_file_size": integer,        # 0
                    #"archive_file_name": string,
                    #"enable_sync": boolean,                # false
                    #"permission": string                   # "0600"
                },
                #"enable_console_handler": boolean          # false
            }
        }
    ],
    #"error_log": {
        #"level": "string",                                 # "INFO"
        #"format": "string",                                # "ERROR"
        #"enable_exclude_client_address_on_error": boolean, # false
        #"handlers": {
            #"file_handler": {
                "file_name": string,                        # "webtob_error.log"
                #"rotate_by_seconds": integer,              # 0
                #"valid_hours": integer,                    # 0
                #"rotate_by_file_size": integer,            # 0
                #"archive_file_name": string,
                #"enable_sync": boolean,                    # false
                #"permission": string                       # "0600"
            },
            #"enable_console_handler": boolean              # false
        }
    },
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the LOGGING section configuration items.

2.8.1.1. system_log (Required)

Configures the system log.

Item	Description
Data Type	Array (object)

2.8.1.2. system_log/name (Required)

Specifies the target system logger to be set.

Item	Description
Data Type	Object

Logger names are hierarchical and separated by a period (.). The settings for loggers prioritize the lowest-level configuration. For example, if both (A)'webtob' and (B)'webtob.http' are configured, the 'webtob.http' logger follows the configuration (B), while other loggers follow the configuration (A). The top-level system logger 'webtob' must be configured.

The following describes the loggers that can be configured.

Logger Name	Description
webtob.http	Manages logs related to handling HTTP messages.
webtob.wjp	Manages logs related to handling WJP messages.
webtob.ssl	Manages logs related to SSL.
webtob.network	Manages logs related to network operations.
webtob.server	Manages logs related to the startup and termination of each thread.
webtob.em.acceptor	Manages logs related to the Acceptor, which handles new connections.
webtob.em.control	Manages logs related to Admin API communication.
webtob.em.worker	Manages logs related to HTH.
webtob.util	Manages logs related to other utility functions.

The following is a configuration example:

```
"system_log": [
  {
    "name": "webtob",
    "level": "DEBUG",
    "handlers": {
      "file_handler": {
        "file_name": "webtob.log"
      },
      "enable_console_handler": false
    },
    "dump": [
      "ToClient",
      "FromClient",
      "ToServer",
      "FromServer",
      "SSLRead",
      "SSLWrite"
    ]
  },
  {
    "name":"webtob.http",
    "level":"DEBUG",
    "handlers": {
      "file_handler": {
        "file_name": "webtob_http.log"
      },
      "enable_console_handler": false
    }

  }
]
```

2.8.1.3. system_log/level

Specifies the log level.

If the level of the system log message is higher than the set value, it will be output. The supported log levels from lowest to highest are "TRACE", "DEBUG", "INFO", "WARNING", "FATAL", and "OFF".

Item	Description
Data Type	String
Range	"TRACE" | "DEBUG" | "INFO" | "WARNING" | "FATAL" | "OFF"
Default Value	"INFO"

The lower the level, the more detailed the log messages are output.

2.8.1.4. system_log/dump

Configures whether to output dump logs. This setting applies only when the log level is DEBUG or TRACE, and can only be set for loggers named "webtob".

Item	Description
Data Type	Array (string)
Range	Up to 6 items

The following describes each configuration value.

Value	Description
FromClient	Logs data sent by the client.
ToClient	Logs data sent to the client.
FromServer	Logs data sent by the server.
ToServer	Logs data sent to the server.
SSLRead	Logs data read with SSL encryption.
SSLWrite	Logs data that will be written with SSL encryption.

2.8.1.5. system_log/handlers

Sets the file and console to which log messages will be recorded.

Item	Description
Data Type	Object

2.8.1.6. system_log/handlers/file_handler

Sets the file to which logs will be recorded.

Item	Description
Data Type	Object

2.8.1.7. system_log/handlers/file_handler/file_name (Required)

Sets the path to the file where the system log will be stored. If the relative path does not start with a slash (/), it is automatically replaced with "$WEBTOB6_HOME_PATH/relative path".

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"webtob_system.log"

The following substitution strings specified in the file name are replaced with the actual values when the file is created. For example, a value set to "log/system_%Y%%M%%D%.log" in FileName can be replaced with "$WEBTOB6_HOME_PATH/log/system_20231023.log" when the log file is created.

The following describes each substitution string:

String	Actual Value
%Y%	4-digit year (Example: 2009)
%M%	2-digit month (Example: 11)
%D%	2-digit day (Example: 05)
%h%	2-digit hour (Example: 10)
%m%	2-digit minute (Example: 30)
%s%	2-digit second (Example: 45)

2.8.1.8. system_log/handlers/file_handler/rotate_by_seconds

Specifies the maximum period of time to create a new log file. If an existing log file is older than this period of time, a new log file is created to record a new log.

Item	Description
Data Type	Integer
Range	0 ~ 86400
Default Value	0

2.8.1.9. system_log/handlers/file_handler/valid_hours

Specifies the creation of a new log file at the specified interval.

Item	Description
Data Type	Integer
Range	0 ~ 23
Default Value	0

2.8.1.10. system_log/handlers/file_handler/rotate_by_file_size

Specifies the file size threshold for log messages. If an existing log file surpasses this specified size, a new log file will be created.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX

Item	Description
Default Value	0

2.8.1.11. system_log/handlers/file_handler/archive_file_name

Specifies the log file name format for an archived file. When a new log file is created, the existing log file is renamed as set in archive_file_name. New log files will then be created in the format specified in file_name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.8.1.12. system_log/handlers/file_handler/enable_sync

Specifies whether to write to the file immediately when logs are recorded.

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
True	Log messages are written directly to files without being buffered in WebtoB memory. **[Note]** Services that require immediate logs should use this feature to immediately identify errors.

2.8.1.13. system_log/handlers/file_handler/permission

Specifies the permission for system log files.

This configuration is the same as the file access permissions used in UNIX and can be used only in a UNIX/Linux environment.

Item	Description
Data Type	String
Range	"0600" ~ "0777"

2.8.1.14. system_log/handlers/enable_console_handler

Specifies whether to display the log in the console.

Item	Description
Data Type	Boolean
Range	False

2.8.1.15. access_log (Required)

Configures access logs.

Item	Description
Data Type	Array (object)

2.8.1.16. access_log/name

Sets the name of the access logger. This name is used to specify the target for logging access logs in other sections, such as the HTTP section.

Item	Description
Data Type	Object

2.8.1.17. access_log/level

Sets the access log level.

Item	Description
Data Type	String
Range	"INFO" | "OFF"
Default Value	"INFO"

The following describes each configuration value.

Value	Description
OFF	Does not output access logs.

2.8.1.18. access_log/format

Sets the format of the messages to be recorded in the access log file.

Item	Description
Data Type	String
Range	Up to 255 characters

The following describes each configuration value.

Value	Description
DEFAULT	Default log file format. (Log format: "%h %t \"%r\" %s %b %D")
COMMON	Common log file format. (Log format: "%h %l %u %t \"%r\" %s %b")
COMBINED	Combined log file format. (Log format: "%h %l %u %t \"%r\" %s %b \"%{*Referer*}i\" \"%{*User-Agent*}i\"")
COMBINEDIO	CombinedIO log file format. (Log format: "%h %l %u %t \"%r\" %s %b \"%{*Referer*}i\" \"%{*User-Agent*}i\" %I %O")
ERROR	Error log file format. (Log format: "%r")
%a	Displays the IP address of the device that sent the request. Same as %h.
%A	Displays the IP address of the server.
%b	Displays the bytes of the response excluding headers.
%{_attr_name_}C	Displays the value corresponding to '_attr_name_' among the cookie header values of the HTTP request.
%d	Displays the time the response was sent.
%D	Displays the time taken to process the request (unit: milliseconds).
%{ENV_NAME}e	Displays the environment variable ENV_NAME.
%h	Displays the IP address of the device that sent the request. Same as %a.
%H	Displays the HTTP version used.
%{HEADER_FIELD}i	Displays the value of the HEADER_FIELD header of an HTTP request.
%I	Displays the bytes of the request.
%l	Displays the name for remote login.
%m	Displays the HTTP request method.
%O	Displays the bytes of the response.
%p	Displays the server port number that received the request.
%q	Displays the query value of the HTTP request.
%r	Displays the entire request line of the HTTP request.
%s	Displays the HTTP status code used in the response.

Value	Description
%S	Displays HTTP and HTTPS.
%t	Displays the time when the request processing is completed.
%T	Displays the time taken to process the request (unit: seconds).
%u	Displays the username used for HTTP authentication.
%U	Displays the HTTP request URI.
%v	Displays the Host Header field value.

2.8.1.19. access_log/handlers

Sets the file and console to which log messages will be recorded.

Item	Description
Data Type	Object

2.8.1.20. access_log/handlers/file_handler

Sets the file to which logs will be recorded.

Item	Description
Data Type	Object

2.8.1.21. access_log/handlers/file_handler/file_name (Required)

Sets the path to the file where the access log will be stored. If the relative path does not start with a slash (/), it is automatically replaced with "$WEBTOB6_HOME_PATH/relative path".

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"webtob_access.log"

The following substitution strings specified in the file name are replaced with the actual values when the file is created. For example, a value set to "log/access_%Y%%M%%D%.log" in FileName can be replaced with "$WEBTOB6_HOME_PATH/log/access_20231023.log" when the log file is created.

The following describes each substitution string:

String	Actual Value
%Y%	4-digit year (Example: 2009)

String	Actual Value
%M%	2-digit month (Example: 11)
%D%	2-digit day (Example: 05)
%h%	2-digit hour (Example: 10)
%m%	2-digit minute (Example: 30)
%s%	2-digit second (Example: 45)

2.8.1.22. access_log/handlers/file_handler/rotate_by_seconds

Specifies the maximum period of time to create a new log file. If an existing log file is older than this period of time, a new log file is created to record a new log.

Item	Description
Data Type	Integer
Range	0 ~ 86400
Default Value	0

2.8.1.23. access_log/handlers/file_handler/valid_hours

Specifies the creation of a new log file at the specified interval.

Item	Description
Data Type	Integer
Range	0 ~ 23
Default Value	0

2.8.1.24. access_log/handlers/file_handler/rotate_by_file_size

Specifies the file size threshold for log messages. If an existing log file surpasses this specified size, a new log file will be created.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	0

2.8.1.25. access_log/handlers/file_handler/archive_file_name

Specifies the log file name format for an archived file. When a new log file is created, the existing log file is renamed as set in archive_file_name. New log files will then be created in the format specified in file_name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.8.1.26. access_log/handlers/file_handler/enable_sync

Specifies whether to write to the file immediately when logs are recorded.

Item	Description
Data Type	Boolean
Range	False

The following describes each configuration value.

Value	Description
True	Log messages are written directly to files without being buffered in WebtoB memory. **[Note]** Services that require immediate logs should use this feature to immediately identify errors.

2.8.1.27. access_log/handlers/file_handler/permission

Specifies the permission for access log files.

This configuration is the same as the file access permissions used in UNIX and can be used only in a UNIX/Linux environment.

Item	Description
Data Type	String
Range	"0600" ~ "0777"

2.8.1.28. access_log/handlers/enable_console_handler

Specifies whether to display the log in the console.

Item	Description
Data Type	Boolean
Default Value	False

2.8.1.29. error_log (Required)

Sets up error logs.

Item	Description
Data Type	Array (object)

2.8.1.30. error_log/level

Sets the error log level.

Item	Description
Data Type	String
Range	"INFO" | "OFF"
Default Value	"INFO"

The following describes each configuration value.

Value	Description
OFF	Does not output error logs.

2.8.1.31. error_log/format

Sets the format of the messages to be recorded in the error log file.

Item	Description
Data Type	String
Range	Up to 255 characters

The following describes each configuration value.

Value	Description
DEFAULT	Default log file format. (Log format: "%h %t \"%r\" %s %b %D")
COMMON	Common log file format. (Log format: "%h %l %u %t \"%r\" %s %b")

Value	Description
COMBINED	Combined log file format. (Log format: "%h %l %u %t \"%r\" %s %b \"%{*Referer*}i\" \"%{*User-Agent*}i\"")
COMBINEDIO	CombinedIO log file format. (Log format: "%h %l %u %t \"%r\" %s %b \"%{*Referer*}i\" \"%{*User-Agent*}i\" %I %O")
ERROR	Error log file format. (Log format: "%r")
%a	Displays the IP address of the device that sent the request. Same as %h.
%A	Displays the IP address of the server.
%b	Displays the bytes of the response excluding headers.
%{_attr_name_}C	Displays the value corresponding to '_attr_name_' among the cookie header values of the HTTP request.
%d	Displays the time the response was sent.
%D	Displays the time taken to process the request (unit: milliseconds).
%{ENV_NAME}e	Displays the environment variable ENV_NAME.
%h	Displays the IP address of the device that sent the request. Same as %a.
%H	Displays the HTTP version used.
%{HEADER_FIELD}i	Displays the value of the HEADER_FIELD header of an HTTP request.
%I	Displays the bytes of the request.
%l	Displays the name for remote login.
%m	Displays the HTTP request method.
%O	Displays the bytes of the response.
%p	Displays the server port number that received the request.
%q	Displays the query value of the HTTP request.
%r	Displays the entire request line of the HTTP request.
%s	Displays the HTTP status code used in the response.
%S	Displays HTTP and HTTPS.
%t	Displays the time when the request processing is completed.
%T	Displays the time taken to process the request (unit: seconds).
%u	Displays the username used for HTTP authentication.
%U	Displays the HTTP request URI.
%v	Displays the host header field value.

2.8.1.32. error_log/handlers

Sets the file and console to which log messages will be recorded.

Item	Description
Data Type	Object

2.8.1.33. error_log/handlers/file_handler

Sets the file to which logs will be recorded.

Item	Description
Data Type	Object

2.8.1.34. error_log/handlers/file_handler/file_name (Required)

Sets the path to the file where the error log will be stored. If the relative path does not start with a slash (/), it is automatically replaced with "$WEBTOB6_HOME_PATH/relative path".

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"webtob_error.log"

The following substitution strings specified in the file name are replaced with the actual values when the file is created. For example, a value set to "log/error_%Y%%M%%D%.log" in FileName can be replaced with "$WEBTOB6_HOME_PATH/log/error_20231023.log" when the log file is created.

The following describes each substitution string:

String	Actual Value
%Y%	4-digit year (Example: 2009)
%M%	2-digit month (Example: 11)
%D%	2-digit day (Example: 05)
%h%	2-digit hour (Example: 10)
%m%	2-digit minute (Example: 30)
%s%	2-digit second (Example: 45)

2.8.1.35. error_log/handlers/file_handler/rotate_by_seconds

Specifies the maximum period of time to create a new log file. If an existing log file is older than this period of time, a new log file is created to record a new log.

Item	Description
Data Type	Integer
Range	0 ~ 86400
Default Value	0

2.8.1.36. error_log/handlers/file_handler/valid_hours

Specifies the creation of a new log file at the specified interval.

Item	Description
Data Type	Integer
Range	0 ~ 23
Default Value	0

2.8.1.37. error_log/handlers/file_handler/rotate_by_file_size

Specifies the file size threshold for log messages. If an existing log file surpasses this specified size, a new log file will be created.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	0

2.8.1.38. error_log/handlers/file_handler/archive_file_name

Specifies the log file name format for an archived file. When a new log file is created, the existing log file is renamed as set in archive_file_name. New log files will then be created in the format specified in file_name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.8.1.39. error_log/handlers/file_handler/enable_sync

Specifies whether to write to the file immediately when logs are recorded.

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
True	Log messages are written directly to files without being buffered in WebtoB memory. **[Note]** Services that require immediate logs should use this feature to immediately identify errors.

2.8.1.40. error_log/handlers/file_handler/permission

Specifies the permission for error log files.

This configuration is the same as the file access permissions used in UNIX and can be used only in a UNIX/Linux environment.

Item	Description
Data Type	String
Range	"0600" ~ "0777"

2.8.1.41. error_log/handlers/enable_console_handler

Specifies whether to display the log in the console.

Item	Description
Data Type	Boolean
Default Value	False

2.8.2. Example

The following is an example of configuring the LOGGING section:

```
"logging": {
    "system_log": [
        {
            "name": "webtob",
            "level": "DEBUG",
            "handlers": {
                "file_handler": {
                    "file_name": "webtob-%Y%%M%%D%-%h%%m%%s%.log"
            }
        }
    ],
    "access_log": [
        {
            "name": "access_log1",
            "level": "INFO",
            "format": "DEFAULT",
            "handlers": {
                "file_handler": {
                    "file_name": "access-%Y%%M%%D%%s%.log"
                }
            }
        },
        {
            "name": "access_log2",
            "level": "INFO",
            "format": "DEFAULT",
            "handlers": {
                "file_handler": {
                    "file_name": "access2-%Y%%M%%D%%s%.log"
                }
            }
        }
    ],
    "error_log": {
        "level": "INFO",
        "format": "ERROR",
        "handlers": {
            "file_handler": {
                "file_name": "logs/errors%s%.log"
            }
        }
    }
}
```

2.9. NODE Section

The NODE section configures each node that composes WebtoB. This section affects all operations related to the node.

The NODE section can define 'WebtoB system path' and 'other system configuration'.

2.9.1. Configuration Items

The following is the configuration format of the NODE section.

```
"node": {
    "name": string
    "hth_count": integer,                           # 1 (1-255)
    #"worker_threads": integer,                     # 8 (1-100)
    #"hth_schedule": string,                        # "RR"
    #"connection_pool_size": integer,               # 8192 (1-INT_MAX)
    #"graceful_shutdown_timeout": integer        # 30 (1-INT_MAX)
    #"cache_key": string,                        # "HOST_URI"
    #"cache_entry": integer                      # 128 (0-INT_MAX)
    #"max_cache_memory_size": integer            # 100 (0-INT_MAX)
    #"cache_max_file_size": integer              # 8192 (0-INT_MAX)
    #"listen_backlog": integer                   # 4096 (0-INT_MAX)
    #"limit_request_body_size": integer          # 0 (0-INT_MAX)
    #"limit_request_header_field_count": integer # 100 (0-INT_MAX)
    #"limit_request_header_field_size": integer  # 8190 (0-INT_MAX)
    #"limit_request_line_size": integer          # 8190 (0-INT_MAX)
    #"system_filters": [string]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the NODE section configuration items.

2.9.1.1. name (Required)

Specifies the node name.

Item	Description
Data Type	String
Range	Up to 127 characters

The following an example of setting the name. If the prompt displays 'user@webtob_node:~$', set the name as follows.

```
"node": {
    "name": "webtob_node",
    ...
}
```

The node name must be the same as the host name. If it is different, it cannot be run.

2.9.1.2. hth_count

Specifies the number of HTTP handlers (HTH).

Item	Description
Data Type	Integer
Range	1 ~ 100
Default Value	1

The following is an example of the hth_count configuration. If 'hth_count' is set to 3 as follows, 'HTH-0', 'HTH-1', and 'HTH-2' will be created.

```
"node": {
    "name": "webtob_node",
    "hth_count": 3,
    ...
}
```

You can check the generated HTH through the log (webtob.log).

```
EventManager(HTH-2) Thread Created. Thread ID: 4800
```

2.9.1.3. worker_threads

Sets the number of worker threads to assign to HTH. Worker threads process static files, SSL/TLS (handshake, encryption, and decryption), compression, and HTTP authentication.

Item	Description
Data Type	Integer
Range	1 ~ 100
Default Value	8

The following is an example of the worker_threads configuration. If 'worker_threads' is set to 5 as follows, 'HTMLS-0', 'HTMLS-1', 'HTMLS-2', 'HTMLS-3', and 'HTMLS-4' will be created.

```
"node": {
    "name": "webtob_node",
    "hth_count": 3,
    "worker_threads":5,
    ...
}
```

You can check the created worker thread through the log (webtob.log).

```
EventManager(HTMLS-4) Thread Created. Thread ID: 19271
```

Since the number of FDs used equals hth_count × worker_threads, you must consider the number of FDs when configuring hth_count and worker_threads. If the set value exceeds the number of FDs, the server will shut down and an error log occurs as follows:

```
Error: eventFd() failed: Too many open files
Too many open files
```

2.9.1.4. hth_schedule

Specifies the method to specify the HTH to process the request when there are multiple HTHs.

Item	Description
Data Type	String
Range	"RR"
Default Value	"RR"

The following describes each configuration value.

Value	Description
RR	Assigns requests to HTHs sequentially using the round robin method.

The following is an example of setting hth_schedule:

```
"node": {
    "name": "webtob_node",
    "hth_count": 2,
    "worker_threads":8,
    "hth_schedule": "RR",
    ...
}
```

You can check in the webtob.log file that requests are assigned sequentially based on the number set in hth_count.

```
[2024-07-25 06:49:19.113263413][DEBUG][HTH-0][SVR_3455]WJP2RequestMessageHandler.cpp:87 serviceMessage: WJP Request Loop Done.
[2024-07-25 06:49:19.113268083][DEBUG][HTH-0][NET_2004]EventManager.cpp:129 loop: End of this loop.
[2024-07-25 06:49:19.425343033][DEBUG][HTH-1][NET_4000]EpollMultiplexer.cpp:33 select: Multiplexer awakened. Event Count=0.
[2024-07-25 06:49:19.425360647][DEBUG][HTH-1][NET_2004]EventManager.cpp:129 loop: End of this loop.
[2024-07-25 06:49:20.114399067][DEBUG][HTH-0][NET_4000]EpollMultiplexer.cpp:33 select: Multiplexer awakened. Event Count=0.
[2024-07-25 06:49:20.114416261][DEBUG][HTH-0][NET_2004]EventManager.cpp:129 loop: End of this loop.
[2024-07-25 06:49:20.426438758][DEBUG][HTH-1][NET_4000]EpollMultiplexer.cpp:33
```

```
select: Multiplexer awakened. Event Count=0.
[2024-07-25 06:49:20.426456148][DEBUG][HTH-1][NET_2004]EventManager.cpp:129 loop:
End of this loop.
```

2.9.1.5. connection_pool_size

Sets the size of the pool that manages connections for each HTH.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	8192

Similar to hth_count and worker_threads, this setting is affected by the number of FDs, so the number of FDs must be considered. If the set value exceeds the number of FDs, the following error will occur and the server will shut down.

```
Error: eventFd() failed: Too many open files
Too many open files
```

2.9.1.6. graceful_shutdown_timeout

Sets a timeout to wait before forcing a shutdown after a graceful shutdown command.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX

The following is an example of setting graceful_shutdown_timeout:

```
{
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "graceful_shutdown_timeout":5
  },
  ...
```

2.9.1.7. cache_key

Sets the format of the key used to identify cached data.

Item	Description
Data Type	String
Range	"HOSTS" | "REAL_PATH"
Default Value	"HOST_URI"

The following describes each configuration value.

Value	Description
HOST_URI	Use the request URI and the HOST value of the request header to create a Key. Even if REQUEST_URI is the same, the VHOST depends on HTTP_HOST. This means that the actual path can be different. Although VHOST is different, the same actual path can cause duplicate caching. The response is faster than using REAL_PATH because the real path does not need to be calculated. Therefore, if DocRoot differs by VHOST, it is better to use the HOST_URI. Since responses handled by JEUS or Reverse Proxy cannot be calculated, HOST_URI value is used.
REAL_PATH	Uses the actual path of the request URI to generate a Key. This value is applied only when SVRTYPE is HTML and helps to address potential issues that may occur when using HOST_URI. Even if the HTTP_HOST is different, the actual path of the same requested file may be the same. In this case, only one file is cached regardless of the HTTP_HOST, which can reduce memory usage. However, since the actual path needs to be calculated, the response time may be slightly slower than using HOST_URI.

The following is an example of setting cache_key. To store the cache, 'enable_cache' must be set to true in the destination section.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "cache_key":"HOST_URI",
    "cache_entry":128,
    "max_cache_memory_size":100,
    "cache_max_file_size":8192
    ...
    "destination": {
      "htmls": [
      {
        "name": "htmls1",
        "enable_cache":true
      }
    ...
```

The cache status can be checked in the wsadmin’s cache-list. If the cache_key setting value is 'HOST_URI', it will be displayed as 192.168.0.1:80/image.jpg, and if it is 'REAL_PATH', it will be displayed as /home/tmax/webtob6/docs/image.jpg.

```
[wsadmin]>> cache-list
-------------------------------------------------------------------
| HTH-0: Cache List Info                                          |
-------------------------------------------------------------------
|          Cache key           |    Expired time      | Cache size |
-------------------------------------------------------------------
| 192.168.0.1:80/image.jpg | 2024-01-01 00:00:00 |         300|
-------------------------------------------------------------------
| Cache count : 1                                                 |
| Memory usages : 300                                             |
-------------------------------------------------------------------
```

2.9.1.8. cache_entry

Sets the size of the hash table Key for the HTH cache.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	128

The following describes each configuration value.

Value	Description
0	Does not cache the response.

The following is an example of setting cache_entry. To store the cache, 'enable_cache' must be set to true in the destination section.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "cache_key":"HOST_URI",
    "cache_entry":128,
    "max_cache_memory_size":100,
    "cache_max_file_size":8192
    ...
    "destination": {
      "htmls": [
      {
        "name": "htmls1",
        "enable_cache":true
      }
    ...
```

The cache status can be checked in the wsadmin's cache-list. If the cache_entry setting value is 2, a maximum of 2 rows will be displayed.

```
[wsadmin]>> cache-list
----------------------------------------------------------------------------------
-
| HTH-0: Cache List Info
|
----------------------------------------------------------------------------------
-
|                  Cache key                   |    Expired time     | Cache size
|
----------------------------------------------------------------------------------
-
| 192.168.0.1:80/image.jpg | 2024-01-01 00:00:24 |         345|
| 192.168.0.1:80/image1.jpg  | 2024-01-01 00:00:21 |         345|
----------------------------------------------------------------------------------
-
| Cache count : 2
|
| Memory usages : 1035
|
----------------------------------------------------------------------------------
-
```

2.9.1.9. max_cache_memory_size

Specifies the maximum amount of memory used by a HTH process for caching.

This is not the total memory size used by all HTH processes, but it applies to each HTH process.

Item	Description
Data Type	Integer
Unit	Mbytes
Range	0 ~ INT_MAX
Default Value	100

The following describes each configuration value.

Value	Description
0	Does not cache the response.

The following is an example of setting max_cache_memory_size. To store the cache, 'enable_cache' must be set to true in the destination section.

```
"node": {
  "name": "webtob_node",
  "hth_count": 1,
  "worker_threads":8,
  "cache_key":"HOST_URI",
  "cache_entry":128,
  "max_cache_memory_size":100,
  "cache_max_file_size":8192
  ...
  "destination": {
    "htmls": [
    {
      "name": "htmls1",
      "enable_cache":true
    }
  ...
```

The cache status can be checked in the wsadmin's cache-list. If the max_cache_memory_size setting is 100, the size of images stored in the cache cannot exceed 100 MB.

```
[wsadmin]>> cache-list
----------------------------------------------------------------------------------
-
| HTH-0: Cache List Info
|
----------------------------------------------------------------------------------
-
|                  Cache key                    |    Expired time     | Cache size
|
----------------------------------------------------------------------------------
-
| 192.168.0.1:80/image.jpg | 2024-01-01 00:00:24 |         345|
| 192.168.0.1:80/image1.jpg  | 2024-01-01 00:00:21 |         345|
----------------------------------------------------------------------------------
-
| Cache count : 2
|
| Memory usages : 1035
|
----------------------------------------------------------------------------------
-
```

2.9.1.10. cache_max_file_size

Specifies the maximum size of a response (headers + body) that can be cached.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	8192

The following describes each configuration value.

Value	Description
0	Does not cache the response.

The following is an example of setting cache_max_file_size. To store the cache, 'enable_cache' must be set to true in the destination section.

```
"node": {
  "name": "webtob_node",
  "hth_count": 1,
  "worker_threads":8,
  "cache_key":"HOST_URI",
  "cache_entry":128,
  "max_cache_memory_size":100,
  "cache_max_file_size":8192
  ...
  "destination": {
    "htmls": [
    {
      "name": "htmls1",
      "enable_cache":true
    }
  ...
```

The cache status can be checked in the wsadmin’s cache-list. If the cache_max_file_size setting is 8192, images exceeding 8192 bytes will not be cached.

```
[wsadmin]>> cache-list
---------------------------------------------------------------------------------
-
| HTH-0: Cache List Info
|
---------------------------------------------------------------------------------
-
|                  Cache key                 |    Expired time     | Cache size
|
---------------------------------------------------------------------------------
```

```
-
| 192.168.0.1:80/image.jpg | 2024-01-01 00:00:24 |         345|
| 192.168.0.1:80/image1.jpg  | 2024-01-01 00:00:21 |         345|
--------------------------------------------------------------------------------
-
| Cache count : 2
|
| Memory usages : 1035
|
--------------------------------------------------------------------------------
-
```

2.9.1.11. listen_backlog

Specifies the maximum number of queued pending connections. This limits the number of sockets attempting to access the service port.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	4096

This may be related to the number of sockets in SYN_RECVD state in the netstat command.

The following is an example of the listening_backlog setting.

```
"node": {
  "name": "webtob_node",
  "hth_count": 1,
  "worker_threads":8,
  "listen_backlog":50
  ...
```

You can check the changed Send-Q using the ss command.

If the listen_backlog setting is set to 511 or higher, the Send-Q value does not exceed 511. This can be adjusted by changing the system value using the 'sysctl -w net.core.somaxconn=[setting value]'.

```
user@webtob_node:~/tmax/webtob6/$ ss -lnt | grep 80
State  Recv-Q    Send-Q             Local Address:Port    Peer Address:Port
LISTEN   0         50                  0.0.0.0:80           0.0.0.0:*
```

2.9.1.12. limit_request_body_size

Specifies the client request body size supported by the server through the HTTP protocol.

If the HTTP request body is 2G (Long type) or larger, it can be processed starting from JEUS version 7 and later.

This value is determined based on the 'Content-Length' value of the HTTP request header, and specifies the maximum body size to allow for chunked requests. If a request body is larger than the set value, the server responds with '413 Request Entity Too Large'.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	No limit on the request body size.

The following is an example of setting limit_request_body_size.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "limit_request_body_size":100
    ...
```

If limit_request_body_size is set to 100, the following message is displayed when Content-Length exceeds 100.

```
* Closing connection 0
The body is too large.
```

2.9.1.13. limit_request_header_field_count

Specifies the maximum number of a client request's HTTP header fields.

Item	Description
Data Type	Integer

Item	Description
Range	0 ~ INT_MAX
Default Value	100

The following describes each configuration value.

Value	Description
0	No limit on the number of header fields.

The following is an example of setting limit_request_header_field_count.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "limit_request_header_field_count":1
    ...
```

When limit_request_header_field_count is set to 1, the following message is displayed if the number of header fields exceeds 1.

```
* Closing connection 0
The number of request header fields exceeds the server limit.
```

2.9.1.14. limit_request_header_field_size

Specifies the maximum size of each HTTP header field in a client request.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ 16382
Default Value	8190

The following describes each configuration value.

Value	Description
0	No limit on the size of header fields.

'WJPv2' must be used for request headers exceeding 10,000 bytes.

The following is an example of setting limit_request_header_field_size.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "limit_request_header_field_size":100
    ...
```

When limit_request_header_field_size is set to 100, the following message is displayed if the size of each header field exceeds 100.

```
* Closing connection 0
The size of a request header field exceeds the server limit.
```

2.9.1.15. limit_request_line_size

Specifies the maximum size of a client request’s HTTP line.

Item	Description
Data Type	Integer
Unit	bytes
Range	0 ~ 16382
Default Value	8190

The following describes each configuration value.

Value	Description
0	No limit on the length of the request line.

'WJPv2' must be used for request lines exceeding 10,000 bytes.

The following is an example of setting limit_request_line_size.

```
  "node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads":8,
    "limit_request_line_size":100
    ...
```

If limit_request_line_size is set to 100, the following message is displayed when the length of the request line exceeds 100.

```
* Closing connection 0
Request Uri too Wrong
```

2.9.1.16. system_filters

Sets the filter to be applied when system events (ON_START, ON_STOP) occur.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

2.9.2. Example

The following is an example of configuring the NODE section.

```
"node": {
    "name": "webtob_node",
    "hth_count": 1,
    "worker_threads": 8,
    "hth_schedule": "RR",
    "connection_pool_size": 8192,
    "graceful_shutdown_timeout": 30,
    "listen_backlog": 4096,
    "cache_key": "HOST_URI",
    "cache_entry": 128,
    "max_cache_memory_size": 100,
    "cache_max_file_size": 8192,
    "limit_request_body_size": 0,
    "limit_request_header_field_count": 100,
    "limit_request_header_field_size": 8190,
    "limit_request_line_size": 8190,
    "system_filters" : [ filter1 ]
}
```

2.10. SERVER Section

In the SERVER section, you can configure connection information (Port, SSL, etc.) that allows WebtoB to receive client requests.

Each protocol that WebtoB can handle must be set in the SERVER section.

WebtoB provides SERVER types such as **'http'**, **'wjp'**, and **'admin'**, each of which can be configured in detail.

2.10.1. Configuration Items

The following is the configuration format of the SERVER section.

```
"server": {
    "http": {
        #"common_config": {                             # COMMON
            #"doc_root": string,                        # "docs/", $ENV, R.PATH
            #"error_document": [string],
            #"enable_keepalive": boolean,               # true
            #"keepalive_timeout": integer,              # 60 (1-3600)
            #"keepalive_max": integer                   # 0 (0-INT_MAX)
            #"keepalive_error_status_code": [integer]
            #"service_order": string,                   # "uri,ext"
            #"access_log": string
            #"headers": [string]
            #"enable_dos_block": boolean                # false
            #"dos_block_table_size": integer            # 3097 (1-INT_MAX)
            #"dos_block_page_count": integer            # 5 (0-INT_MAX)
            #"dos_block_page_interval": integer         # 1 (1-INT_MAX)
            #"dos_block_site_count": integer            # 50 (0-INT_MAX)
            #"dos_block_site_interval": integer         # 1 (1-INT_MAX)
            #"dos_block_period": integer                # 30 (1-INT_MAX)
            #"dos_block_white_list": [string]
            #"url_rewrite_config": string
            #"alias": [string]
            #"index_name": string                       # "index.html"
            #"filters": [string]
            #"rpaf_header": string
            #"enable_directory_index": boolean          # true
        },
        "http_servers": [
            {
                #"common_config": {...},                # COMMON
                "name": string,
                #"port": integer,                       # 80 | 443 (1-65535)
                #"enable_ssl": boolean,                 # false
                #"ssl_name": [string],
                #"idle_timeout": integer                # 300 (1-3600)
                #"initial_connection_timeout": integer  # 10 (0-INT_MAX)
                #"request_header_timeout": integer      # 60 (0-INT_MAX)
                #"request_body_timeout": integer        # 0 (0-INT_MAX)
                #"idle_timeout_status": integer         # 500 (511-599)
                #"vhosts": [
                    {
                        #"common_config": {...},        # COMMON
                        "name": string,
                        "host_name": [string]
                    }
                ]
            }
        ],
        #"mime_type_file": string,                      # "mime.types"
        #"default_mime_type": string                    # "text/html"
    },
    #"wjp": {
        "wjp_servers": [
            {
                "name": string,
                #"min_proc": integer,                   # 1 (1-INT_MAX)
                #"max_proc": integer,
                #"svr_chk_time": integer                # 60 (0-3600)
                #"flow_control": integer                # 50 (1-INT_MAX)
                #"max_jengine_count": integer           # 64 (1-INT_MAX)
                #"ping_timeout_status": integer         # 503 (511-599)
            }
        ],
        #"port": integer,                               # 9900 (1-65535)
        #"enable_ssl": boolean,                         # false
        #"ssl_name": string
    }

    #"admin": {
        #"port": integer                                # 9090 (1-65535)
        #"access_log": string
    }

    #"tcp": {
        "tcp_servers": [
            {
                "name": string
                "port": integer                         # (1-65535)
                "server_address": [string]
                #"idle_timeout": integer                # 300 (0-INT_MAX)
                #"initial_connection_timeout"           # 10 (0-INT_MAX)
            }
        ]
    }
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the SERVER section configuration items.

2.10.1.1. http (Required)

Sets the connection information for processing HTTP requests.

Item	Description
Data Type	Object

2.10.1.2. http/common_config

A common setting for HTTP servers. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object

Item	Description
Priority	The priority of the setting is as follows: 1. vhosts 2. http_servers 3. http

2.10.1.3. http/common_config/doc_root

Specifies the top-level path of the documents served by WebtoB.

Environment variables can be used. If a relative path is set, it will be converted to an absolute path based on $WEBTOB6_HOME_PATH.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"docs"

The following is an example of setting doc_root:

```
  "server": {
    "http": {
      "common_config": {
        "doc_root": "docs",
}
```

If doc_root is set in both common_config and http_servers, the setting in http_servers will take precedence over that in common_config.

2.10.1.4. http/common_config/error_document

Specifies the use of a user specified page instead of the HTTP error page. Configure the names in the ERRORDOCUMENT section.

Item	Description
Data Type	Array(string)
Range	Up to 64 items (within 255 characters)

The following is an example of setting error_document. In this case, the name of the error_document must be written in string format, and the error_document must be specified in the error_document section.

```
"server": {
  "http": {
    "common_config": {
      "error_document": ["error1"]
...
"error_document": {
  "error_document_list": [
    {
      "name": "error1",
      "status": 404,
      "url": "/htmls/not_found.html"
    }
  ]
},
```

2.10.1.5. http/common_config/enable_keepalive

Sets whether to use Keepalive (HTTP persistent connection).

Item	Description
Data Type	Boolean
Default Value	True

The following describes each configuration value.

Value	Description
True	Users can reuse connections to handle multiple requests with a single connection.
False	Users must reconnect the socket every time to process a request.

The following is an example of setting enable_keepalive. To configure keepalive_timeout and keepalive_max, enable_keepalive must be set to true.

```
"server": {
  "http": {
    "common_config": {
      "enable_keepalive": true,
      "keepalive_timeout": 60,
      "keepalive_max":0,
    },
```

2.10.1.6. http/common_config/keepalive_timeout

Specifies the time period during which Keepalive remains active. After the user request has been processed, the connection is disconnected after the specified time.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ 3600
Default Value	60

The following is an example of setting keepalive_timeout.

```
  "server": {
    "http": {
      "common_config": {
        "enable_keepalive": true,
        "keepalive_timeout": 10,
        "keepalive_max":0,
      },
```

The connection time can be checked through Idle_time in ci of wsadmin. When called from a browser, the connection may be terminated before the keepalive_timeout setting value because each browser has its own connection time policy.

```
[wsadmin]>> ci
----------------------------------------------------------------------------------
------------------------------------------------
| HTH-0 : Connection info
|
----------------------------------------------------------------------------------
------------------------------------------------
| No | Server |    Local Address     |   Remote Address     | Remote Type | Ssl |
Status | Request Count | Idle Time | Mapping No |
----------------------------------------------------------------------------------
------------------------------------------------
| 5  | http1  | 192.168.0.1:80 | 192.168.0.51:50965 |   CLIENT    | No  | READY
|       0       |     7     |     -1     |
----------------------------------------------------------------------------------
------------------------------------------------
```

2.10.1.7. http/common_config/keepalive_max

Specifies when to limit the Keepalive request count.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

The following describes each configuration value.

Value	Description
0	No limit on the number of requests.

The following is an example of setting keepalive_max:

```
"server": {
  "http": {
    "common_config": {
      "enable_keepalive": true,
      "keepalive_timeout": 60,
      "keepalive_max":2,
    },
```

The number of connections can be checked through Request Count in ci of wsadmin.

```
[wsadmin]>> ci
---------------------------------------------------------------------------------
------------------------------------------------
| HTH-0 : Connection info
|
---------------------------------------------------------------------------------
------------------------------------------------
| No | Server |     Local Address     |    Remote Address    | Remote Type | Ssl |
Status | Request Count | Idle Time | Mapping No |
---------------------------------------------------------------------------------
------------------------------------------------
| 5  | http1  | 192.168.0.1:80 | 192.168.0.51:50965 |   CLIENT    | No  | READY
|       0       |      7      |      -1      |
---------------------------------------------------------------------------------
------------------------------------------------
```

2.10.1.8. http/common_config/keepalive_error_status_code

Specifies the HTTP status code to use when the user connection is kept alive (persistent connection, keep-alive) after sending an error response.

Item	Description
Data Type	Array (integer)
Range	Up to 64 items (300 - 303 | 305 - 599)

Normally, if WebtoB sends an error response with status code of 3xx (excluding "304 Not Modified"), 4xx, or 5xx, the client connection is terminated. However, if WebtoB sends a status code specified in this option, the client connection is maintained.

The following is an example of WebtoB keeping the connection with client alive after responding with either a "302 Found" or "404 Not Found" message.

```
...
"server": {
  "http": {
    "common_config": {
      "enable_keepalive": true,
      "keepalive_timeout": 60,
      "keepalive_max":0,
      "keepalive_error_status_code": [302, 404],
    },
...
```

2.10.1.9. http/common_config/service_order

Sets the priority between URI and EXT in the SERVICE section when determining the Destination to process the user request.

Item	Description
Data Type	String
Range	"uri,ext" | "ext,uri"
Default Value	"uri,ext"

The following describes each configuration value.

Value	Description
uri,ext	Checks the URI section configuration first. If the configuration does not exist, check the EXT section configuration.
ext,uri	Checks the EXT section configuration first. If the configuration does not exist, check the URI section configuration.

If neither option is specified, the default HTML service (Worker Thread) processes the request.

The following is an example of setting service_order.

```
...
"server": {
  "http": {
    "common_config": {
      "service_order: : "uri,ext"
    },
...
```

2.10.1.10. http/common_config/access_log

Specifies the LOGGING section name corresponding to the access log.

Item	Description
Data Type	String
Range	Up to 255 characters

The following is an example of setting access_log. The access_log specified in the setting must be defined in the access_log entry of the logging section.

```
...
"server": {
  "http": {
    "common_config": {
      "access_log": "access_log1"
    },
    ...
"logging": {
  "access_log": [
    {
      "name": "access_log1",
      "level": "INFO",
      "format": "DEFAULT",
      ...
...
```

2.10.1.11. http/common_config/headers

Sets the name of the HEADERS section to apply.

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

The following is an example of setting headers. The names of the headers are specified in string format in the headers_list of the headers section. Multiple headers can be set.

```
...
"server": {
  "http": {
    "common_config": {
      "headers": ["header1","header2"]
    },
...
"headers": {
  "headers_list": [
    {
      "name": "header1",
      "action": "AppendResponse",
      "field_name": "Content-Type",
      "field_value": "; charset=ISO"
    },
    {
      "name": "header2",
      "action":"AddRequest",
      "field_name":"Accept-Encoding",
      "field_value":"GZIP"
    },
  ]
}
```

2.10.1.12. http/common_config/enable_dos_block

Sets whether to use the DoS attack prevention feature.

Item	Description
Data Type	Boolean
Default Value	False

The following is an example of setting enable_dos_block. Set to true to enable the DoS attack prevention feature.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "enable_dos_block":true,
        "dos_block_table_size":3097,
        "dos_block_site_count":50,
        "dos_block_site_interval":1,
        "dos_block_period":30
        ...
    },
...
```

2.10.1.13. http/common_config/dos_block_table_size

Sets the maximum number of IP addresses to be considered as DoS attacks. Any IP addresses exceeding the set value will be blocked.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	3097

Since the table is managed by HTH, if the HTH value is set to greater than 1, the number of IP addresses managed will be 'dos_block_table_size × hth_count'.

The following is an example of setting dos_block_table_size. Incoming IPs are not managed individually but are managed as objects for each IP. For example, if 192.168.0.1 and 192.168.0.2 come in, each IP is managed as a separate object.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "enable_dos_block":true,
        "dos_block_table_size":3097,
        "dos_block_site_count":50,
        "dos_block_site_interval":1,
        "dos_block_period":30
        ...
    },
...
```

2.10.1.14. http/common_config/dos_block_page_count

Specifies the number of requests to the same page.

If a user IP address requests the same page more than the specified number of times within the dos_block_page_interval, the IP address will be blocked during the time specified in the dos_block_period period.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	5

The following describes each configuration value.

Value	Description
0	Does not check for DoS attacks on the same page.

The following is an example of setting dos_block_page_count. If more requests than the specified count (5) occur within the interval, the corresponding IP address is blocked. The dos_block_page_count is used in conjunction with dos_block_page_interval.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "enable_dos_block":true,
        "dos_block_table_size":1,
        "dos_block_page_count":5,
        "dos_block_page_interval":1,
        "dos_block_period":30,
        ...
    },
...
```

2.10.1.15. http/common_config/dos_block_page_interval

Specifies the period to check for DoS attacks on the same page.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	1

The following is an example of setting dos_block_page_interval. If more requests than the specified count occur within the interval (100 seconds), the corresponding IP address is blocked. The dos_block_page_interval is used in conjunction with dos_block_page_count.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "enable_dos_block":true,
        "dos_block_table_size":1,
        "dos_block_page_count":5,
        "dos_block_page_interval":1,
        "dos_block_period":30,
        ...
    },
...
```

2.10.1.16. http/common_config/dos_block_site_count

Sets the number of requests for the entire site.

If a user’s IP address request any page on the site more than the specified number of times within the dos_block_site_interval, the IP address will be blocked during the time specified in dos_block_period.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	50

The following describes each configuration value.

Value	Description
0	Does not check for DoS attacks on site requests.

2.10.1.17. http/common_config/dos_block_site_interval

Specifies the period to check for DoS attacks on the same site.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	1

2.10.1.18. http/common_config/dos_block_period

Sets the period of time for which specific user IP addresses are blocked.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ INT_MAX
Default Value	30

2.10.1.19. http/common_config/dos_block_white_list

Specifies the IPs to exclude from DoS attack prevention. If a Proxy server is used, specify the server IP.

Item	Description
Data Type	Array (string)
Range	Up to 256 items (within 255 characters)
Default Value	30

The following is an example of dos_block settings. In this example, if more requests than the specified count (3) occur for an IP address with table_size (1) during the interval (100 seconds), the IP address will be blocked for the period (100 seconds). However, IP addresses in the white_list are excluded from the DoS blocking management.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "enable_dos_block":true,
        "dos_block_table_size":1,
        "dos_block_site_count":3,
        "dos_block_site_interval":100,
        "dos_block_period":100,
        "dos_block_white_list":["192.168.0.1"],
        ...
    },
...
```

2.10.1.20. http/common_config/url_rewrite_config

Specifies the path to the configuration file for using the URL rewriting feature.

Item	Description
Data Type	String
Range	Up to 255 characters

The following is an example of url_rewrite_config settings. It places a configuration file written in regular expressions in the given path.

```
...
"server": {
  "http": {
    "common_config": {
        ...
        "url_rewrite_config":"/home/tmax/webtob6/config/rewrite9.conf",
        ...
    },
...
```

2.10.1.21. http/common_config/alias

To change the URI of a specific request to realpath, set the name defined in the ALIAS section.

Item	Description
Data Type	Array (string)

Item	Description
Range	Up to 64 items (within 255 characters)

The following is an example of setting an alias. In this case, when url (/external/path/) is called, it will be called as real_path (/home/tmax/webtob6/docs/).

```
...
"server": {
  "http": {
    "common_config": {
        ...
          "alias": {
            "alias_list":[
              {
                "name":"alias1",
                "url":"/external/path/",
                "real_path":"/home/tmax/webtob6/docs/"
              }
            ]
          }
        ...
    },
...
```

2.10.1.22. http/common_config/index_name

Specifies the index page name for a service directory request. For example, if the request from client is "/somedir/", the path "/somedir/index.html" is serviced.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"index.html"

2.10.1.23. http/common_config/filters

Sets the filters to be applied when the following events occur during request processing.

```
AFTER_SEND_REQUEST, AFTER_SEND_RESPONSE, BEFORE_SEND_REQUEST, BEFORE_SEND_RESPONSE, RECEIVE_REQUEST,
RECEIVE_RESPONSE
```

Item	Description
Data Type	Array (string)
Range	Up to 15 items (within 255 characters)

2.10.1.24. http/common_config/rpaf_header

Sets the remote IP that was changed by a proxy server to the IP of the host that sent the request.

Item	Description
Data Type	String
Range	Up to 255 characters

Adding rpaf_header allows users to specify a header. If a header set with rpaf_header is contained, a remote IP will be changed to the header IP.

The remote IP address specified can be checked in the access.log.

```
"rpaf_header" = "X-Forwarded-For"
```

2.10.1.25. http/common_config/enable_directory_index

Sets whether to use the Directory Index feature when the index_name file does not exist for a directory request.

Item	Description
Data Type	Boolean
Default Value	True

2.10.1.26. http/http_servers (Required)

Sets the connection information for the servers that handle HTTP requests.

Item	Description
Data Type	Array (object)
Range	Up to 100 items

2.10.1.27. http/http_servers/common_config

A common setting for HTTP servers. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object

Item	Description
Priority	The priority of the setting is as follows: 1. vhosts 2. http_servers 3. http

2.10.1.28. http/http_servers/name (Required)

Specifies the name of the HTTP server.

Item	Description
Data Type	String
Range	Up to 31 characters

2.10.1.29. http/http_servers/port

Specifies the service port of the HTTP server that can be accessed by the user.

Item	Description
Data Type	Integer
Range	1 ~ 65535
Default Value	80 or 443 (If SSL is used, the default value is 443.)

2.10.1.30. http/http_servers/enable_ssl

Specifies whether to use the SSL/TLS protocol to connect to HTTP.

Item	Description
Data Type	Boolean
Default Value	False

2.10.1.31. http/http_servers/ssl_name

Sets the SSL section items to apply. This applies only when enable_ssl is set to true.

Item	Description
Data Type	Array (string)
Range	Up to 64 items (within 31 characters)

The following is an example of the ssl_name setting:

```
...
"server": {
  "http": {
    "http_servers": {
        ...
        "enable_ssl": true,
        "ssl_name": ["ssl1"]
        ...
    },
...
```

2.10.1.32. http/http_servers/idle_timeout

Specifies the timeout to read/write data to/from a user socket.

When processing a user request, the socket is closed if the user does not write data to or read data from the socket during the specified time.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ 3600
Default Value	300

2.10.1.33. http/http_servers/initial_connection_timeout

Specifies the timeout for receiving the first request from the user.

This setting closes the socket if the user does not send any requests within the specified time after establishing the TCP connection. This also applies to SSL connections. The socket will be closed if the user does not complete SSL connection within the timeout period or fails to send a request after the connection is established.

Item	Description
Data Type	Integer
Unit	seconds
Range	0 ~ INT_MAX
Default Value	10

The following describes each configuration value.

Value	Description
0	No timeout period is applied.

2.10.1.34. http/http_servers/request_header_timeout

Specifies the amount of time to wait for the body when an HTTP request body exists.

This configuration is applied when a user is sending an HTTP request. If the user does not send the requested HTTP request body during this time, a '408 Request Timeout' error is sent to the user.

Item	Description
Data Type	Integer
Unit	seconds
Range	0 ~ INT_MAX
Default Value	60

The following describes each configuration value.

Value	Description
0	No time limit for sending the HTTP request body.

2.10.1.35. http/http_servers/idle_timeout_status

If an idle timeout occurs, the HTTP status code is returned in the response based on the corresponding setting value.

Item	Description
Data Type	Integer
Range	511 ~ 599
Default Value	500

2.10.1.36. http/http_servers/vhosts

Congirues the virtual host feature that allows the HTTP server to provide different services depending on the requested address.

Item	Description
Data Type	Array (object)
Range	Up to 64 items

The following is an example vhosts configuration:

```
...
"server": {
  "http": {
    "http_servers": {
        ...
        "vhosts":[
          {
            "name":"vhost1",
            "host_name":["www.vh1.com"],
            "common_config":{...}
          },
          {
            "name":"vhost2",
            "host_name":["www.vh2.com"],
            "common_config":{...}
          }
        ],
    }
  }
  ...
},
...
```

2.10.1.37. http/http_servers/vhosts/name (Required)

Sets the name of the virtual host.

Item	Description
Data Type	String
Range	Up to 64 items

2.10.1.38. http/http_servers/vhosts/host_name (Required)

Specifies the host name used to access the virtual host. Each host name must be configured differently to distinguish between virtual hosts.

Item	Description
Data Type	Array (string)
Range	Up to 10 items (within 127 characters)

2.10.1.39. http/mime_type_file

Specifies the path for MIME-Type configuration file that maps MIME-Type and file extensions.

This path can be specified as an absolute path or a relative path to $WEBTOB6_HOME_PATH. If no value is set, the feature is disabled.

Item	Description
Data Type	String
Range	Up to 255 characters
Default Value	"mime.types"

2.10.1.40. http/default_mime_type

Specifies the default Content-Type for documents that cannot decide the MIME-Type.

Item	Description
Data Type	String
Range	Up to 127 characters
Default Value	"text/html"

The following is an example of setting mime_type:

```
...
"server": {
  "http": {
    "http_servers": {
        ...
          "mime_type_file":"mime.types",
          "default_mime_type":"text/html"
        ...
    },
...
```

Since ext has a higher priority than mime_type, you need to remove the ext setting to configure mime_type.

2.10.1.41. wjp

Sets WJP’s connection information for communication with JEUS.

Item	Description
Data Type	Object

2.10.1.42. wjp/port

Sets the service port required for integrating WebtoB with JEUS.

Item	Description
Data Type	Integer
Range	1 ~ 65535
Default Value	9900

If WebtoB opens the port, JEUS connects to the port using the following configuration in the 'domain.xml'.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<domain version="8.0" xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    ...
    <servers>
        <server>
            <web-engine>
                <web-connections>
                    <webtob-connector>
                        <name>webtob1</name>
                        <wjp-version>2</wjp-version>
                        <registration-id>MyGroup</registration-id>
                        <network-address>
                            <port>9900</port>
                        </network-address>
                        <thread-pool>
                            <number>10</number>
                        </thread-pool>
                    </webtob-connector>
                </web-connections>
            </web-engine>
        ...
```

2.10.1.43. wjp/enable_ssl

Specifies the option to use the SSL/TLS protocol for service port.

Item	Description
Data Type	Boolean
Default Value	False

You can set Truststore generated by WebtoB SSL in JEUS to establish a connection.

2.10.1.44. wjp/ssl_name

Sets the SSL section items to apply. This applies only when enable_ssl is set to true.

Item	Description
Data Type	String
Range	Up to 31 characters

2.10.1.45. wjp/wjp_servers (Required)

Set the connection information for the servers that will process WJP.

Item	Description
Data Type	Array (object)
Range	Up to 100 items

The following is an example of setting wjp_servers:

```
...
"server": {
  "wjp": {
    "wjp_servers": [
      {
        "name": "JeusServer1",
        "svr_chk_time": 60
      },
      {
        "name": "JeusServer2",
        "svr_chk_time": 30
      }
    ]

...
```

2.10.1.46. wjp/wjp_servers/name (Required)

Sets the name of the WJP server.

Item	Description
Data Type	String
Range	Up to 15 characters

2.10.1.47. wjp/wjp_servers/svr_chk_time

Specifies the interval at which the connection with JEUS is checked.

For a connection in Ready state without service requests, if there is no response to KeepAlive requests for two consecutive svr_chk_time periods, the connection is recognized as abnormal and is terminated and excluded from service distribution.

Item	Description
Data Type	Integer
Unit	seconds
Range	1 ~ 3600
Default Value	60

2.10.1.48. wjp/wjp_servers/flow_control

Sets the maximum size of the response buffer to control the amount of responses received from the server. A smaller value will cause slower page loading, while larger values results in faster responses.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	50

2.10.1.49. wjp/wjp_servers/max_jengine_count

Sets the maximum number of JEUS servers that can be connected to one server.

Item	Description
Data Type	Integer
Range	1 ~ INT_MAX
Default Value	64

2.10.1.50. wjp/wjp_servers/ping_timeout_status

Sets the HTTP status code to be returned if a PING request to the JSV server times out.

Item	Description
Data Type	Integer
Range	511 ~ 599
Default Value	503

2.10.1.51. admin

Sets the connection information for the WebtoB Admin server. This setting is used when calling the WebtoB API provided by the Admin server.

Item	Description
Data Type	Object

2.10.1.52. admin/port

Sets the port number of the admin server.

Item	Description
Data Type	Integer
Range	1 ~ 65535
Default Value	9090

The port number you set can be checked in the log when running wsadmin.

2.10.1.53. admin/access_log

Specifies the LOGGING section name corresponding to the access log.

Item	Description
Data Type	String
Range	Up to 255 characters

2.10.1.54. tcp

Sets the TCP connection information. The TCP server acts as a proxy that forwards TCP connections from a specific port or IP address to another server. The TCP server never interprets transferred data.

Item	Description
Data Type	Object

2.10.1.55. tcp/tcp_servers (Required)

Sets connection information for servers to process TCP requests.

Item	Description
Data Type	Array(object)
Range	Up to 100 items

2.10.1.56. tcp/tcp_servers/name (Required)

Sets the name of the TCP server.

Item	Description
Data Type	String
Range	Up to 31 characters

2.10.1.57. tcp/tcp_servers/port (Required)

Specifies the service port of the TCP server that can be accessed by the user.

Item	Description
Data Type	Integer
Range	1 ~ 65535

2.10.1.58. tcp/tcp_servers/server_address (Required)

Specifies the pairs of an IP address and a port number of servers that will process client requests.

Item	Description
Data Type	Array (string)
Range	Up to 100 items (within 31 characters)

2.10.1.59. tcp/tcp_servers/idle_timeout

Specifies the timeout to read/write data to/from a user socket.

When processing a user request, the socket is closed if the user does not write data to or read data from the socket during the specified time.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	300

The following describes each configuration value.

Value	Description
0	No time limit for idle state.

2.10.1.60. tcp/tcp_servers/initial_connection_timeout

Specifies the timeout for receiving the first request from the user.

This setting closes the socket if the user does not send any requests within the specified time after establishing the TCP connection.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	10

The following describes each configuration value.

Value	Description
0	No timeout period is applied.

2.10.2. Example

The following is an example of configuring the SERVER section.

```
"server": {
    "http": {
        "common_config": {
            "doc_root": "docs",
            "idle_timeout": 300,
            "service_order": "uri,ext",
            "access_log": "access_log1",
            "keepalive": true,
            "keepalive_timeout": 60
        },
        "http_servers": [
            {
                "name": "http1",
                "port": 8080,
                "enable_ssl": false
            }
        ]
    }

    "wjp": {
        "port": 9900,
        "wjp_servers": [
            {
                "name": "MyGroup1",
                "svr_chk_time": 60
            }
        ]
    }

    "tcp": {
        "tcp_servers": [
            {
                "name": "tcp1",
                "port": 5000,
                "server_address": [
                    "192.168.1.20:5000"
                ]
            }
        ]
    }
}
```

2.11. SERVICE section

Configures the internal server to handle HTTP requests based on the request’s URI and EXT.

2.11.1. Configuration Items

The following is the configuration format of the SERVICE section.

```
"service": {
    "uri": [
        {
            "name": string,
            #"target_http_servers": [string],       # ["*"]
            "match": {
                #"rewrite": string,
                #"type": string,                    # "prefix"
                "target": string,
                #"redirect": string,
                #"redirect_status": integer,        # 302
                #"enable_cache": boolean            # false
            },
            "destination": {
                "type": string,
                "target": string
            },
            #"access": string
        }
    ]
    "ext": [
        {
            "name": string,
            #"target_http_servers": [string],       # ["*"]
            "match": {
                #"type": string,                    # "exact"
                "target": string,
                #"enable_cache": boolean            # false
            },
            "destination": {
                "type": string,
                "target": string
            },
            #"access": string
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the SERVICE section configuration items.

2.11.1.1. uri

The URI section is used to provide different destinations to process requests according to the Uniform Resource Identifier (URI) value of the client request.

For example, if the address http://www.tmax.co.kr/JSV/test is requested, the "/JSV/" URI can be set to jeus.

Item	Description
Data Type	Array (object)

In the following example, the URI section must be configured for <first> or a URI that starts with <first> must be used.

```
http://<hostname>:<port>/<first>/<second>/index.html
```

2.11.1.2. uri/name (Required)

Sets the name of the URI section.

Item	Description
Data Type	String
Range	Up to 31 characters

2.11.1.3. uri/target_http_servers

Specifies the server name to process the request. (Example: "target_http_servers": "http1")

If a specific vhost is to handle the request, configure it as 'server name.vhost name'. (Example: "target_http_servers": "http1.vhost1")

Item	Description
Data Type	Array [string]

Item	Description
Range	Up to 64 items [within 31 characters]
Default Value	["*"]

The following describes each configuration value.

Value	Description
"*"	Applies to all servers.

2.11.1.4. uri/match (Required)

Sets a rule to handle a request if its URI matches a specific pattern.

Item	Description
Data Type	Object

2.11.1.5. uri/match/rewrite

Sets the value to change for uri/match/target.

Item	Description
Data Type	String
Range	Up to 255 characters

2.11.1.6. uri/match/type

Specifies the type of pattern configured for the URI. The matching method used for HTTP Request path varies depending on the pattern type.

Item	Description
Data Type	String
Range	"prefix"
Default Value	"prefix"

The following describes each configuration value.

Value	Description
prefix	The pattern set in uri/match/target is the prefix of the HTTP request URL. (For example, the pattern "/uri/" matches request paths such as "/uri/a/", "/uri/a/b/", "/uri/a/b/c", etc.)

2.11.1.7. uri/match/target (Required)

Specifies the pattern that matches the HTTP request path. Once matched, the URI section configuration is applied to the request.

Item	Description
Data Type	String
Range	Up to 255 characters

2.11.1.8. uri/match/redirect

Maps a request for the specified URI to another URI.

Depending on the value of redirect_status set in the URI section, the value set in redirect is set in the Location header field of the HTTP response and delivered to the user. If the value of redirect_status is omitted, the default value of "302 Found" is used.

Item	Description
Data Type	String
Range	Up to 255 characters

2.11.1.9. uri/match/redirect_status

Specifies the HTTP status code to be used when using redirection.

Item	Description
Data Type	Integer
Range	301 | 302 | 303 | 305 | 407 | 410
Default Value	302

The following describes each configuration value.

Value	Alias	Description
301	permanent	Responds with "301 Moved Permanently".
302	found	Responds with "302 Found".
303	seeother	Responds with "303 See Other".
305	useproxy	Responds with "305 Use Proxy".
307	temp	Responds with "307 Temporary Redirect".
410	gone	Responds with "410 Gone".

2.11.1.10. uri/match/enable_cache

Sets whether the content is cached.

Item	Description
Data Type	Boolean
Default Value	False

2.11.1.11. uri/destination (Required)

Sets the destination of the URI service.

Item	Description
Data Type	Object

2.11.1.12. uri/destination/type (Required)

Sets the type of the destination.

Item	Description
Data Type	String
Range	"HTMLS" | "JEUS" | "REVERSE_PROXY"

2.11.1.13. uri/destination/target (Required)

Sets the target of the destination. It must match the name specified in the DESTINATION section.

Item	Description
Data Type	String
Range	Up to 127 characters

2.11.1.14. uri/match/access

Sets whether to allow requests from a specific IP. The setting value must match the name set in access/access_list/name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.11.1.15. ext

Maps a client request to a specific process based on the requested file extension.

Item	Description
Data Type	Array (object)

WebtoB is already configured with mappings between basic MIME types and processes. However, if additional settings are needed, they can be modified in the relevant section.

2.11.1.16. ext/name (Required)

Sets the name of the EXT section.

Item	Description
Data Type	String
Range	Up to 31 characters

2.11.1.17. ext/target_http_servers

Specifies the server name to process the request. (Example: "target_http_servers": "http1")

If a specific vhost is to handle the request, configure it as 'server name.vhost name'. (Example: "target_http_servers": "http1.vhost1")

Item	Description
Data Type	Array (string)
Range	Up to 64 items (within 31 characters)
Default Value	["*"]

The following describes each configuration value.

Value	Description
"*"	Applies to all servers.

2.11.1.18. ext/match (Required)

Sets a rule to process a file if its extension matches a specific pattern.

Item	Description
Data Type	Object

2.11.1.19. ext/match/type

Sets the type of pattern configured in ext/match/target. The matching method used for HTTP Request path varies depending on the pattern type.

Item	Description
Data Type	String
Range	"exact"
Default Value	"exact"

The following describes each configuration value.

Value	Description
exact	Matches if the set pattern matches the extension. (Example: The pattern "abc" matches only if the extension is "abc". No other extensions will match.)

2.11.1.20. ext/match/target (Required)

Specifies the pattern that matches the HTTP request path. Once matched, the EXT section configuration is applied to the request.

Item	Description
Data Type	String
Range	Up to 127 items

2.11.1.21. ext/match/enable_cache

Sets whether the content is cached.

Item	Description
Data Type	Boolean
Default Value	False

2.11.1.22. ext/destination (Required)

Sets the destination for the Extension service.

Item	Description
Data Type	String

2.11.1.23. ext/destination/type (Required)

Sets the type of the destination.

Item	Description
Data Type	String
Range	"HTMLS" | "JEUS" | "REVERSE_PROXY"

2.11.1.24. ext/destination/target (Required)

Sets the target of the destination. It must match the name specified in the DESTINATION section.

Item	Description
Data Type	String
Range	Up to 127 characters

2.11.1.25. ext/match/access

Sets whether to allow requests from a specific IP. The setting value must match the name set in access/access_list/name.

Item	Description
Data Type	String
Range	Up to 255 characters

2.11.2. Example

The following is an example of configuring the SERVICE section.

```
"service": {
    "uri": [
        {
            "name": "static_uri",
            "target_http_servers": [
                "*"
            ],
            "match": {
                "type": "prefix",
                "target": "/static",
                "rewrite": "/"
            },
            "destination": {
                "type": "HTMLS",
                "target": "htmls1"
            }
        }
    ],
    "ext": [
        {
            "name": "ext1",
            "target_http_servers": [
                "http1.vhost1"
            ],
            "match": {
                "type": "exact",
                "target": "text/html"
            },
            "destination": {
                "type": "HTMLS",
                "target": "html1"
            }
        }
    ]
}
```

2.12. SSL Section

The SSL section configures SSL functions used in WebtoB. SSL service operates according to these settings.

2.12.1. Configuration Items

The following is the configuration format of the SSL section.

```
#"ssl": {
    #"common_config": {                                 # COMMON
        #"verify_depth": integer,                       # 0 (0-INT_MAX)
        #"protocols": [string],                         # ["TLSv1", "TLSv1.1", "TLSv1.2", "TLSv1.3"]
        #"required_ciphers": string,                    # "HIGH:!RSA"
        #"tls13_required_ciphers": string               #
"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256"
    },
    #"ssl_configs": [
        {
            "name": string,
            "certificate_file": string,
            "certificate_key_file": string,
            #"certificate_chain_file": string,
            #"certificate_key_password": string,        # "builtin"
            #"ca_certificate_file": string,
            #"ca_certificate_path": string,
            #"ssl_server_name": [string],
            #"verify_client": integer,                  # 0 (0-3)
            #"renegotiation_level": string,             # "secure"
            #"enable_stapling": boolean,                # false
            #"common_config": {...}                     # COMMON
        }
    ],
    #"proxy_ssl_configs": [
        {
            "name": string,
            #"proxy_certificate_file": string,
            #"proxy_certificate_key_file": string,
            #"proxy_certificate_chain_file": string,
            #"proxy_certificate_key_password": string,  # "builtin"
            #"proxy_ca_certificate_file": string,
            #"proxy_ca_certificate_path": string,
            #"ssl_server_name": [string],
            #"enable_insecure": boolean,                # false
            #"common_config": {...}                     # COMMON
        }
    ]
}
```

Refer to Types of Setting Values and Configuration Methods for more information on symbols and details of the SSL section configuration items.

2.12.1.1. common_config

A common setting for SSL sections. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. "ssl_configs", "proxy_ssl_configs" 2. "ssl"

2.12.1.2. common_config/verify_depth

Specifies the level to trace and validate the chain of CAs for authentication. If verification from a single CA is sufficient, set this value to 1.

Item	Description
Data Type	Integer
Range	0 ~ INT_MAX
Default Value	0

2.12.1.3. common_config/protocols

Specifies the protocols that can be used by the server. This can determine whether to support a specific TLS version. To disable a specific protocol, add a hyphen (-) before the protocol name.

Item	Description
Data Type	Array (string)
Range	1 to 4 items ("TLSv1" | "TLSv1.1" | "TLSv1.2" | "TLSv1.3")
Default Value	["TLSv1", "TLSv1.1", "TLSv1.2", "TLSv1.3"]

'SSLv2' and 'SSLv3' are not supported.

2.12.1.4. common_config/required_ciphers

Specifies the ciphers that can be used by the server. You can also configure support for specific ciphers and SSL/TLS versions.

Item	Description
Data Type	String
Range	Up to 1023 characters
Default Value	"HIGH:!RSA"

Since WebtoB uses OpenSSL, refer to the OpenSSL guide for cipher names.

2.12.1.5. common_config/tls13_required_ciphers

Specifies the TLS 1.3 cipher suites that can be enabled. You can also configure support for specific ciphers and SSL/TLS versions.

Item	Description
Data Type	String
Range	Up to 1023 characters
Default Value	"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256"

Since WebtoB uses OpenSSL, refer to the OpenSSL guide for cipher names.

2.12.1.6. ssl_configs

Settings for when WebtoB operates as an SSL server.

Item	Description
Data Type	Array (object)
Range	Up to 100 items

2.12.1.7. ssl_configs/name (Required)

The name of the SSL server configuration. You must set this 'name' when using SSL server configuration in other sections.

Item	Description
Data Type	String
Range	Up to 31 characters

2.12.1.8. ssl_configs/certificate_file (Required)

Specifies the server certificate encoded in PEM.

The certificate is encoded using DER rules and transmitted on the web in ASCII format. If the certificate is encrypted, you will be asked to enter a passphrase.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.9. ssl_configs/certificate_key_file (Required)

Specifies the private key of the PEM-encoded certificate used on the server.

If the key is not combined with the certificate, use this directive to specify the location of the key. Generally, this file is placed in the WebtoB SSL directory.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.10. ssl_configs/certificate_chain_file

Specifies the path to CA certificates used to create a server certificate chain. For client authentication, the ca_certificate_file or ca_certificate_path item must be set.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.11. ssl_configs/certificate_key_password

Specifies how to retrieve the passphrase for an encrypted private key file when SSL is used.

Item	Description
Data Type	String
Default Value	"builtin"

The following describes each configuration value.

Value	Description
builtin	Asks for a passphrase when starting WebtoB.
exec:<program path>	When starting WebtoB, a program is executed and its output is used as a passphrase. A file that is executed with 'exec' can be a compiled executable or a shell script.
raw:<passphrase>	Uses this passphrase when WebtoB starts up.
file:<passphrase file path>	Uses the passphrase from this file that was generated by the mkpwd tool when WebtoB is started.

The following is an example of configuring the certificate_key_password.

```
"ssl_configs":[{
  "name":"ssl1",
  "certificate_file":"/home/webtob6/ssl/server.crt",
  "certificate_key_file":"/home/webtob6/ssl/server.key",
  "certificate_key_password":"exec:/home/webtob6/ssl/password.sh"
}],
```

2.12.1.12. ssl_configs/ca_certificate_file

Use this directive to verify the server from a single CA (Certificate Authority). The certificate file must be encoded in PEM.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.13. ssl_configs/ca_certificate_path

Specifies the directory where the certificate will be saved. The certificate contains the information required for user authentication and should generally be encoded in PEM format.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.14. ssl_configs/ssl_server_name

Sets the server name that can be used as an alias in SSL.

Item	Description
Data Type	Array (string)
Range	Up to 100 items

2.12.1.15. ssl_configs/verify_client

Sets the authentication level to request from SSL clients.

Item	Description
Data Type	Integer
Range	0 ~ 3
Default Value	0

The following describes each configuration value.

Value	Description
0	No authentication is requested.
1	The user must provide valid authentication information to the server.
2	The user must provide valid authentication information to the server.
3	The user must provide valid authentication information. If the server does not have a certificate, the verification process is not required.

2.12.1.16. ssl_configs/renegotiation_level

Specifies the renegotiation level when SSL is used.

Item	Description
Data Type	String

Item	Description
Range	"secure" | "insecure" | "disable"
Default Value	"secure"

The following describes each configuration value.

Value	Description
secure	Proceeds to renegotiate when the client and web server are secure. Example: RFC5746
insecure	Proceeds to renegotiate although the client and web server are not secure. Example: CVE-2009-3555
disable	Does not renegotiate under any circumstances.

If renegotiation proceeds in an insecure situation, it may be vulnerable to Man in the Middle (MITM) attacks or Denial of Service (DoS) attacks.

2.12.1.17. ssl_configs/enable_stapling

Specifies whether WebtoB SSL operates with Online Certificate Status Protocol (OCSP) stapling.

Item	Description
Data Type	Boolean
Default Value	False

2.12.1.18. ssl_configs/common_config

A common setting for SSL sections. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. "ssl_configs", "proxy_ssl_configs" 2. "ssl"

2.12.1.19. proxy_ssl_configs

Settings for when WebtoB operates as an SSL client. It is used when WebtoB acts as a reverse proxy and performs SSL communication.

Item	Description
Data Type	Array (object)
Range	Up to 100 items

2.12.1.20. proxy_ssl_configs/name (Required)

The name of the SSL client configuration. You must set this 'name' when using SSL in reverse proxy configuration.

Item	Description
Data Type	String
Range	Up to 31 characters

2.12.1.21. proxy_ssl_configs/enable_insecure

Sets whether to allow SSL connections when the internal server’s certificate is invalid.

Item	Description
Data Type	Boolean
Default Value	False

The following describes each configuration value.

Value	Description
true	Allows SSL connections.

2.12.1.22. proxy_ssl_configs/proxy_certificate_file

Specifies the client certificate file encoded in PEM. This must be configured if the internal server requires client authentication.

The certificate is encoded using DER rules and transmitted on the web in ASCII format. If the certificate is encrypted, you will be asked to enter a passphrase.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.23. proxy_ssl_configs/proxy_certificate_key_file

Specifies the private key of the PEM-encoded certificate used for client authentication. This must be configured if the internal server requires client authentication.

If the key is not combined with the certificate, use this directive to specify the location of the key. Generally, this file is placed in the WebtoB SSL directory.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.24. proxy_ssl_configs/proxy_certificate_chain_file

Specifies the path to CA certificates used to create a client certificate chain.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.25. proxy_ssl_configs/proxy_certificate_key_password

Specifies how to retrieve the passphrase for an encrypted private key file when client authentication (proxy_certificate_file, proxy_certificate_key_file) is used in PROXY_SSL.

Item	Description
Data Type	String
Default Value	"builtin"

The following describes each configuration value.

Value	Description
builtin	Asks for a passphrase when starting WebtoB.
exec:<program path>	When starting WebtoB, a program is executed and its output is used as a passphrase. A file that is executed with 'exec' can be a compiled executable or a shell script.
raw:<passphrase>	Uses this passphrase when WebtoB starts up.
file:<passphrase file path>	Uses the passphrase from this file that was generated by the mkpwd tool when WebtoB is started.

2.12.1.26. proxy_ssl_configs/proxy_ca_certificate_file

Use this directive to verify the server from a single CA (Certificate Authority). The certificate file must be encoded in PEM.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.27. proxy_ssl_configs/proxy_ca_certificate_path

Sets the directory where the certificate will be stored. The certificate contains the information required to verify the server’s certificate and should generally be encoded in PEM format.

Item	Description
Data Type	String
Range	Up to 255 characters

2.12.1.28. proxy_ssl_configs/ssl_server_name

Sets the server name that can be used as an alias in SSL.

Item	Description
Data Type	Array (string)
Range	Up to 100 items (within 255 characters)

2.12.1.29. proxy_ssl_configs/common_config

A common setting for SSL sections. If set in the parent item, the settings will be reflected to the child items without the need for additional configuration.

Item	Description
Data Type	Object
Priority	The priority of the setting is as follows: 1. "ssl_configs", "proxy_ssl_configs" 2. "ssl"

2.12.2. Example

The following is an example of configuring the SSL sections:

```
{
    "ssl": {
        "ssl_configs": [
            {
                "name": "ssl1",
                "certificate_file": "server.crt",
                "certificate_key_file": "server_key.crt",
                "certificate_chain_file": "server_chain.crt",
                "certificate_key_password": "builtin",
                "ca_certificate_file": "server_ca.crt",
                "ssl_server_name": [
                    "example.com"
                ],
                "verify_client": 0,
                "renegotiation_level": "secure"
            }
        ],
        "proxy_ssl_configs": [
            {
                "name": "proxy_ssl1",
                "proxy_certificate_file": "server.crt",
                "proxy_certificate_key_file": "server_key.crt",
                "proxy_certificate_chain_file": "server_chain.crt",
                "proxy_certificate_key_password": "builtin",
                "proxy_ca_certificate_file": "internal_server_ca.crt"
            }
        ]
    }
}
```

3. URL Rewrite

3.1. Configuting URLRewrite Feature

To use the URLRewrite feature, configuration related to Condition and Rule must be set in the 'url_rewrite_config' section of common_config in /server/http.

The following is an example of the /server/http configuration to enable the URLRewrite feature:

```
"server": {
    "http": {
        "common_config": {
            "url_rewrite_config": "config/rewrite.conf",
            ...
```

In url_rewrite_config, you can use RewriteCond and RewriteRule settings from Apache’s mod_rewrite feature. However, not all configurations are supported, and some functionalities are restricted.

> The URLRewriteConfig settings described in the guide are based on Apache 2.2’s 'mod_rewrite'. Some of the described features may not function in WebtoB.

3.2. Defining Rewriting Conditions

To configure RewriteCond, specify rewriting conditions as follows. Compare the TestString and CondPattern. If the conditions are met, RewriteCond is changed to the pattern defined in RewriteRule configuration.

```
RewriteCond <TestString> <CondPattern> [flags]
```

3.2.1. TestString Configuration

In TestString, the following reserved words and general strings can be used.

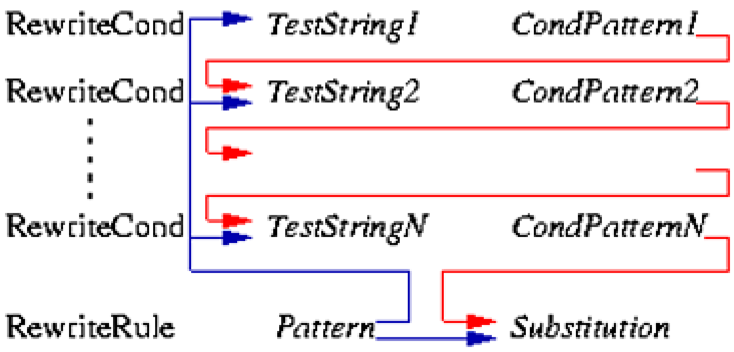
- **$N (0 <= N <= 9)**

  Refer to the pattern wrapped in parentheses among RewriteRule’s patterns.

- **%N (1 <= N <= 9)**

  Refer to the pattern wrapped in parentheses among RewriteCond’s CondPatterns.

  $N and %N have the following Regex Back-Reference structure:



- **%{SERVER_VARIABLE_NAME}**

  Refer to the additional variables among the server environment variables used in CGI.

Type	Variable
Server Environment Variable	%{ENV:variable}: Refers to environment variables, %{HTTP:header}: Refers to HTTP request headers
HTTP headers	HTTP_USER_AGENT, HTTP_REFERER, HTTP_COOKIE, HTTP_FORWARDED, HTTP_HOST, HTTP_PROXY_CONNECTION, HTTP_ACCEPT
Connection & request	REMOTE_ADDR, REMOTE_PORT, REMOTE_METHOD, QUERY_STRING, AUTH_TYPE
Server internals	DOCUMENT_ROOT, SERVER_NAME, SERVER_ADDR, SERVER_PORT, SERVER_PROTOCOL, SERVER_SOFTWARE
Date and time	TIME_YEAR, TIME_MON, TIME_DAY, TIME_HOUR, TIME_SEC, TIME_WDAY, TIME
Specials	THE_REQUEST, REQUEST_URI, HTTPS

The following variables provided by Apache’s mod_rewrite cannot be used.

```
%{SSL:variable}, %{LA-U:variable}, %{LA-F:variable}, REMOTE_HOST,
REMOTE_IDENT, SCRIPT_FILENAME, SCRIPT_USER, SCRIPT_GROUP, SERVER_ADMIN,
API_VERSION, REQUEST_FILENAME, IS_SUBREQ
```

3.2.2. CondPattern Configuration

CondPattern can use Perl compatible regular expressions. The following additional patterns can also be used.

Directive	Description
!Pattern	Specifies the cases that do not match the specified pattern(s).
<CondPattern	Matches when TestString is before CondPattern alphabetically.

Directive	Description
>CondPattern	Matches when TestString is after CondPattern alphabetically.
=CondPattern	Matches when TestString is same as CondPattern.
-d	Matches when TestString is considered a path and it is a directory.
-f	Matches when TestString is considered a path and it is a file.
-s	Matches when TestString is considered a path and the file size is greater than 0.
-l	Matches when TestString is considered a path and it is a symbolic link.
-x	Matches when TestString is considered a path and it is an executable file.
-F	Matches when TestString is considered a path and it is accessible in server configuration.
-U	Matches when TestString is considered a URL and it is accessible in server configuration.

All of these patterns can be prefixed by an exclamation mark '!' to negate their meaning.

The following are the directives used in regular expressions.

Directive	Description
.	Matches any single character.
+	Repeats the previous pattern one or more times.
*	Repeats the previous pattern zero or more times.
?	Optionally matches the previous pattern.
^	Optionally matches the previous pattern.
$	Matches the end of the string.
( )	Groups multiple characters into a single unit, and captures the match results for use in backreferences.
[ ]	A character class - matches one of the characters.
[^]	A negative character class - matches any character not specified.

3.2.3. Flags Configuration

You can change the way patterns are matched by using [flags] in CondPattern.

Directive	Description
nocase|NC	Not case-sensitive in pattern matching.

Directive	Description
ornext|OR	If a different RewriteCond exists after RewriteCond, combine the next RewriteCond with the logical OR operator. If not explicitly specified, the next condition is combined with AND.
novary|NV	If HTTP Header is used as TestString, do not add Vary Header to Response.

The following is an example of using the [OR] flag:

```
RewriteCond %{HTTP_HOST}  ^host1.*  [OR]
RewriteCond %{HTTP_HOST}  ^host2.*  [OR]
RewriteCond %{HTTP_HOST}  ^host3.*
RewriteRule ...
```

If the [OR] flag is not used, each RewriteCond/RewriteRule must be written separately.

3.3. Defining Rewriting Behavior

RewriteRule configures the rewriting of user requests. If a user request matches RewriteCond, the pattern in the user request is replaced with Substitution specified in the RewriteRule setting.

```
RewriteRule <Pattern> <Substitution> [flags]
```

3.3.1. Pattern Configuration

Patterns can use Perl compatible regular expressions, and you can use '!' to specify cases where the pattern does not match.

If '!' is used, the group pattern ($N) enclosed in parentheses cannot be used because unmatched patterns do not have values for group patterns ($N).

If RewriteRule is used alone, pattern matching uses URL-path. If RewriteRule is used with RewriteCond, the final matched pattern is used.

3.3.2. Substitution Configuration

Specifies what to substitute the matched URL with. The following values can be used.

Value	Description
file-system path	When specifying the absolute path of the file system that starts with '/', use the file in the user response. However, the configured path must exist in the file-system.
URL-path	When a general URL path is used, use the appropriate resource.
Absolute URL	If an absolute URL is specified (for example, "http://<hostname>/file.html"), and the hostname matches the server, the scheme and hostname are stripped out and the resulting path is treated as a URL path. Otherwise, it is redirected to an external server.
-	It means no substitution should be performed.
$N (N=0..9)	Indicates the Nth group pattern among RewriteRule's patterns.
%N (N=1..9)	Indicates the Nth group pattern among the last matched RewriteCond patterns.
%{VARNAME}	Refers to VARNAME provided by the server. The items applied to TestString of RewriteCond can be used.
?	The Query string is not changed. In order to change the string, add '?' to the substitute string. In order to delete a query string, place a '?' at the end of the substitute string.

'${mapname:key|default}' cannot be used because WebtoB does not support RewriteMap provided by Apache's mod_rewrite.

3.3.3. Flags Configuration

The detailed behavior of RewriteRule can be configured through the settings in [flags].

Multiple plags can be set using commas (,), and the following values can be used:

- **B**

  By default, backreferences used in Substitution ($N or %N) remove the %-encoding of the URL. However, if the [B] option is used, the URL's %-encoding can be preserved.

  For example, if set in the following, '/C%2b%2b' is mapped to 'index.php?show=/C++' by default. However, if the [B] option is used, it will be mapped to 'index.php?show=/C%2b%2b'.

  ```
  RewriteRule ^(.*)$ index.php?show=$1
  ```

- **chain|C**

  If the current rule does not match, the next rule will not be checked. If the next rule also uses the [C] option, it is also skipped.

- **cookie|CO=NAME:VAL:domain[:lifetime[:path[:secure[:httponly]]]]**

  Add the Set-Cookie Header to a response to add a Cookie to a user’s browser.

- **discardpathinfo|DPI**

  In per-directory context, RewriteRule distinguishes URI and PATH_INFO and combines them each time the rule is applied.

  [DPI] options apply RewriteRule without distinguishing PATH_INFO.

- **env|E=VAR:VAL**

  Adds VAR=VAL to an environment variable. In VAL, regexp backreferences ($N or %N) can be used. This environment variable can be used in SSI or CGI, and can be used as %{ENV:VAR} among RewriteCond patterns.

- **forbidden|F**

  Sends a "403 Forbidden" response.

- **gone|G**

  Sends a "410 Gone" response.

- **handle|H=Content-handler**

  Configures a content-handler.

- **last|L**

  Signifies the end of the rewriting process. This option is similar to the break command in C.

- **next|N**

  Executes the rewriting process again from the start to the changed URL. This option is similar to the continue command in C.

  Be careful when using this option, as it may cause an infinite loop.

- **nocase|NC**

  Turns off case-sensitivity in patterns. For example, 'A' and 'a' are considered the same.

- **noescape|NE**

  Does not use %-encoding for URL in the rewriting process.

- **nosubreq|NS**

  Stops rewriting in the case of an internal request.

- **passthrough|PT**

Uses the result of rewriting from another handler.

- **qsappend|QSA**

  When rewriting a query string, appends to the existing string instead of overwriting it.

- **redirect|R[=code]**

  When the substitute string is an absolute URL, the URL is forcibly redirected even when the hostname matches a server’s host.

  If the code is not specified, 302 Moved Temporarily is used. In the code, the status code can be entered directly or temp (default), permanent, and seeother can be specified.

- **skip|S=num**

  Skips the next 'num' rules. num is a numeric value.

- **type|T=MIME-type**

  Specifies the content-type of the response.

3.4. Examples of URL Rewrite Configuration

The following are examples of URLRewriteConfig settings for different situations that may occur in a web server.

- **Example 1: Basic URL redirection**

  The following is an example configuration that matches a URL pattern like 'www.test.com/' and redirects it to 'www.test.com/rewrite.html'.

  ```
  # url rewrite config - ex1
  RewriteCond %{HTTP_HOST} ^www\.test\.com$       # if {Host} == "www.test.com"
  RewriteRule ^/$ /rewrite.html [L]               # then "/" > "/rewrite.html"
  ```

- **Example 2: Redirecting to an error page if a file does not exist**

  The following is an example configuration that will convert a request like 'www.test.com/temp/xxx.html' to 'www.test.com/temp/temp_error.html' when the file 'xxx.html' is not found in the temp directory.

  ```
  # url rewrite config - ex2
  RewriteCond %{REQUEST_FILENAME} !-f
  # if {Requested file name} != file or the file pointer
  RewriteRule ^/([^/]+) /$1/$1_error.html [L]
  # then "/temp/xxx.html" > "/temp/temp_error.html"
  ```

- **Example 3: Redirecting HTTP requests to HTTPS**

  The following is an example configuration that converts a request like 'http://www.test.com:80' to 'https://www.test.com:443'.

  ```
  # url rewrite config - ex3
  RewriteCond %{HTTP_HOST} ^www\.test\.com$
  # if {Host} == "www.test.com"
  RewriteCond %{SERVER_PORT} 80
  # AND {Port} == "80"
  RewriteRule .* https://www.test.com:443$0 [R]
  # then > "https://www.test.com:443$0"   ($0: request uri)
  ```

- **Example 4: Domain change (ignoring URI)**

  The following is an example of a configuration that matches a URL pattern such as 'www.test.com/xxx.html' and changes it to 'www.test_new.com/', ignoring the request URI.

  ```
  # url rewrite config - ex4
  RewriteCond %{HTTP_HOST} ^www\.test\.com$       # if {Host} == "www.test.com"
  RewriteRule .* http://www.test_new.com [R]       # then > "http://www.test_new.com"
  ```

- **Example 5: Domain change (preserving URI)**

  The following is an example configuration that matches a URL pattern like 'www.test.com/test.html' and changes it to 'www.test_new.com/test.html', preserving the request URI.

  ```
  # url rewrite config - ex5
  RewriteCond %{HTTP_HOST} ^www\.test\.com$
  # if {Host} == "www.test.com"
  RewriteRule .* http://www.test_new.com$0 [R]
  # then > "http://www.test_new.com$0"   ($0: request uri)
  ```

- **Example 6: Forbidden error if no Referer header in POST requests**

  The following is an example configuration that returns a "403 Forbidden" error when the request method is POST and the Referer header does not exist.

  ```
  # url rewrite config - ex6
  RewriteCond %{REQUEST_METHOD} POST              # if {Method} == "POST"
  RewriteCond %{HTTP_REFERER} =""                 # AND {Referer} == ""
  RewriteRule . - [F]                             # then > 403 Forbidden Return
  ```

- **Example 7: URL redirection by subdomain**

  The following is an example of a configuration that processes "www.test.com/aaa/xxx.html" if the request is "aaa.test.com/xxx.html" and processes "www.test.com/bbb/xxx.html" if the request is "bbb.test.com/xxx.html".

```
# url rewrite config - ex8
RewriteCond %{HTTP_HOST} ^(aaa|bbb)\.test\.com
# if {Host} == ("aaa.test.com" OR "bbb.test.com")
RewriteRule .* /%1$0 [L]
# then "/xxx.html" > "/(aaa|bbb)/xxx.html" (%1: Opening parenthesis of RewriteCond, $0: request
uri)
```

- **Example 8: Changing ports based on query string**

  The following is an example of a configuration that rewrites "www.test.com/test.do?query=value1" as "www.test.com:8080/test.do?query=value1" and "www.test.com/test.do?query=value2&.." as "www.test.com:8080/test.do?query=value2&..".

```
# url rewrite config - ex9
RewriteCond %{QUERY_STRING} ^query=value1$ [OR]
# if {QueryString} == "query=value1"
RewriteCond %{QUERY_STRING} ^query=value2&
# OR {QueryString} == "query=value2&.."
RewriteRule .* http://www.test.com:8080$0 [R]
# then > "http://www.test.com:8080$0"   ($0: request uri)
```

- **Example 9: Redirecting requests from specific subdomains to the www domain**

  The following is an example a configuration that processes a request to "aaa.test.com/test.html" (not starting with 'www') by redirecting it to "www.test.com/aaa/test.html".

```
# url rewrite config - ex10
RewriteCond %{HTTP_HOST} !^www\.test\.com$
# if {Host} != www.test.com
RewriteCond %{HTTP_HOST} ^([a-zA-Z0-9]+)\.test.com$
# AND {Host} == "(alphanumeric characters).test.com"
RewriteRule  .* /%1/$0 [L]
# then > "(alphanumeric string)/$0"   (%1: Opening parenthesis of RewriteCond, $0: request uri)
```

- **Example 10: Blocking CSS/JS requests without a Referer header**

  The following is an example of a configuration that processes requests for CSS or JS files without a Referer header by returning a 403 Forbidden error.

```
# url rewrite config - ex11
RewriteCond %{HTTP_REFERER} !.        # if {HTTP_REFERER} == ""
RewriteRule \.(css|js)$ - [F]         # then "*.css|*.js" > 403 Forbidden Return
```

- **Example 11: Converting a specific pattern into a URL query parameter**

  The followsig is an example configuration that matches a URL pattern "/user@somehost.com/"

and changes it to "/req_test.jsp?blogId=user@somehost.com".

```
# url rewrite config - ex12 (REQEUST_URI)
RewriteCond %{REQUEST_URI} /([a-zA-Z0-9_-]+(@([a-zA-Z0-9_-]+).(com|net|co.kr))?)/?$
RewriteRule . /req_test.jsp?blogId=%1 [PT,L]
```

- **Example 12: Redirection based on specific host and URI pattern**

  The following is an example of a configuration that redirects requests when the HTTP request header's host ends with "tmaxsoft.com" and the URL starts with "/redirect/" to "http://www.tmaxsoft.com/redirect.html".

```
# url rewrite config - ex13 (HTTP_HOST)
RewriteCond %{HTTP_HOST} tmaxsoft.com$
RewriteCond %{REQUEST_URI} /redirect/.*$
RewriteRule . http://www.tmaxsoft.com/redirect.html [R,L]
```

4. WebtoB Admin API

WebtoB provides management features through the Admin API using Rest API. A port can be specified in the /server/admin settings, and API communication is performed through that port.

The admin settings in config are configured as follows. In the following example settings, Admin API communication is possible on port 9090.

```
"server":{
    "admin": {
        "port": 9090
    }
}
```

The following describes the APIs provided by the Admin API.

API	Description
/config	Displays environment configuration settings.
/updateconfig	Reloads the config file to dynamically apply settings.
/cliinfo	Checks the connected web browser.
/connectioninfo	Displays connection information.
/svginfo	Checks server group information.
/statinfo	Retrieves statistics about the state of processes, threads, and services.
/restat	Resets statistics for the server process.
/shutdown	Shuts down WebtoB.
/cachelist	Displays information about responses stored in the HTTP response cache.
/cacherefresh	Deletes responses stored in the HTTP response cache.

The main features of WebtoB Admin API are as follows:

- All APIs communicate using the POST method.
- For APIs operating on HTH, if there is no response from HTH for a certain period of time, it will be displayed as 'busy_hth' after a timeout. For example, if HTH-1 and HTH-2 do not respond within the timeout period for the /cliinfo API, the information will be displayed as 'busy_hth' as shown in the following example.

```
{
    "result": {
        "HTH-0": [
            {
                ...
            }
        ]
    },
    "busy_hth": [
        "HTH-1",
        "HTH-2"
    ]
}
```

- If the API operates correctly, it returns HTTP response code 200, otherwise it returns an appropriate response code depending on the situation.

  The following describes common error response codes.

Response Code	Description
400	Failed to validate JSON schema for HTTP body.
401	User authentication failed.
403	No permission to perform actions on the requested API.
404	Requested API not found.
405	Invalid method used for the request.
500	An error occurred during API operation. (The specific error and message differ depending on the API)

- If the response code is not 200, it is commonly structured as follows:

```
{
    "error_code": string,
    "error_message": string
}
```

  For example, a response code of 404 may appear as follows:

```
{
    "error_code": "APIERR_COMMON_0003",
    "error_message": "No such API"
}
```

4.1. Environment Information

4.1.1. /config

Retrieves the environment information of the currently running system. This process allows you to check not only the values specified in the environment settings file but also the values applied by default.

- Usage

```
{
    "path": string
}
```

Option	Description
Path	Specifies the JSON path to the environment configuration file. The path must always start with '/'. If the path is omitted or set to '/', all currently configured settings will be retrieved.

- Example

  The following is an example of checking the NODE section environment settings with the "/config" Admin API. For detailed information of the configuration items in the NODE section, refer to Configuration Items.

```
[Request]
{
    "path": "/node"
}

[Response]
{
    "result": {
        "cache_entry": 128,
        "cache_key": "HOST_URI",
        "cache_max_file_size": 8192,
        "connection_pool_size": 8192,
        "graceful_shutdown_timeout": 5,
        "hth_count": 1,
        "hth_schedule": "RR",
        "limit_request_body_size": 0,
        "limit_request_header_field_count": 100,
        "limit_request_header_field_size": 8190,
        "limit_request_line_size": 8190,
        "listen_backlog": 4096,
        "max_cache_memory_size": 100,
        "name": "webtob-server",
        "system_filters": [],
        "worker_threads": 1
    }
}
```

4.1.2. /updateconfig

Reloads the config file to dynamically apply the settings. However, only the settings specified with the '|dynamic|' key in the schema file will be applied.

- Usage

  This API does not require any separate parameters.

- Example

  The following is an example of updating the settings with the "/updateconfig" Admin API after modifying the config file.

  ```
  [Request]
  { }

  [Response]
  { }
  ```

4.2. Action State Information

4.2.1. /cliinfo

Displays the environment settings, such as the current status, IP address, number of processed requests, of the currently connected client (usually a web browser).

- Usage

  ```
  {
      "http_server": string,
      "virtual_host": string,
      "hth_number": integer
  }
  ```

Option	Description
None	Retrieves information about all clients connected to the server.
http_server	Retrieves information about clients connected to the specified HTTP server.
virtual_host	Retrieves information about the clients connected to the specified virtual host. This option can only be set if http_server is set, and the value must be a virtual host name included in the http_server.
hth_number	Retrieves information about clients connected to the specified HTH. If not specified, information for all HTHs will be retrieved.

- Example

  The following is an example of checking the client information connected to the vhost1 virtual host in HTH 3 using the "/cliinfo" Admin API.

```
[Request]
{
    "http_server": "http1",
    "virtual_host": "vhost1",
    "hth_number": 3
}

[Response]
{
    "result": {
        "HTH-3": [
            {
                "no": 1,
                "idle": 24,
                "local_ipaddr:port": "192.168.15.167:8080",
                "remote_ipaddr:port": "192.168.15.168:48006",
                "request_count": 0,
                "response_count": 0,
                "ssl": false,
                "status": "READY",
                "http_server": "http1",
                "virtual_host": "vhost1"
            },
            {
                "no": 0,
                "idle": 24,
                "local_ipaddr:port": "192.168.15.167:8080",
                "remote_ipaddr:port": "192.168.15.168:48005",
                "request_count": 2,
                "response_count": 2,
                "ssl": false,
                "status": "READY",
                "http_server": "http1",
                "virtual_host": "vhost1"
            }
        ]
    }
}
```

The following describes each output item.

Item	Description
no	The connection number managed internally by WebtoB.
idle	Client idle time with no data exchanges.
local_ipaddr:port, remote_ipaddr:port	IP address and port number of the server and client.
request_count	The number of requests sent by the client.
response_count	The number of responses sent to the client.
ssl	Specifies whether the client is using an SSL connection.

Item	Description
status	Client state within the server. ◦ CONNECTING: Trying to connect. ◦ CONNECTED: Connection established. ◦ READY: Receiving requests from the client. ◦ RUNNING: A client request is being processed on the server. ◦ DISCONNECTING: Attempting to terminate the connection. ◦ DISCONNECTED: Connection terminated.
http_server	The HTTP server that the client connected to.
virtual_host	The last virtual host accessed by the client.

4.2.2. /connectioninfo

Displays the environment settings, such as the current status, IP address, number of processed requests, of the currently connected client (usually a web browser) and server.

- Usage

```
{
    "hth_number": integer
}
```

Option	Description
None	Retrieves information about all clients connected to WebtoB.
hth_number	Retrieves information about clients and servers connected to the specified HTH. If not specified, information for all HTHs is retrieved.

- Example

  The following is an example of checking the client and server information connected to HTH 3 using the "/connectioninfo" Admin API.

```
[Request]
{
    "hth_number": 3
}

[Response]
{
    "result": {
        "HTH-3": [
            {
                "acceptor": "WJP-0",
                "request_count": 1,
                "response_count": 1,
                "counterpart": "",
                "idle": 3,
                "local_ipaddr:port": "127.0.0.1:9999",
                "no": 0,
                "remote_ipaddr:port": "127.0.0.1:44866",
                "remote_type": "JSV",
                "ssl": false,
                "status": "READY",
                "virtual_host": ""
            },
            {
                "acceptor": "http1",
                "request_count": 2,
                "response_count": 2,
                "counterpart": "",
                "idle": 2,
                "local_ipaddr:port": "192.168.15.167:8080",
                "no": 10,
                "remote_ipaddr:port": "192.168.15.168:64223",
                "remote_type": "CLIENT",
                "ssl": false,
                "status": "READY",
                "virtual_host": ""
            },
            {
                "acceptor": "",
                "request_count": 1,
                "response_count": 1,
                "counterpart": "",
                "idle": 2,
                "local_ipaddr:port": "192.168.15.167:60504",
                "no": 11,
                "remote_ipaddr:port": "192.168.15.124:8080",
                "remote_type": "RPROXY",
                "ssl": false,
                "status": "READY",
                "virtual_host": ""
            }
        ]
    }
}
```

The following describes each output item.

Item	Description
no	The connection number managed internally by WebtoB.
idle	Idle time with no data exchanges.
local_ipaddr:port, remote_ipaddr:port	IP address and port number of the server and client.
request_count	The number of requests sent through this connection.
response_count	The number of responses sent through this connection.

Item	Description
ssl	Specifies whether the connection is established using SSL.
status	Client state within the server. ◦ CONNECTING: Trying to connect. ◦ CONNECTED: Connection established. ◦ READY: Receiving requests from the client. ◦ RUNNING: A client request is being processed on the server. ◦ DISCONNECTING: Attempting to terminate the connection. ◦ DISCONNECTED: Connection terminated.
acceptor	The server that this connection is connected to.
virtual_host	The last virtual host accessed by this connection. (Applies only to clients)
counterpart	The number of the counterpart to this connection. If the current connection is a client, the counterpart refers to a server connection, and vice versa. For queued WJP connection, it may be displayed as -1.

4.2.3. /svginfo

Displays information about each server group that is currently running.

- Usage

```
{
    "jeus" : array(string),
    "rproxy" : array(string)
}
```

Option	Description
None	Displays information about all server groups.
jeus	Specifies the JEUS server group name. Including "*" displays information for all server groups.
rproxy	Specifies the reverse proxy group name. Including "*" displays information for all server groups.

- Example

  The following is an example of checking information for all JEUS server groups and the reverse proxy group named rproxy1 using the "/svginfo" Admin API.

```
[Request]
{
    "jeus" : [*],
    "rproxy" : ["rproxy1"]
}

[Response]
"result": [
    {
        "HTH-0": {
            "jeus": [
                {
                    "aqcnt": 0,
                    "count": 0,
                    "cqcnt": 0,
                    "qpcnt": 0,
                    "reqs": 0,
                    "rscnt": 0,
                    "status": "NRDY",
                    "svgname": "MyGroup2"
                },
                {
                    "aqcnt": 0,
                    "count": 0,
                    "cqcnt": 0,
                    "qpcnt": 0,
                    "reqs": 0,
                    "rscnt": 0,
                    "status": "NRDY",
                    "svgname": "MyGroup1"
                }
            ],
            "rproxy": [
                {
                    "aqcnt": 0,
                    "count": 0,
                    "cqcnt": 0,
                    "qpcnt": 0,
                    "reqs": 0,
                    "rscnt": 0,
                    "status": "RDY",
                    "svgname": "rproxy1"
                }
            ]
        }
    }
]
```

The following describes each output item.

Item	Description
aqcnt	Cumulative number of queued requests. (Cumulative cqcnt value)
count	Number of processed requests.
cqcnt	Number of requests in the queue.

Item	Description
qpcnt	Number of queued requests that have been removed from the queue due to request timeout, qp command, etc.
reqs	Number of requests sent to the server.
rscnt	Number of server restarts due to abnormal termination.
status	Client state within the server. ◦ RDY: Request can be processed. There are server processes connected to WebtoB. ◦ NRDY: Request cannot be processed. There are no server processes connected to WebtoB. ◦ BLK: Server is suspended by administrator command. Request cannot be processed.
svgname	Server group name specified in configuration.

4.2.4. /statinfo

Displays statistics on the running processes and services of the currently running server.

You can check dynamic information such as the number of services processed by each server and the number of connections.

- Usage

```
{
    "hth_number" : integer,
    "target":{
        "jeus": array(string),
        "rproxy" : array(string),
        "htmls": boolean
    }
}
```

Option	Description
hth_number	Specifies the HTH number for which to check statistics.
target/jeus	Specifies the JSV server for which to display statistical information. If "*" is included, information for all JSV servers is displayed.
target/rproxy	Specifies the reverse proxy server for which to display server status. If '*' is included, information for all reverse proxy servers is displayed.
target/htmls	Specifies whether to display HTMLS statistical information.

At least one item must be specified under 'target'.

- Example

  The following is an example of checking information for the JEUS server called MyGroup1, the server in the reverse proxy group called rproxy1, and the HTMLS server using the "/statinfo" Admin API.

```
[Request]
{
    "hth_number" : 1,
    "target":{
        "jeus": ["MyGroup1"],
        "rproxy" : ["rproxy1"],
        "htmls": true
    }
}

[Response]
{
    "result": {
        "HTH-0": {
            "jeus": [
                {
                    "average_processed_time": 0.0,
                    "connections": 50,
                    "request_count": 0,
                    "response_count": 0,
                    "server": "amV1c19kb21haW4vc2VydmVyMQ==",
                    "server_group": "MyGroup1",
                    "sticky_routed_count": 0
                }
            ],
            "rproxy": [
                {
                    "average_processed_time": 0.0,
                    "connections": 0,
                    "request_count": 0,
                    "response_count": 0,
                    "server": "192.168.15.167:8090",
                    "server_group": "rproxy1",
                    "sticky_routed_count": 0
                },
                {
                    "average_processed_time": 0.0,
                    "connections": 0,
                    "request_count": 0,
                    "response_count": 0,
                    "server": "192.168.15.167:8088",
                    "server_group": "rproxy1",
                    "sticky_routed_count": 0
                }
            ],
            "htmls": {
                "average_processed_time": 0.000309519,
                "queue_count": 0,
                "request_count": 5,
                "response_count": 5,
                "server": "HTMLS"
            }
        }
    }
}
```

The following describes each output item.

Item	Description
average_processed_time	The average processing time per request. (Unit: seconds)
connections	The number of connections linked to the server.
request_count	The number of requests sent to the process.
response_count	The number of requests processed by the process.
server	The name of a server belonging to the server group. ◦ jeus: jengineid ◦ Reverse proxy: Target server address ◦ HTMLS: HTMLS (fixed)
server_group	The server group name
sticky_routed_count	The number of times the request was routed to the server based on the Sticky ID received from the client.

4.2.5. /restat

Initializes statistics of server processes. Use this command to reset a specific service or server process statistical information.

- Usage

```
{
   "hth_number" : integer,
   "target":{
       "jeus": array(string),
       "rproxy" : array(string),
       "htmls": boolean
   }
}
```

Option	Description
None	Initializes all HTHs.
hth_number	Specifies the HTH number to initialize. If not set, all HTHs are initialized.
target	Specifies the target to initialize. At least one target must be set.
target/jeus	Specifies the JEUS server to initialize. If '*' is included, all JEUS servers are initialized.

Option	Description
target/rproxy	Specifies the reverse proxy servers to initialize. If '*' is included, all reverse proxy servers are initialized.
target/htmls	Specifies whether to initialize HTMLS statistics.

- Example

The following is an example of initializing statistics of a JEUS server named "MyGroup1".

```
[Request]
{
   "target": {
       "jeus": ["MyGroup1"]
   }
}

[Response]
"HTH-0": {
   "jeus": [
       "MyGroup1"
   ]
}
```

Returns the names of the targets that were successfully initialized for each HTH.

4.3. Process Shutdown

4.3.1. /shutdown

Shuts down the running WebtoB.

- Usage

```
{
   "timeout": integer
}
```

Option	Description
"timeout"	Specifies the timeout for the graceful shutdown process in seconds. If WebtoB does not shut down within the specified time, it will be forcibly terminated. If omitted, the value set in /node/graceful_shutdown_timeout is applied.

- Example

The following is an example of calling the "/shutdown" Admin API to forcibly terminate WebtoB if it hasn’t shut down after attempting a graceful shutdown for 30 seconds.

```
[Request]
{
   "timeout": 30
}

[Response]
{
   "result": {}
}
```

The response is returned immediately after WebtoB receives the command. However, the WebtoB process may still be running even after the response is received.

4.4. Cache Information Management

4.4.1. /cachelist

Displays the response information currently stored in WebtoB’s HTTP response cache.

- Usage

```
{
    "hth_number": integer
}
```

Option	Description
None	Displays all cache information.
"hth_number"	Specifies the HTH number to query cache information.

- Example

The following is an example of checking cache information for HTH 0 using the "/cachelist" Admin API. The rest of the request path is used only for internal server debugging.

```
[Request]
{
    "hth_number": 0
}
```

```
[Response]
"result": {
    "HTH-0": {
        "cache_map": [
            {
                "expire_time": "2023-10-25 16:35:18",
                "key": "192.168.15.167:8080/index.html",
                "size": 4312
            },
            {
                "expire_time": "2023-10-25 16:34:52",
                "key": "192.168.15.167:8080/favicon.ico",
                "size": 4507
            },
            {
                "expire_time": "2023-10-25 16:34:52",
                "key": "192.168.15.167:8080/4.html",
                "size": 219
            }
        ],
        "cache_map_size": 3,
        "memory_usage": 9038
    }
}
```

The following describes each output item.

Item	Description
cache_map	Returns a key (URI) and related information managed in the cache.
cache_map/expire_time	The expiration time of the item.
cache_map/key	The URI of the item.
cache_map/size	The total size of the cache item.
cache_map_size	The number of cache items currently managed in the cache.
memory_usage	The total amount of memory used by the cache item content. (Unit: bytes).

4.4.2. /cacherefresh

Removes the response data from the WebtoB HTTP response cache.

- Usage

```
{
   "url": string,
   "ext": string
}
```

Option	Description
None	Clears the entire cache.
url	Specifies the URL of the cache to be removed.
ext	Specifies the extension of the cache to be removed.

Only one of url or ext can be specified.

The following is an example of removing the response for "test.domain.com/test.html" from the cache using the "/cacherefresh" Admin API.

```
[Request]
{
   "url": "test.domain.com/test.html"
}

[Response]
"result": {
   "HTH-0": {
       "removed_count": 1
   }
}
```

The following describes each output item.

Item	Description
removed_count	The number of cache items that were removed.

5. WebtoB Console Tool

WebtoB provides the following tools to manage engine processes and server processes.

- **Admin Tools**

Console Tool	Description
wsadmin	A tool used for the overall WebtoB system management, supporting system information retrieval and administrator tasks.

- **Other Tools**

Console Tool	Description
configValidator	Performs schema validation on WebtoB environment files.
mkpwd	Creates a file to store passwords for SSL certificate keys.

5.1. wsadmin

wsadmin provides a text-based management environment. It waits in the command prompt and executes a command that has been entered.

- Execution

  To run the wsadmin tool, use the **wsadmin** command.

  ```
  $ wsadmin
  ```

  If the wsadmin tool runs successfully, the following message is displayed at the prompt:

  ```
  $$1 [wsadmin]>>
  ```

- Exit

  To exit the wsadmin tool, use the **exit** command.

  ```
  $$3 [wsadmin]>> exit
  ```

The following describes the commands provided by wsadmin.

Command	Abbreviation	Description
help		Displays help for commands available in the wsadmin tool.

Command	Abbreviation	Description
client-info	(cli)	Displays client connection information for WebtoB.
connect		Connects to a specific WebtoB Admin server.
stat-info	(st)	Displays statistics about server and service status.
config	(cfg)	Displays environment settings information.
cache-list	(cachelist)	Displays information about the HTTP response cache.
svg-info	(and)	Displays information about the internal servers processing the client requests.
connection-info	(There)	Displays connection information between WebtoB's clients and internal servers.
exit		Exits the wsadmin tool.

Each command provides the following common options:

Option	Description
[-i <repeat interval>]	Specifies the repetition interval in seconds of the command. (Default: 1 second)
[-r <number of repetitions>]	Specifies the number of repetitions. (Default: 0) If set to 0, it will repeat infinitely.

The following is an example of executing the cli command 2 times with a 2-second interval.

```
wsadmin]>> cli -i 2 -r 2
Command will be executed repeatedly. Count=2. Interval=2(sec).
* Repeat count = 1
-----------------------------------------------------------------------------------------------------
-----
| Client Info : HTH-0
|
-----------------------------------------------------------------------------------------------------
-----
| No | Status | Request Count | Response Count | Idle |    Local Address    |    Remote Address    |
Ssl |
-----------------------------------------------------------------------------------------------------
-----
| 12 | READY  |       1       |       1        |  22  | 192.168.15.120:8080 | 192.168.15.120:53844 |
No  |
-----------------------------------------------------------------------------------------------------
-----


* Repeat count = 2
-----------------------------------------------------------------------------------------------------
-----
| Client Info : HTH-0
|
-----------------------------------------------------------------------------------------------------
-----
```

```
| No | Status | Request Count | Response Count | Idle |    Local Address    |    Remote Address    |
Ssl |
----------------------------------------------------------------------------------------------------
-----
| 12 | READY  |       1       |       1        |  24  | 192.168.15.120:8080 | 192.168.15.120:53844 |
No  |
----------------------------------------------------------------------------------------------------
-----


Repeat done.
```

5.1.1. help

Displays the help for commands available in the wsadmin tool.

- Usage

```
>  help [command]
```

Option	Description
[Command]	Specifies the command for which to output usage information.

- Example

No Options Used

```
[wsadmin]>> help
----------------------------------------------------------------------------------------------
| Help                                                                                       |
----------------------------------------------------------------------------------------------
|  Command Name  |   Command Alias    |                 Command Descriptor                   |
----------------------------------------------------------------------------------------------
| client-info    | cliinfo, cli       | show client info                                     |
| connect        |                    | connect to the api server                            |
| stat-info      | stat, st           | Show statistics info.                                |
|                |                    | jeus, reverser proxy server, html server statistic. |
| config         | cfg                | Config info                                          |
| cache-list     | cachelist          | Cache info list                                      |
| svg-info       | svginfo, sgi, si   | show server group info                               |
| connectioninfo | coninifo, ci, coni | show connection info                                 |
----------------------------------------------------------------------------------------------
| You can use the help command for each individual command.                                  |
----------------------------------------------------------------------------------------------
```

Options Used

```
[wsadmin]>> help st
----------------------------------------------------------------------------------------------------
--------------------------------------
| Help : stat-info( stat, st )
```

```
|
--------------------------------------------------------------------------------------------------
--------------------------------------
| OptionName | IsRequired | HasArgs |                                           Description
|
--------------------------------------------------------------------------------------------------
--------------------------------------
| hth_number |     No     |   Yes   | target hth number
|
| jeus       |     No     |   Yes   | jeus statistic info
|
| rproxy     |     No     |   Yes   | reverse proxy statistic info
|
| htmls      |     No     |   No    | html server statistic info
|
| ---        |    ---     |   ---   | ---
|
| i          |     No     |   Yes   | Default: 1 (second) if a repeat is needed. Interval
specifies the time delay between repeats.   |
| r          |     No     |   Yes   | Default: 0 (unlimited) if a repeat is needed. To limit the
repeats, set a value greater than 0. |
--------------------------------------------------------------------------------------------------
--------------------------------------
| Show statistics info.
|
| jeus, reverser proxy server, html server statistic.
|
--------------------------------------------------------------------------------------------------
--------------------------------------
```

5.1.2. client-info (cli, cliinfo)

Displays the environment settings, such as the current status, IP address, number of processed requests, of the currently connected client (usually a web browser).

- Usage

```
> cli [-vhost <VHOST name>][-hth <HTH number>]
```

Option	Description
[-vhost <VHOST name>]	Specifies the VHOST name for which client information will be displayed.
[-hth <HTH number>]	Specifies the HTH number for which client information will be displayed.

- Example

  Using the cli without options displays the following result. Each entry represents one client.

```
[wsadmin]>> cli
--------------------------------------------------------------------------------------------------
```

```
---------
| Client Info : HTH-0
|
----------------------------------------------------------------------------------------------------
---------
| No | Status | Request Count | Response Count | Idle |    Local Address    |    Remote Address
| Ssl |
----------------------------------------------------------------------------------------------------
---------
| 5  | READY  |       0       |       0        |  4   | 192.168.15.120:8080 |
192.168.15.120:38572 | No  |
----------------------------------------------------------------------------------------------------
---------
```

The following describes each output item.

Item	Description
No	The connection number managed internally by WebtoB.
Status	Client state within the server. ◦ READY: Receiving requests from the client. ◦ RUNNING: A client request is being processed on the server.
Request Count	The number of requests sent by the client.
Response Count	The number of responses received by that client.
Idle	Client idle time with no data exchanges.
Local Address, Remote Address	IP addresses and port numbers of the server and client
Ssl	Specifies whether the client is using an SSL connection.

5.1.3. connect

This command is used to connect to the WebtoB Admin server from wsadmin.

- Usage

```
> connect [-ip <Admin server IP>][-port <Admin server port>]
```

Option	Description
[-ip <Admin server IP>]	Specifies the IP address of the WebtoB Admin server to connect to.
[-port <Admin server port>]	Specifies the port number of the WebtoB Admin server to connect to.

- Example

```
[wsadmin]>> connect -ip localhost -port 9090
Connection established successfully with localhost:9090.
```

5.1.4. stat-info (st, stat)

Provides information about the operational state of the system and internal servers in operation.

You can check dynamic information such as the current status of the internal server, the names of the services being processed, the number of services processed, service status, and the number of requests waiting in the service queue.

- Usage

```
> st [-hth <HTH number>][-jeus <JEUS server group name>][-rproxy <reverse proxy server group
name>][-htmls <HTML server name>]
```

Option	Description
[-hth <HTH number>]	Specifies the HTH number for which internal server statistics will be output.
[-jeus <JEUS server group name>]	Specifies the JEUS server group name for which statistics will be output.
[-rproxy <reverse proxy server group name>]	Specifies the name of the reverse proxy server group for which statistics should be output.
[-htmls <HTML server name>]	Specifies the HTML server name for which statistics will be output.

- Example

```
[wsadmin]>> st
Prints all information. Use options to print specific items ( -jeus, -rproxy, -htmls )
-------------------------------------------------------------------------------------------------
------------------------------------------------
| HTH-0 : Jeus statistic
|
-------------------------------------------------------------------------------------------------
------------------------------------------------
|              Server              | Server Group | Connections | Average Processed Time |
Request Count | Response Count | Sticky Routed Count |
-------------------------------------------------------------------------------------------------
------------------------------------------------
| amV1c19kb21haW4vYWRtaW5TZXJ2ZXI= |   MyGroup1   |     10      |        0.000000        |
0       |       0        |          0          |
-------------------------------------------------------------------------------------------------
------------------------------------------------


-------------------------------------------------------------------------------------------------
-----------------------------------
```

```
| HTH-0 : ReverseProxy statistic
|
--------------------------------------------------------------------------------------------------
-----------------------------------
|       Server        | Server Group | Connections | Average Processed Time | Request Count |
Response Count | Sticky Routed Count |
--------------------------------------------------------------------------------------------------
-----------------------------------
| 192.168.15.120:8088 |   rproxyg1   |      0      |        0.000000        |       0       |
0       |          0          |
--------------------------------------------------------------------------------------------------
-----------------------------------

----------------------------------------------------------------------------------
| HTH-0 : Html server statistic                                                  |
----------------------------------------------------------------------------------
| Server | Queue Count | Average Processed Time | Request Count | Response Count |
----------------------------------------------------------------------------------
| HTMLS  |      0      |        0.002725        |      10       |       10       |
----------------------------------------------------------------------------------
```

The following describes each output item.

- Output items for JEUS server statistics

Item	Description
Server	The name of the JEUS server managed internally.
Server Group	The name of the JEUS server group.
Connections	The number of connections established with the JEUS server.
Average Processed Time	The average processing time. (Unit: seconds)
Request Count	The number of requests sent to the internal server.
Response Count	The number of requests processed by the internal server.
Sticky Routed Count	The number of requests sent to the internal server with a Sticky ID from the client.

- Output items for reverse proxy server statistics

Item	Description
Server	The address of the reverse proxy server.
Server Group	The name of the reverse proxy server group.
Connections	The number of connections established with the reverse proxy server.
Average Processed Time	The average processing time. (Unit: seconds)
Request Count	The number of requests sent to the internal server.
Response Count	The number of requests processed by the internal server.

Item	Description
Sticky Routed Count	The number of requests sent to the internal server with a Sticky ID from the client.

- Output items for HTML server statistics

Item	Description
Server	The HTML server name.
Queue Count	The number of requests waiting in the internal server’s queue.
Average Processed Time	The average processing time. (Unit: seconds)
Request Count	The number of requests sent to the internal server.
Response Count	The number of requests processed by the internal server.

5.1.5. config (cfg)

Displays the environment information of the currently running system. It allows you to check all environment information including default values by nodes, server groups, servers, and services defined in the environment configuration file.

- Usage

```
> config [-path <JSON path>]
```

Option	Description
[-path <JSON path>]	Specifies the JSON path to the configuration file to check. The path must always start with '/'. If the path is omitted or set to '/', all currently set environment information will be retrieved.

- Example

The following is an example of displaying the environment configuration for the NODE section. For detailed information about the cofngiruation items in the NODE section, refer to Configuration Items.

```
[wsadmin]>> cfg -path /node
----------------------------------------------
| Target Config /node                        |
----------------------------------------------
|                   Values                   |
----------------------------------------------
| {                                          |
|   "cache_entry": 128,                      |
|   "cache_key": "HOST_URI",                 |
|   "cache_max_file_size": 8192,             |
|   "connection_pool_size": 8192,            |
|   "graceful_shutdown_timeout": 30,         |
|   "hth_count": 1,                          |
|   "hth_schedule": "RR",                    |
|   "limit_request_body_size": 0,            |
|   "limit_request_header_field_count": 100, |
|   "limit_request_header_field_size": 8190, |
|   "limit_request_line_size": 8190,         |
|   "listen_backlog": 4096,                  |
|   "max_cache_memory_size": 100,            |
|   "name": "ksh",                           |
|   "system_filters": [],                    |
|   "worker_threads": 1                      |
| }                                          |
----------------------------------------------
```

5.1.6. cache-list (cachelist)

Displays information about the responses stored in WebtoB's HTTP response cache.

- Usage

```
> cachelist [-hth <HTH number>]
```

Option	Description
[-hth <HTH number>]	Specifies the HTH for which to retrieve cache information. If not specified, cache information for all HTHs is retrieved.

- Example

  The following is an example of a cached response to the request "/test.txt". The rest of the request path is used only for internal server debugging.

```
[wsadmin]>> cachelist
-------------------------------------------------------------------
| HTH-0: Cache List Info                                          |
-------------------------------------------------------------------
|          Cache key           |    Expired time     | Cache size |
-------------------------------------------------------------------
| 192.168.15.120:8080/test.txt | 2024-10-29 12:34:53 |         286|
-------------------------------------------------------------------
| Cache count : 1                                                 |
| Memory usages : 286                                             |
-------------------------------------------------------------------
```

5.1.7. svg-info (si, sgi, svginfo)

Displays information about the internal server that processes the client's request.

- Usage

```
> si [-jeus <JEUS server group name>][-rproxy <reverse proxy server group name>]
```

Option	Description
[-jeus <JEUS server group name>]	Specifies the JEUS server group name for which information will be output.
[-rproxy <reverse proxy server group name>]	Specifies the name of the reverse proxy server group for which information will be output.

- Example

```
[wsadmin]>> si
--------------------------------------------------------------------
| HTH-0 : Jeus ServerGroup Info                                    |
--------------------------------------------------------------------
| svgname  | status | aqcnt | count | cqcnt | qpcnt | reqs | rscnt |
--------------------------------------------------------------------
| MyGroup2 |  NRDY  |   0   |   0   |   0   |   0   |  0   |   0   |
| MyGroup1 |  RDY   |   0   |   0   |   0   |   0   |  0   |   0   |
--------------------------------------------------------------------


--------------------------------------------------------------------
| HTH-0 : Reverse Proxy ServerGroup Info                           |
--------------------------------------------------------------------
| svgname  | status | aqcnt | count | cqcnt | qpcnt | reqs | rscnt |
--------------------------------------------------------------------
| rproxyg1 |  RDY   |   0   |   0   |   0   |   0   |  0   |   0   |
--------------------------------------------------------------------
```

The following describes each output item.

Item	Description
svgname	The server group name.
status	Client state within the server. ◦ RDY: Request can be processed. There are internal server processes connected to WebtoB. ◦ NRDY: Request cannot be processed. There are no internal server processes connected to WebtoB. ◦ BLK: Server is suspended by administrator command. Request cannot be processed.

Item	Description
aqcnt	Cumulative number of queued requests. (Cumulative cqcnt value)
count	Number of processed requests.
cqcnt	Number of requests in the queue.
qpcnt	Number of queued requests that have been removed from the queue due to request timeout, qp command, etc.
reqs	Number of requests sent to the server.
rscnt	Number of server restarts due to abnormal termination.

5.1.8. connection-info (ci, coni, coninfo)

Displays connection information between WebtoB’s clients and internal servers.

- Usage

```
> ci [-jeus <JEUS server group name>][-rproxy <reverse proxy server group name>]
```

Option	Description
[-jeus <JEUS server group name>]	Specifies the JEUS server group name for which information will be output.
[-rproxy <reverse proxy server group name>]	Specifies the name of the reverse proxy server group for which information will be output.

- Example

```
[wsadmin]>> ci
-------------------------------------------------------------------------------------------------
--------------------------------------------------
| HTH-0 : Connection info
|
-------------------------------------------------------------------------------------------------
--------------------------------------------------
| No | Server |    Local Address    |    Remote Address    | Remote Type | Ssl | Status | Request
Count | Response Count | Idle Time | Mapping No |
-------------------------------------------------------------------------------------------------
--------------------------------------------------
| 0  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40158 |     JSV     | No  | READY  |       1
|      1       |     23     |     -1     |
| 1  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40170 |     JSV     | No  | READY  |       1
|      1       |     13     |     -1     |
| 2  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40186 |     JSV     | No  | READY  |       1
|      1       |     42     |     -1     |
| 3  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40202 |     JSV     | No  | READY  |       1
|      1       |     31     |     -1     |
| 4  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40214 |     JSV     | No  | READY  |       1
|      1       |     26     |     -1     |
| 5  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40220 |     JSV     | No  | READY  |       1
|       1        |    12     |     -1     |
| 6  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40224 |     JSV     | No  | READY  |       0
|       0        |    27     |     -1     |
| 7  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40230 |     JSV     | No  | READY  |       0
|       0        |    27     |     -1     |
| 8  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40234 |     JSV     | No  | READY  |       0
|       0        |    27     |     -1     |
| 9  | WJP-0  | 192.168.15.120:9900 | 192.168.15.120:40240 |     JSV     | No  | READY  |       0
|       0        |    27     |     -1     |
| 21 | http1  | 192.168.15.120:8080 | 192.168.15.120:47098 |   CLIENT    | No  | READY  |       3
|       3        |    12     |     -1     |
-------------------------------------------------------------------------------------------------
--------------------------------------------------
```

The following describes each output item.

Item	Description
No	The connection number managed internally by WebtoB.
Server	The server name managed internally by WebtoB.
Local Address, Remote Address	IP address and port number of the server and client.
Remote Type	The type of connection. ◦ CLIENT: Connection with the client ◦ CLIENT_WEBSOCKET: Websocket connection with the client ◦ RPROXY: Connection with an internal reverse proxy server ◦ RPROXY_PERSISTENT: Keep-Alive connection with an internal reverse proxy server. ◦ RPROXY_WEBSOCKET: Websocket connection with an internal reverse proxy server. ◦ JSV: Connection with an internal JEUS server.
Ssl	Specifies whether the connection is established using SSL.
Status	The status of the connection. ◦ READY: Ready to receive requests from clients. ◦ RUNNING: Processing the client’s request.
Request Count	The number of requests sent through this connection.
Response Count	The number of requests processed through this connection.
Idle Time	Idle time with no data exchanges.

Item	Description
Mapping No	The connection number associated with this connection. If it is a client connection, it displays the server connection number handling the request. If it is a server connection, it displays the client connection number that sent the request. If there is no associated connection, it displays -1.

5.1.9. exit

Exits the wsadmin tool.

- Usage

```
[wsadmin]>> exit
```

5.2. configValidator

The configValidator performs schema validation on WebtoB configuration files.

Before running WebtoB, you can check the verification results for the configuration file using the configValidator tool.

WebtoB's default configuration file is webtob-config.json, and you can change it by setting the environment variable WEBTOB6_CONFIG_FILE_NAME.

- Usage

```
$ configValidator
```

- Example

Validating a Normal WebtoB Configuration File

```
$ configValidator
Config file path: ../config/webtob-config.json
Schema file path: ../config/webtob-config.schema.json
WEBTOB6_HOME_PATH = ../
WEBTOB6_CONFIG_FILE_PATH = ../config/
WEBTOB6_LIBRARY_PATH = ../lib/
WEBTOB6_SSL_PATH = ../ssl/
WEBTOB6_LICENSE_PATH = ../license/
WEBTOB6_SCHEMA_PATH = ../schema/
WEBTOB6_CONFIG_FILE_NAME = webtob-config.json
******* Start config validation *******
1. Opened schema file
```

```
2. Opened config file
3. Config parsing finished
4. Pre-validation process finished
5. Validation using json-schema finished
6. Post-validation process finished
Config file "../config/webtob-config.json" is validated with schema file "../schema/webtob-
config.schema.json"
Success to load config files : webtob-config.json
******* Validation success *******
```

Validating an Abnormal WebtoB Configuration File

```
$ configValidator
Config file path: ../config/webtob-config.json
Schema file path: ../config/webtob-config.schema.json
WEBTOB6_HOME_PATH = ../
WEBTOB6_CONFIG_FILE_PATH = ../config/
WEBTOB6_LIBRARY_PATH = ../lib/
WEBTOB6_SSL_PATH = ../ssl/
WEBTOB6_LICENSE_PATH = ../license/
WEBTOB6_SCHEMA_PATH = ../schema/
WEBTOB6_CONFIG_FILE_NAME = webtob-config.json
******* Start config validation *******
1. Opened schema file
2. Opened config file
3. Config parsing finished
4. Pre-validation process finished
[Exception on json-schema validation][Config validation failed]At /node/hth_count of -1 -
instance is below minimum of 1
```

5.3. mkpwd

mkpwd is a tool to support certificate_key_password in the SSL section.

When an encrypted private key is set in the SSL section, WebtoB prompts you to enter a passphrase every time it starts. To avoid repeatedly entering the passphrase, you can set the certificate_key_password. The mkpwd tool creates a passphrase file that stores the passphrase so that it can be applied to the certificate_key_password.

For more information on using certificate_key_password, refer to Configuration Items.

- Usage

```
$ mkpwd <file path> <SSL name>
```

Options	Description
<file path>	Name of the file used for certificate_key_password. The result of running mkpwd is saved to this file.
<SSL name>	The name set in the SSL section.

- Example

No Options Used

```
$ mkpwd
<< Usage >>
$ mkpwd file_path ssl_name
     file_path: output file for ssl certificate key password
     ssl_name: name of SSL section
```

Speficying file_path and ssl_name

```
$ mkpwd ssl.ppd ssl1
Make password for SSL certificate key password
Enter password: (Enter password)
Successfully Added password for [ssl1] to a file [ssl.ppd].

$ls -al ssl.ppd
-rw-rw-r--  1 webtob webtob 14 Nov 6 12:34 ssl.ppd
```

- Verifying File Contents

```
$ cat ssl.ppd
ssl1 dGVzdA==
```