
ZooNavigator Documentation

Release 1.0.0

Lubos Kozmon

Apr 27, 2020

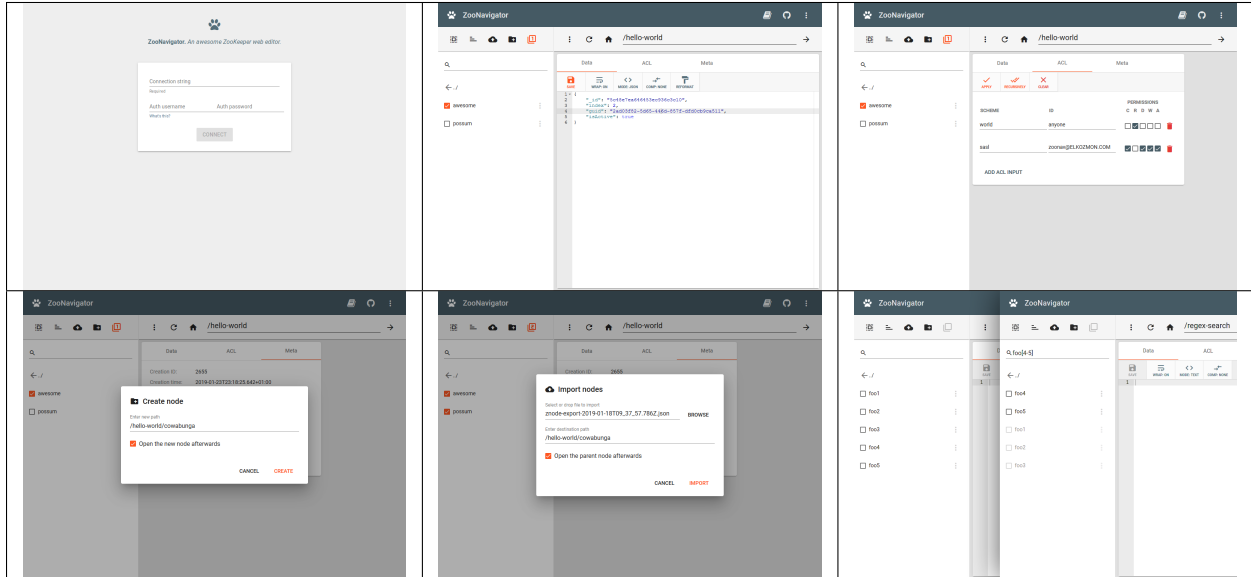
Contents:

1	Quick Start	3
2	Features	5
2.1	Compatibility	5
3	Docker	7
3.1	Configuration	7
3.2	Auto connect	10
3.3	Custom URL	11
4	Development Guide	13
5	Frequently Asked Questions	15
5.1	What should I fill in for Auth username and Auth password?	15
5.2	How to enable Basic Authentication?	15
5.3	I'm getting error that says "Unable to establish connection with ZooKeeper."	15
6	Change Log	17
6.1	0.8.0	17
6.2	0.7.1	17
6.3	0.7.0	18
6.4	0.6.2	18
6.5	0.6.1	18
6.6	0.6.0	18
6.7	0.5.1	19
6.8	0.5.0	19
6.9	0.4.0	19
6.10	0.3.0	20

ZooNavigator is a **web-based ZooKeeper UI and editor/browser** with many features.

It is an open source project and the source code is available on [GitHub](#).

If you like this project, please, consider supporting me by [buying me a beer](#), thanks!



CHAPTER 1

Quick Start

Start Docker container with ZooNavigator Docker image:

```
docker run \  
-d --network host \  
-e HTTP_PORT=9000 \  
--name zoonavigator \  
--restart unless-stopped \  
elkozmon/zoonavigator:latest
```

Go to <http://localhost:9000>.

Windows and Mac users: If wanting to access ZooKeeper running locally on host machine (not in Docker container), follow this advice.

The main goal of this project is to provide a way to efficiently manage ZNodes. This is a brief summary of what you get:

- **ZNode operations**
 - move, copy & paste,
 - export & import,
 - mass delete,
 - regex search tool (currently limited to children nodes)
- **ZNode data editing**
 - data validation, syntax highlighting and auto-formatting
 - edit binary data in base64 mode
 - edit compressed data (gzip)
- **Auth & security**
 - SASL and Auth scheme ZooKeeper authentication
 - ZNode ACL management (supports recursive changes)

2.1 Compatibility

ZooKeeper versions 3.4.x and 3.5.x are currently supported.

ZooNavigator ships as a [Docker image](#).

The following pages explain how to configure your Docker deployment.

3.1 Configuration

ZooNavigator's Docker image can be configured using **environment variables**.

Configuration options could be split into three groups:

- *Server* - configures the web server that serves the application
- *Java* - configures the Java Virtual Machine
- *ZooKeeper client* - configuration related to ZooKeeper

3.1.1 Server

HTTP_PORT

default: 9000

Tells the HTTP server which port to bind to. To disable HTTP set this variable to `disabled`.

HTTPS_PORT

If set, HTTPS server will bind to this port.

SSL_KEYSTORE_PATH

The path to the keystore containing the private key and certificate, if not provided generates a keystore for you.

SSL_KEYSTORE_PASSWORD

The password to the keystore, **defaults to a blank password**.

SSL_KEYSTORE_TYPE

default: JKS

The key store type.

SESSION_TIMEOUT_MILLIS

default: 900000

Sets session inactivity timeout for users. This value is in milliseconds.

SECRET_KEY

Secret key for Play Framework - used for signing session cookies and CSRF tokens. Defaults to 64 random characters generated from */dev/urandom*.

AUTO_CONNECT_CONNECTION_STRING

If set, ZooNavigator will use this value as a default connection string and skip the connect form, automatically connecting to listed ZooKeeper servers.

AUTO_CONNECT_AUTH_INFO

In addition to presetting connection string as explained above, you can also set *Auth info* which ZooNavigator will use to authenticate with ZooKeeper during the auto-connect. **Use semicolon (;) to separate multiple entries.**

3.1.2 Java

JAVA_OPTS

Custom Java arguments.

JAVA_XMS

Sets initial Java heap size. This value is in bytes if no unit is specified.

JAVA_XMX

Sets maximum Java heap size. This value is in bytes if no unit is specified.

JAVA_JAAS_LOGIN_CONFIG

Path to JAAS login configuration file to use.

JAVA_KRB5_DEBUG

If set to `true`, enables debugging mode and detailed logging for Kerberos.

JAVA_KRB5_REALM

Sets the default Kerberos realm.

JAVA_KRB5_KDC

Sets the default Kerberos KDC.

3.1.3 ZooKeeper client

ZK_CLIENT_TIMEOUT_MILLIS

default: 5000

Sets inactivity timeout for ZooKeeper client. If user doesn't make any request during this period ZooKeeper connection will be closed and recreated for the future request if any. **Note that user does not get logged out unlike in event of session timeout.** This value is in milliseconds.

ZK_CONNECT_TIMEOUT_MILLIS

default: 5000

Sets timeout for attempt to establish connection with ZooKeeper. This value is in milliseconds.

ZK_SASL_CLIENT

default: `true`

Set the value to `false` to disable SASL authentication.

ZK_SASL_CLIENT_CONFIG

default: `Client`

Specifies the context key in the JAAS login file.

ZK_SASL_CLIENT_USERNAME

default: zookeeper

Specifies the primary part of the server principal. [Learn more here.](#)

ZK_SERVER_REALM

Realm part of the server principal.

By default it is the client principal realm.

ZK_CLIENT_SECURE

If you want to connect to the server secure client port, you need to set this property to `true`. This will connect to server using SSL with specified credentials.

Note that it requires using the Netty client.

ZK_CLIENT_CNXX_SOCKET

default: org.apache.zookeeper.ClientCnxnSocketNIO

Specifies which ClientCnxnSocket to be used. If you want to connect to server's secure client port, you need to set this property to `:org.apache.zookeeper.ClientCnxnSocketNetty`.

ZK_SSL_KEYSTORE_PATH

Specifies the file path to a JKS containing the local credentials to be used for SSL connections.

ZK_SSL_KEYSTORE_PASSWORD

Specifies the password to a JKS containing the local credentials to be used for SSL connections.

ZK_SSL_TRUSTSTORE_PATH

Specifies the file path to a JKS containing the remote credentials to be used for SSL connections.

ZK_SSL_TRUSTSTORE_PASSWORD

Specifies the password to a JKS containing the remote credentials to be used for SSL connections.

3.2 Auto connect

Auto connect is a feature that lets you skip the [Connect form](#) where you enter ZooKeeper server address and throws you directly into the [Editor UI](#). This is useful when you only have a single (or preferred) ZooKeeper cluster.

To enable this feature simply add these environment variables to ZooNavigator's Web Docker container:

- `AUTO_CONNECT_CONNECTION_STRING`

- `AUTO_CONNECT_AUTH_INFO` (this is optional)

See *Configuration* for description on these environment variables.

3.3 Custom URL

By default ZooNavigator will be available on root of your domain, e.g. `https://www.example.com/`

If you need to deploy ZooNavigator with custom URL using a suffix like `/zoonavigator`, you will have to build ZooNavigator's Docker image yourself.

There is a Docker build argument for that called `BASE_HREF` in ZooNavigator's Dockerfile. You will need to clone its [Git repository](#) and from within that directory build your own Docker image like so:

```
git submodule init

docker build \
  -t my-zoonavigator \
  -f build/docker/Dockerfile \
  --build-arg BASE_HREF=/zoonavigator/ \
  .
```

Note: Note that `BASE_HREF` has to start and end with forward slash /

CHAPTER 4

Development Guide

Development setup requires [NPM](#) and [SBT](#) to be installed on your machine.

ZooNavigator API handles communication with ZooKeeper while ZooNavigator Web is the front-end which in turn talks to the ZooNavigator API.

First, let's get ZooNavigator API running:

```
git clone https://github.com/elkozmon/zoonavigator-api.git
cd zoonavigator-api
sbt play/run
```

Play HTTP server should now be listening on port 9000. The default port is defined in `play/conf/application.conf`.

Next step is to start up ZooNavigator Web:

```
git clone https://github.com/elkozmon/zoonavigator-web.git
cd zoonavigator-web
npm install -g @angular/cli
npm install
npm run dev
```

This starts up Angular development server with proxy configured in `proxy.conf.json` that will forward API requests to <http://localhost:9000> where ZooNavigator API should be running.

Once you've done all that open up <http://localhost:4200> in your browser and you should see the [Connect form](#).

Frequently Asked Questions

5.1 What should I fill in for Auth username and Auth password?

Auth username and password are optional, so feel free to leave those empty. Every username & password pair entered (you can enter many) serves as an extra ACL identity, so ZNodes can be given some permissions only for users logged in with certain credentials.

5.2 How to enable Basic Authentication?

At the moment there is no configuration option to enable Basic Auth out-of-the-box.

I suggest placing a proxy which supports Basic Auth in front of ZooNavigator. For example [Traefik](#).

5.3 I'm getting error that says "Unable to establish connection with ZooKeeper."

ZooNavigator cannot find ZooKeeper host(s) using the connection string you entered in the connect form.

Windows and Mac users: If you're trying to access ZooKeeper running locally on host machine (not in Docker container), [follow this advice](#).

6.1 0.8.0

February 15, 2020

Changes:

- Gzip exports (json imports are still supported)
- Enable Docker image to run under arbitrary user
- When using node filter, show non-matching nodes as semi-transparent
- Clear node filter when navigating to other node
- PUT `/api/znode/data` API endpoint accepts `application/json` instead of `text/plain`
- Add Donation link

Fixes:

- Fix SSL support for ZooKeeper 3.5.x
- Fix false warnings about discarding changes

6.2 0.7.1

November 08, 2019

Minor bug fixes.

Changes:

- Increase default ZNode import size limit to 10mb

Fixes:

- Fix `BASE_HREF` Docker build argument

- Fix app freezing after session expired
- Fix Docker healthcheck when `HTTP_PORT` is set to `disabled`

6.3 0.7.0

September 18, 2019

Release ZooNavigator as a single Docker image.

Changes:

- Single Docker image release
- Increase default session timeout to 1 hour

Fixes:

- Fix save button keyboard shortcut in editor

6.4 0.6.2

June 20, 2019

Allows to run ZooNavigator API's Docker image as an unprivileged user.

Changes:

- Docker container set to run under user `zoonavigator-api`

6.5 0.6.1

April 13, 2019

Allows to run ZooNavigator Web's Docker image as an unprivileged user.

Changes:

- Switched from official *Nginx* Docker image to *Nginxinc*

6.6 0.6.0

January 24, 2019

Implements several feature requests and fixes minor bugs.

Features:

- Export & import ZNode trees
- Read & write gzip compressed ZNode data
- Base64 editor mode for editing binary data
- Custom URLs using added `BASE_HREF` Dockerfile build argument

Changes:

- Turned off access logs on Dockers healthchecks

6.7 0.5.1

November 01, 2018

Decreases Dockers health-check interval so it doesn't cause startup delays in tools like Docker Swarm.

Changes:

- Decreases Dockers health-check interval to 30 seconds

6.8 0.5.0

May 29, 2018

Adds support for SASL authentication with ZooKeeper and HTTPS.

Features:

- SASL authentication with ZooKeeper
- HTTPS support

Changes:

- Changed editors font to Fira Code
- Base Docker images on Alpine
- Enable applying ACLs recursively without making any changes

Fixes:

- Prevent submitting editor via keyboard shortcut when save button is disabled

6.9 0.4.0

April 02, 2018

Adds auto-format feature and fixes some minor bugs.

Features:

- Auto-format feature in editor
- Remember editor options (wrap, mode) for each ZNode
- Log ZooNavigator version on Docker startup

Changes:

- Editor buttons got minor polishing
- Changed editors font to DejaVu Sans Mono

Fixes:

- Switching editor tabs between changes (to data or acl) causes 'Bad version' error
- When session expires two 'Session lost' dialogs spawn at once

6.10 0.3.0

January 11, 2018

Adds many new ZNode editor features.

Features:

- Move ZNode feature
- Duplicate ZNode feature
- Children ZNode list sorting
- Data editor text wrapping
- Address bar for manual navigation
- Auto connect feature - skip connect form and use preconfigured connection parameters
- Hotkey for submitting ZNode data changes (ctrl+s)
- JVM memory settings via environment variables (API Docker image)

Changes:

- UI changes, redesigned editor tabs