FastX System Configuration

Introduction

FastX is highly configurable allowing system administrators to fine tune their FastX installations to better integrate with their existing infrastructure. Users with the administrator permission are allowed to access the system configuration in the web server. Access the system configuration by clicking on the gear icon  . Use this document as a reference for the system configuration.

Local System

All local configuration data is contained in this section. Configuration is stored in files on the system in */usr/lib/fastx/var/config*

Database

This section gives basic information about the running FastX database. There are 2 databases by default: local and share. Local is local to this system only. Share is a shared database that syncs across the cluster. As manipulating the database when it is in use can cause issues, it is better to perform database operations offline using the provided command line tools. These tools should be typically avoided if possible.

Compact Database

Run a database compaction. This can free up space on the system. Database compaction is automatically run periodically

Create Archive

Create an archive of the database for backup purposes.

Rebuild Database

Rebuild the entire database

WWW

WWW (Tab)

This section provides for the basic operation and configuration of the web server.
File: /usr/lib/fastx/var/config/www.json

Require web server

Prevents SSH connections to FastX when the web server is down.

Port

Port used to access FastX through a web browser. (Default 3300)

Certificate File

* More information on security, please go to: https://www.starnet.com/xwin32kb/fastx-31-security/.

Path to the certificate file. Relative paths are searched in the following order:

- */usr/lib/*fastx/var/local/certs
- */usr/lib/*fastx/var/certs

Private Key File

Path to the private key file. Relative paths are searched in the following order:

- */usr/lib/*fastx/var/local/certs
- */usr/lib/*fastx/var/certs

Certificate Authority File

Path to certificate authority file. Relative paths are searched in the following order:

- */usr/lib/*fastx/var/local/certs
- */usr/lib/*fastx/var/certs

Pfx File

Path to the pfx file. Relative paths are searched in the following order:

- */usr/lib/*fastx/var/local/certs
- */usr/lib/*fastx/var/certs

Note: Changes take effect when server restarts

Https (Tab)

JSON object containing the low level HTTPS options for the NodeJS server. For configuration options please see
https://nodejs.org/dist/latest-v10.x/docs/api/https.html#https_https_createserver_options_requestlistener

Settings

This section provides basic settings for the system
File: /usr/lib/fastx/var/config/settings.json

Enable logins on this server

Check box to allow users to login directly to this FastX host

Start new sessions on this server

Check box to allow users to start new sessions on this FastX host. Unchecked, users can only reconnect to existing running sessions.

Rebuild Database when server starts

Reduces the size of database (disk space reduction) at start but increases startup time

Disable Sudo Commands

Disables attempts to run sudo commands when logging in or starting sessions

Override Hostname

Change the hostname that the server sends to other cluster members.

Compact Database

Cron time to compact the database. This will remove old revisions and save disk space.

SSH Port

Port used by SSH (Default 22)

Static Data

File: /usr/lib/fastx/var/config/static.json

JSON object of static data to send to the cluster members on server updates. This can be used for example to tag the server in a group, or notify cluster members that this system contains specific applications or a GPU.

Cluster Setup

Cluster Setup is the main configuration portion of Clustering. This file can be copied to other cluster members as a quick way to distribute the configuration.

File: /usr/lib/fastx/var/config/cluster.json

Database Connections (Tab)

A list of urls that will be used to connect to Database Access Points. Every system in the cluster needs to either have urls listed, or be the url that is listed in a different cluster member's configuration. See Setting up a Cluster for more information

Cluster Keys (Tab)

A list of the secret shared keys that the server will use to access a cluster. When the server makes a request to another cluster member, it will create JSON Web Tokens from each key and pass them as HTTP header information. If ANY key is valid, access to the cluster will be granted. See Cluster Keys for more information

Client Verification (Tab)

A custom function for verifying the client SSL certificate for requests made between cluster members. This is for advanced users with specific security concerns.

Advanced (Tab)

Server is External

Mark this server as External. Enable this feature if the server does not have direct HTTP access to the other cluster members (i.e. can't ping it). All other users in the cluster will make an HTTP EventSource request and maintain the connection so the external server can pass data to the other servers.

Enabling this feature adds extra load on this server since it must always maintain a connection to all other servers in the cluster. This feature is useful if you want to have all your Session Servers behind a firewall and make an External Gateway Server as a public access point. See External Server for more information

Enable Server Reservations

Limit the NonDaemon Logins to one user. This feature adds extra checks to make sure there is only one user on the system. See Server Reservations for more information

Disable Direct Client Connections

By default, a client tries to connect directly to the Session Server of the session he wants to connect to. If the direct connection fails, the client then tries to route traffic through the Gateway Server.

An Administrator may want to route all traffic through a Gateway Server first (for example if the Session Server is behind a firewall making the direct connection always fail). Enable this feature to disable direct connections.

Note that Disabling direct connections makes the connection slower because of multiple hops and also adds more load on the Gateway Server.

Require valid SSL certificates for all cluster members

By default cluster members accept self-signed SSL certificates. This allows the potential for a Man-In-The-Middle attack. Enable this feature to require valid SSL certificates

Fatal Error if session connection fails

The running session attempts to make an outgoing connection to the Gateway Server when connecting indirectly. If this fails (typically because of an SSL error), the running session then connects to the local web server to act as a proxy. This proxy method adds an extra hop and is slower, but should always work. Enabling this feature turns off the proxy method and makes the session connection fail early. This is typically only used for debugging purposes.

Use Authorized Keys directory for cluster keys

Every cluster member has a unique public/private key pair that it uses for encrypting messages, verifying tokens, and other various security features. By default the public keys are stored in the shared database. This poses a potential security risk. If the database is compromised a bad server could upload its own public key and receive messages possibly intended for a different system. Enabling this feature tells the cluster to not use the public keys in the shared database, but rather use public keys stored in the Authorized Keys directory.

The extra security for a potential threat of an already compromised system comes with added work on the administrator's side. The administrator needs to add all the public keys of each cluster member to each cluster member's authorized keys directory (it could be a shared directory).

Authorized Keys Directory

The directory that the server will search when looking for authorized keys.

Request Timeout

How long one cluster member has to send a message to another cluster member and get a response before a timeout error occurs

Debug

Print Debug statements to the server log. Debugging prints out a lot of information so it is best to disable them once finished debugging.

API

Api Requests

Logs Api requests

Api Process

Results from the api request

Websocket

WebSocket Connections

Logs websockets connections

WebSocket Messages

Logs messages between the web browser and destination

Cluster

Cluster Communication

Logs communication between cluster nodes

Cluster External Connections

Logs communications for external nodes

Cluster Database

Logs database communication between cluster members

Link

Link Authentication

Logs authentication phase of link connections

Link Messages

Logs communications between link or session process and the web server

Link Netstring

Logs low level communication between link or session process and the web server

Misc

Web Requests

Logs all requests to the server

Log

Log general non essential messages

Errors

Log general non essential errors

OpenID Connect

Logs OpenID Connect communication

SSH

Logs SSH Verbose Connection information

Sessions

This section contains information for creating and starting sessions

Bookmarks

Bookmarks (Tab)

Create and configure sessions that can be started by users on the FastX server.

New Bookmark

General (Tab)

Command

Command used by the bookmark

Name

Name of the bookmark

Window Mode

Single
The entire session environment is rendered in a single window with desired resolution.
Multiple
The session is managed by multiple windows.

Custom Forms

Ordered List of Custom Forms to attach to the bookmark. The server will determine which forms and in which order the user will access the form.

Profile

Choose the desired Profile which the bookmark will configure to.

Tags

Used to group bookmarks. Enter the name of the tag and hit enter to create.

Icon (Tab)

Drag and drop or click to add an icon to add to the bookmark.

Custom Parameters (Tab)

JSON object of custom parameters to add to the bookmark. These can be overridden by the user.

Static Data

JSON object attached to the bookmark for use in filtering bookmarks

Filter (Tab)

Javascript function that can be used to filter out system bookmarks on a per user basis. This function returns a limited set of bookmarks based on your filtering criteria.

```
function (input) {
  return input.bookmarks;
}
```

Input Object

JSON object passed to the function

```
{
    "user": USER_OBJECT,
    "isAdmin": false, /* Is the user an admin */
    "isManager": false, /* Is the user a manager */
    "userGroups": ["group1", "group2" ..., "groupN"], /* Array of user's
Linux User Groups */
    "bookmarks":  [ BOOKMARK1, BOOKMARK2, ... BOOKMARKN], /* Array of
Bookmark_Objects */
}
```

Return

Filtered Array of bookmarks

Profiles

Profiles define the abilities and limitations of the session.

There is a listing of existing Profiles which can be edited or deleted by clicking on the desired profile and then . The Default Profile cannot be deleted.

New Profile

General (Tab)

Name

Name of Profile

Description

Description of Profile

Id

Id of the Profile. Autogenerated if not entered.

Session Settings (Tab)

Enable Client Clipboard

Allow clipboard data to be sent from client to server

Enable Server Clipboard

Allow clipboard data to be sent from server to client

Server Clipboard Max Bytes

Maximum number of bytes that can be sent from server to client per clipboard operation.

Save Xauthority cookies to the $HOME/.Xauthority file

By default Xauthority cookies are stored in their own private location. This is to avoid polluting the $HOME/.Xauthority file with old session cookies. However, some misbehaving applications

only look in the $HOME/.Xauthority file for authorization. Enabling this option puts the cookie in the $HOME/.Xauthority file to enable these applications to function properly

Enable iglx Support

Older OpenGL applications may use different libraries and fail to run properly. Enable this option if your OpenGL application is failing.

Enable X11 TCP Support

Certain applications require systems to listen for incoming connections in order to function properly. Enable this option if your application needs to connect to its X11 DISPLAY.

Enable VirtualGL Detection (experimental)

By default, attempt to run sessions through VirtualGL. Disabling this option will explicitly turn off running the application through the shipped VirtualGL.

VirtualGL Options

Vglrun command line options.

Terminate session after it has been disconnected for a period of time

When all users have disconnected, terminate the session after x minutes. Set to 0 to disable.

Multisurface Support

Multisurface speeds up rendering in multiple window mode and optimizes certain operations in single window mode. However, it may also make a different set of operations slower. Most users should leave this as Audodect (Single window mode: false, Multiple window mode: true).

CPU Percentage

Maximum % of CPU FastX will use for compressing images. Systems with fewer cores should have a lower CPU percentage. Systems with 8 or more cores should use 100%. Autodetect will determine the number of cores on the system and will adjust accordingly.

Extensions (Tab)

Enable Specific ClientComm Extensions on the session. ClientComm extensions are protocol extensions that interact with the client system adding extra functionality. However they effectively "break out" of the sandboxed session environment adding potential security concerns.

Enable Notification Extension

This extension allows a script on the linux system to send a notification to a running session to be displayed to the user (and possibly receive a response)

Enable Url Extension

This extension allows a script on the linux system to fire off a URL that will be opened on the client system

Custom Extensions

Integrators can add their own extensions to be used. Provide a list of extensions that are enabled.

Event Scripts (Tab)

Event Scripts are custom scripts that are fired off when a session event occurs. These scripts are different from Events in that they are executed as the owner of the session process. (Events are executed as the owner of the web server process)

Event Name

Choose created Event to be used with Profile.

With the exception of ‘terminate’, the scripts that are triggered run in the background. This means that if the script is long-running, the session may terminate while the script is still running. In the case of ‘terminate’, the session monitor will not exit, and thus the session directory will not be removed, until the terminate script exits.

Script

Script that will be executed as the owner of the session.

If the script specified is a single line (after removing blank lines at the start and end), it will be run under the shell. This means that special characters, such as file pattern matching, environment variable expansion, and file redirection, will all work.

If the script is multiple lines, then the entire script is put into a script file, which is executed when the trigger happens. This means that any multiple line event script should start with an interpreter, e.g. !/bin/bash.

The following environment variables will be set for an event script:

$DISPLAY, $XAUTHORITY, $FASTX_SESSION_ID, $FASTX_SESSION_DIR, as well as the usual ones (HOME, USER, etc.)

This means that the event script could run an X11 application, such as xmessage or xwininfo.

Static Data (Tab)

User defined JSON Object used for setting static data on the profile. This can be used as a way to filter profiles.

Debug (Tab)

License Diagnostics

Logs licensing communications.

Clipboard

Logs clipboard communication.

Forms

Custom forms can be created and displayed to users when starting a session. Custom forms allow administrators to collect extra information from the user. This information is typically used in conjunction with load balancing and job scheduling to determine how and where to start the session. The form data collected is added to the *params* field when starting the session.

There is a listing of existing Forms, which can be edited, deleted, or set as default by clicking on the desired form and then  .

General (Tab)

Name

Name of Form

HTML Form

The HTML form that will be displayed to the user. FastX uses Bootstrap 4 styling which can be added to keep the look and feel consistent through the form. Valid input will be added to the **params** object of the start data.

Allow unsafe HTML tags in the form

Explicitly allow unsafe HTML in the form

Id

Id of form. Autogenerated if not entered.

Static Data (Tab)

User Defined JSON Object of static data. This can be used when filtering forms.

Events

Events are fired off when some action happens in a session. Use Events to integrate FastX into your current workflow

Api Events (Tab)

A listing of existing Session Events, which can be edited or deleted by clicking on the desired Session Event and then  .

New Session Event

Api

The API call that will fire the event.

Script

Javascript function that will be executed. The script is executed immediately before the event is fired. Throwing an error in the script will make the api call fail with an error. You can use this to prevent users/actions from occurring.

```
function(input) {
   return input;
}
```

Input Object

```
{
   "action":  "API ACTION",
   "token": WEB_TOKEN,
   "login": "username",
   "data": POST_DATA_FROM_API_REQUEST,
   "session": SESSION_OBJECT /* if there is an associated
session */
}
```

Return

Return the Input Object. The input object can be modified in the javascript function. (for example, changing the command in a start script)

Execute on Web Server Update (Tab)

Javascript function that will be executed when the session updates the web server.

```
function(oldSession,newSession) {
  if(!oldSession) {
    //Session Started
  } else if(newSession.terminated) {
    //session has terminated
  }else if(newSession.clientCount > oldSession.clientCount) {
    //client connected
  } else if(newSession.clientCount < oldSession.clientCount) {
    //client disconnected
    if(newSession.clientCount === 0) {
      //all users disconnected
    }
  }
}
```

oldSession

Current Session Object in the Database

newSession

New Session Object to be updated

Session Start

Load Balancing (Tab)

Set of Load Balancing Scripts that will be executed when starting a new session. Load balancing is used to determine on which server in a cluster the session will start.

There is a listing of existing Scripts, which can be edited, deleted, or set start script by clicking on the desired form and then  .

New Script

Script Name

Name of script

Javascript Function

Script written in javascript. Templates are provided for:

- Allow users on a system
- Blacklist user
- Maximum Sessions

See Load Balancing for more information

Scheduling (Tab)

The Javascript function that will be executed to determine if the start command should be launched immediately or started at a later time. If the function returns a falsy value it will start immediately. If the function returns a string, it will interpolate the template string and execute that instead. See Job Scheduling for more information.

Templates provided for the following job schedulers.

- MOAB
- LSF
- SLURM
- Run a random command

Allow HTML in output

By default HTML is escaped. Check this box to allow output to be interrupted by the client as HTML.

Forms (Tab)

Javascript function used to determine which form to send to the user when starting a session.

```
async function(input) {
  // input.data -- input sent from the api call
  // input.forms -- array of form object

  //return a falsy value to continue
  //return a form object to pop up a form
  //return the string 'SERVERLIST' to allow the user to choose from the
available servers (dynamically generated form from the serverList function)
  return;
}
```

Input Object

```
{
        user: USER_LOGIN,
        isAdmin: false,  /* Is user an admin */
        isManager: false, /* Is user a manager */
        userGroups: ["g1", "g2", ... "gN"], /* Array of User's Linux User
Groups */
        data: START_DATA,
        forms: [FORM1, FORM2 ...] /* Array of Form Objects */
}
```

input.user
[See Login Object]

input.data
[See API Start]

Form Object
[See Form Object]

Return

If the value is Falsy, forms have been completed, and continue
If the value is JSON Object, this is the form object that will be returned
If the value is the special string "SERVERLIST", return a form of the filtered server list

Server List (Tab)

Javascript function used to filter out serverIds based on user input. This allows a partial list of all cluster members to be sent to the user for selection.

```
async function(input) {
  // input.data -- input sent from the api call
  // input.servers -- array of server objects
  //return a filtered list of servers for the user specified by username,
or none to disable server selection
  return input.servers;
}
```

Input Object

```
{
```

```
        user: USER_LOGIN,
        isAdmin: false,  /* Is user an admin */
        isManager: false, /* Is user a manager */
        userGroups: ["g1", "g2", ... "gN"], /* Array of User's Linux User
Groups */
        data: START_DATA,
        servers: [SERVER1, SERVER2 ...] /* Array of Server Objects */
}
```

input.user
See Login Object

input.data
See API Start
Note* If there is a password field, it is changed to **********

Server Object
See Server Object

Return

If the value is Falsy, disable Server Selection
Return an array of server objects that you want the user to select from

Users

Configuration for users

Logins

For detailed information about Logins, click here.

Shutdown Link Daemon on Logout

Logging out shuts down link daemon

Maximum age of Api Token

Number of maximum minutes for the age of the api token

Permissions

Set the permissions that a limited user can do on himself and his sessions. If a user is in the Full User Group, all permissions will be set to true.
[See FastX Permissions Guide for more information]

General (Tab)

Admin Groups

Input group name or username and hit enter. Click “x” to delete the group.

Enable Manager Permissions

Enables Manager permissions. If this is disabled, there are no managers.

Manager Groups

Input group name and hit enter. Click “x” to delete the group.

Enable User Permissions

Enables fine tuning of permissions. If this is disabled all users are full users (no restrictions)

Full User Groups

Input username for group and hit enter. Click “x” to delete the group.

Limited Users Permissions (Tab)

Set the permissions that a limited user can do on himself and his sessions. If a user is in the Full User Group, all permissions will be set to true.
[Click here for more information]

Start Actions

- Start Session from Bookmark
- Start Session with Custom Command

Session Actions

- Session Info
- Shortcuts
- Connect
- Disconnect
- Parameters
- Exec

- Log
- Purge
- Terminate

Bookmarks

- System Bookmarks
- User Bookmarks

User Actions

- User

Manager Permissions (Tab)

Set the permissions that a manager can do on a different user.
Click here for information

Session Actions

- Session Info
- Connect
- Disconnect
- Parameters
- Exec
- Log
- Purge
- Terminate

User Actions

- User

Server

- Server

Authentication

This section provides configuration for the authentication methods available in FastX.

SSH

SSH is the default authentication method for FastX.

General (Tab)

Customize login options. Displays the current login look.

Disable SSH Authentication

Checkbox to disable SSH login.

Use Transparent Background

Checkbox to change background to transparent.

Hide Password Field

Checkbox to hide password field.

Disable Public Key Authentication

Checkbox to remove “Use public key authentication” option.

Theme

Choose from the following login themes

- Default
- Light
- Classic
- None

Load Balancing (Tab)

There is a listing of existing Scripts, which can be edited, deleted, or set as login scripts by clicking on the desired form and then  .

New Script

Script Name

Name of the script

Javascript Function

Script written in javascript. Templates are provided for:

- Allow users on a system
- Blacklist user
- Maximum Sessions

[See Load Balancing for more information]

Server List (Tab)

Javascript function used to filter out available server ids when authenticating via SSH

```
async function(input) {
  // input.data -- input sent from the api call
  // input.servers -- array of server objects
  //return a filtered list of servers for the user specified by username,
or none to disable server selection
  return input.servers;
}
```

Input Object

```
{
        data: LOGIN_DATA,
        servers: [SERVER1, SERVER2 ...] /* Array of Server Objects */
}
```

input.data
[See SSH Login]
Note* Password is changed to **********

Server Object
[See Server Object]

LDAP

Set up LDAP to get extra profile information from users using the SSH login.

LDAP Url

Url to the LDAP Server

Bind DN

Bind DN of the LDAP Server

Bind Password

Bind Password of the LDAP Server

Search Base

Search Base of the LDAP when searching for users

Note: Changes take effect when server restarts

OpenId Connect

Enable

Enables OpenID Connect

Redirect URL

URL to redirect to. https://server.example.com:3300/auth/oidc/callback

Issuer

OpenID Connect Issuer. Refer to your OpenId Connect Provider for more information

Client Id

OpenId Connect Client Id. Refer to your OpenId Connect Provider for more information

Client Secret

OpenId Connect Client Secret. Refer to your OpenId Connect Provider for more information

Logout Redirect URL

URL to redirect when user logs out. This should be on the approval list in your provider's configuration. https://server.example.com:3300/auth/oidc/

Note: Changes take effect when server restarts

Customization

Themes

Themes allow you to customize the look and feel of the website to add your own colors, images and taglines. There is a listing of existing Themes, which can be edited, deleted, or used by clicking on the desired form and then  .

New Theme

General (Tab)

Name

Name of Theme

Description

Description of Theme

Branding (Tab)

Logo

Drag and drop logo image or click on current logo to replace

Tagline

Tagline that will be displayed at the browser login screen

Background (Tab)

Preview

Preview of current background

Image

Drag and drop image or click choose file to replace current background

Color

Choose background color. Options include:

- Color
- transparent
- initial
- inherit

Repeat

Choose background image repeat options:

- initial
- repeat
- repeat-x
- repeat-y
- no-repeat

- space
- round
- inherit

Position

Background image position value

Size

Size of background image

Origin

Choose origin of background image:

- initial
- padding-box
- border-box
- content-box
- inherit

Clip

Choose clip of background image

- initial
- padding-box
- order-box
- content-box
- inherit

Attachment

Choose attachment of background image

- initial
- scroll
- fixed
- inherit

Colors (Tab)

Choose to customize the colors for:

- Primary
- Secondary
- Success
- Danger
- Warning
- Info
- Light

- Dark

Custom (Tab)

Custom SASS configuration to include in the theme

Client Settings

Default Keyboard Layout

Language of keyboard

Hide Client Menu

Hides the settings menu. This is useful for integrators who wish to implement their own UI elements.

Hide All Dialogs

Hides any dialogs that may appear (usually on disconnect or error). This is useful for integrators who wish to implement their own UI elements.

Stop Message Propagation to iframe parent window

Disables messages from propagating to parent windows when client is in an iframe

Hide Home button in menu

Hides the home button in the menu

Home button opens in the same window

When clicking the home button, open the url in the same window. This will disconnect the session

Url of the home button

URL location of home button

Assets

Files that can be uploaded to the host by an administrator.

Public (Tab)

Click “Upload Files” to upload assets to a public directory on the host. There is a listing of existing public files, which can be deleted by clicking on the desired file and then 

Private (Tab)

Click “Upload Files” to upload assets to a private directory on the host. There is a listing of existing private files, which can be deleted by clicking on the desired file and then 

System (Tab)

Click “Upload Files” to upload assets to a system directory on the host. There is a listing of existing system files, which can be deleted by clicking on the desired file and then 

Metrics

Shows a list of the Metrics that will be sent on the server update. A metric is a Javascript function that will be executed on every server update. The return value of the Javascript function is the value of the metric. See Metrics for more information.

There is a listing of existing metics, which can be edited or deleted by clicking on the desired metric and then 

New Metric

Metric

Enter name of the metric

Javascript Function

Add the desired metric in javascript

```
function() {
   return true
}
```

The return value of the function will be the value of the metric

Web

General configuration for the website

HTTP Headers

Allows you to send custom HTTP headers on responses

New Http Header

Http Header

Name of http header

Value

Value of http header

Routes

Choose route of http header

- All Routes
- Authentication
- Running Sessions
- Custom

Website Settings

Website (Tab)

Home Page

Choose user's homepage

- Sessions
- Shortcuts

Autoconnect to Bookmark

Autoconnect to a bookmark on login

Default Login Page

Set default login page

- SSH
- OpenID Connect

Show Timed Out Sessions

Show sessions that are in the database regardless of if the web server can communicate with them.

Welcome Page (Tab)

Custom web page that will appear when a user first connects. You can use this to add special messages/warnings etc

Not Found (404) (Tab)

Custom 404 Error Page

Permission Denied (403)

Custom 403 Error Page

Internal Error (500)

Custom 500 Error Page

Tools

Terminal

Launches a terminal that the administrator can use for text based access.

Licenses

Files (Tab)

New License

Activation Key (Tab)

Enter Activation Key to register

License Server (Tab)

- **Hostname**
  Enter hostname or IP of the license server
- **Port**
  Port used by the license server. (Default port 5053)

License File (Tab)

Select “Choose” when registering with a license file

Usage (Tab)

Displays the licenses that are in use. The list is sorted by license file.

System Log

Displays system logs

Export

Exporting the configuration files will create an override.json file in which can be placed in the var/config directory to override the default configuration for FastX 3.
Export the following configuration files:

- Config Defaults
- Bookmarks
- Bookmarks Filter
- Profiles
- Session Start
- Permissions
- HTTP Headers
- Themes
- SSH
- LDAP
- Client Settings
- OpenID Connect
- Events
- Load Balancing
- Metrics
- Jobs Scheduling
- User Profiles
- User Bookmarks
- User Favorites
- User History