ZooNavigator Documentation

Release 2.0.0

Lubos Kozmon

Jun 08, 2026

CONTENTS

ZooNavigator is a **feature-rich web interface for ZooKeeper**.

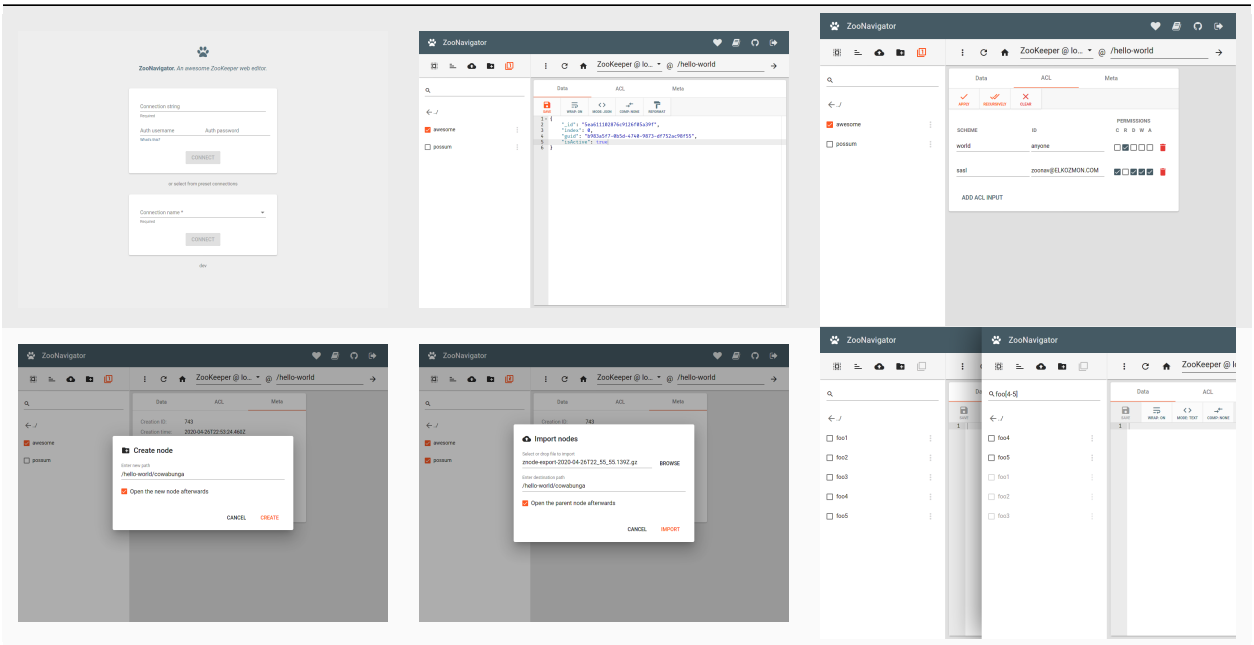
The source code is licensed under AGPLv3 and is available on GitHub.

If you like this project, please, consider supporting me by buying me a beer, thanks!

CHAPTER

ONE

SCREENSHOTS



1.1 Quick Start

1. **Run ZooNavigator** via *Docker* or *Snap*
2. **Open your browser** and go to http://localhost:9000
3. **Enter connection details:**
   - **Connection String:** Your ZooKeeper ensemble (e.g., `zk1:2181,zk2:2181,zk3:2181`)
   - **Auth Info:** Optional, feel free to leave empty
4. **Click Connect**

1.1.1 Docker

Run the ZooNavigator Docker image from Docker Hub:

```
docker run \
  -d \
```

(continues on next page)

(continued from previous page)

```
  -p 9000:9000 \
  -e HTTP_PORT=9000 \
  --name zoonavigator \
  --restart unless-stopped \
  elkozmon/zoonavigator
```

Note

If you want to access ZooKeeper running locally on host machine (not in Docker container):

- **Linux** users may use `--net host` instead of exposing the port
- **Windows and Mac** users should follow this advice

1.1.2 Snap

Install ZooNavigator from Snap store:

```
sudo snap install zoonavigator
```

Next Steps

- Explore the *Features* ZooNavigator has to offer
- Learn how to configure your *Docker* or *Snap* deployments

1.2 Features

The main goal of this project is to provide a way to efficiently manage ZNodes.

This is a brief summary of what you get:

- **Manage multiple ZooKeeper clusters at once**
- **Advanced ZNode operations**
  - move, copy & paste
  - export & import
  - mass delete
  - regex search tool (currently limited to children nodes)
- **Smart ZNode data editor**
  - 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)

1.2.1 Compatibility

ZooKeeper versions 3.5.x to 3.9.x are currently supported.

1.3 Docker

ZooNavigator Docker image is available at Docker Hub.

The following pages explain how to configure your Docker deployment.

1.3.1 Configuration

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

Configuration options could be split into three groups:

- *ZooNavigator* - configures ZooNavigator and the web server
- *ZooKeeper client* - configuration related to ZooKeeper
- *Java* - configures the Java Virtual Machine

ZooNavigator

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.

SECRET_KEY

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

BASE_HREF

default: */*

Sets base URL where ZooNavigator will be served. If you want ZooNavigator to be available at 'http://www.your-domain.com/zoonavigator' instead of 'http://www.your-domain.com' set this variable to */zoonavigator*.

> **Note**
>
> base href must start with '/'

REQUEST_TIMEOUT_MILLIS

default: `10000`

Sets timeout for ZooNavigator requests. This value is in milliseconds.

REQUEST_MAX_SIZE_KB

default: `10000`

Sets maximum request size. Important for large ZNode imports. This value is in kilobytes.

CONNECTION_<MYZK>_NAME

Optional name for preset ZooKeeper connection *'<MYZK>'*

> **Note**
>
> environment variable name should consist only of uppercase letters, digits and underscores.

CONNECTION_<MYZK>_CONN

Connection string for preset ZooKeeper connection *'<MYZK>'*

> **Note**
>
> environment variable name should consist only of uppercase letters, digits and underscores.

CONNECTION_<MYZK>_AUTH_<MYAUTH>_SCHEME

Auth scheme for auth entry *'<MYAUTH>'* for preset ZooKeeper connection *'<MYZK>'*

> **Note**
>
> environment variable name should consist only of uppercase letters, digits and underscores.

CONNECTION_<MYZK>_AUTH_<MYAUTH>_ID

Auth id for auth entry *'<MYAUTH>'* for preset ZooKeeper connection *'<MYZK>'*

> **Note**
>
> environment variable name should consist only of uppercase letters, digits and underscores.

AUTO_CONNECT_CONNECTION_ID

If set, enables *Auto Connect* feature.

Set to `MYZK` to automatically connect to connection defined by `CONNECTION_MYZK_CONN` environment variable.

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. This value is in milliseconds.

> **Note**
>
> on client timeout user does not get logged out unlike in event of session timeout

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.

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.

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.

1.3.2 Preset Connections

Instead of manually entering connection string every time you open ZooNavigator, you can preset your ZooKeeper clusters when starting the Docker image and then select which cluster to connect to from the drop-down in the Connect form.

Adding a Preset Connection

Each connection must have a unique id, which is part of the environment variable name.

To add a preset connection, you must enter a valid connection string as `CONNECTION_<ID>_CONN` environment variable.

Optionally, you can add a name to your preset connection using environment variable `CONNECTION_<ID>_NAME` which will be displayed in the UI instead of the id.

> **Note**
>
> Replace <ID> by any id string you choose. Keep in mind that environment variable name should consist only of uppercase letters, digits and underscores.

For example, if one wants to add a ZooKeeper cluster running at *localhost:2181*, name it "Local ZooKeeper" and give it id *LOCALZK*, then he/she would set the environment variables like so:

- `CONNECTION_LOCALZK_NAME` = **Local ZooKeeper**
- `CONNECTION_LOCALZK_CONN` = **localhost:2181**

Adding auth entry to a Preset Connection

Similarly, one can add auth entry to any Preset Connection.

Each auth entry also has to have a unique id (unique constraint only applies for auth entries for the same connection). To add auth entry you need to set these environment variables:

- `CONNECTION_<ID>_AUTH_<AUTHID>_SCHEME`
- `CONNECTION_<ID>_AUTH_<AUTHID>_ID`

1.3.3 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, first add the ZooKeeper cluster you want to use as a *Preset Connection* to ZooNavigator.

Then simply set the `AUTO_CONNECT_CONNECTION_ID` environment variable to the id of that preset connection.

For example, if you added your ZooKeeper cluster using `CONNECTION_MYZK_CONN` environment variable, then you would set `AUTO_CONNECT_CONNECTION_ID` to `MYZK`.

That's it. When you open up ZooNavigator, it will automatically connect to your ZooKeeper cluster.

1.4 Snap

ZooNavigator is available at Snap store.

The following pages explain how to manage ZooNavigator Snap.

1.4.1 Configuration

ZooNavigator exposes Snap configuration options that could be split into three groups:

- *ZooNavigator* - configures ZooNavigator and the web server
- *ZooKeeper client* - configuration related to ZooKeeper
- *Java* - configures the Java Virtual Machine

ZooNavigator

zoonavigator.server.http.port

default: `9000`

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

zoonavigator.server.https.port

If set, HTTPS server will bind to this port.

zoonavigator.server.https.key-store.path

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

zoonavigator.server.https.key-store.password

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

zoonavigator.server.https.key-store.type

default: `JKS`

The key store type.

zoonavigator.secret-key

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

zoonavigator.base-href

default: */*

Sets base URL where ZooNavigator will be served. If you want ZooNavigator to be available at 'http://www.your-domain.com/zoonavigator' instead of 'http://www.your-domain.com' set this option to */zoonavigator*.

> **Note**
>
> base href must start with '/'

zoonavigator.request.timeout

default: `10000`

Sets timeout for ZooNavigator requests. This value is in milliseconds.

zoonavigator.request.max-size

default: `10000`

Sets maximum request size. Important for large ZNode imports. This value is in kilobytes.

zoonavigator.connection.<MYZK>.name

Optional name for preset ZooKeeper connection *'<MYZK>'*

zoonavigator.connection.<MYZK>.connection-string

Connection string for preset ZooKeeper connection *'<MYZK>'*

zoonavigator.connection.<MYZK>.auth.<MYAUTH>.scheme

Auth scheme for auth entry *'<MYAUTH>'* for preset ZooKeeper connection *'<MYZK>'*

zoonavigator.connection.<MYZK>.auth.<MYAUTH>.id

Auth id for auth entry *'<MYAUTH>'* for preset ZooKeeper connection *'<MYZK>'*

zoonavigator.auto-connect.connection-id

If set, enables *Auto Connect* feature.

Set to `MYZK` to automatically connect to connection defined by `zoonavigator.connection.MYZK.connection-string` configuration option.

ZooKeeper client

zookeeper.client.max-age

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. This value is in milliseconds.

> **Note**
>
> on client timeout user does not get logged out unlike in event of session timeout

zookeeper.client.connect-timeout

default: `5000`

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

zookeeper.sasl.client.enabled

default: `true`

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

zookeeper.sasl.client.config

default: `Client`

Specifies the context key in the JAAS login file.

zookeeper.sasl.client.username

default: `zookeeper`

Specifies the primary part of the server principal. Learn more here.

zookeeper.server.realm

Realm part of the server principal.

By default it is the client principal realm.

zookeeper.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.

zookeeper.ssl.key-store.location

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

zookeeper.ssl.key-store.password

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

zookeeper.ssl.trust-store.location

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

zookeeper.ssl.trust-store.password

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

Java

java.xmx

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

java.xms

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

java.security.auth.login.config

Path to JAAS login configuration file to use.

java.security.krb5.debug

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

java.security.krb5.realm

Sets the default Kerberos realm.

java.security.krb5.kdc

Sets the default Kerberos KDC.

1.4.2 Preset Connections

Instead of manually entering connection string every time you open ZooNavigator, you can preset your ZooKeeper clusters when starting the Docker image and then select which cluster to connect to from the drop-down in the Connect form.

Adding a Preset Connection

Each connection must have a unique id, which is part of the configuration option name.

To add a preset connection, you must enter a valid connection string as `zoonavigator.connection.<id>.connection-string` configuration option.

Optionally, you can add a name to your preset connection using configuration option `zoonavigator.connection.<id>.name` which will be displayed in the UI instead of the id.

> **Note**
>
> Replace <id> by any id string you choose.

For example, if one wants to add a ZooKeeper cluster running at *localhost:2181*, name it "Local ZooKeeper" and give it id *localzk*, then he/she would set the configuration options like so:

```
snap set zoonavigator zoonavigator.connection.localzk.name=Local ZooKeeper
snap set zoonavigator zoonavigator.connection.localzk.connection-string=localhost:2181
```

Adding auth entry to a Preset Connection

Similarly, one can add auth entry to any Preset Connection.

Each auth entry also has to have a unique id (unique constraint only applies for auth entries for the same connection). To add auth entry you need to set these configuration option:

- `zoonavigator.connection.<connectionId>.auth.<authId>.scheme`
- `zoonavigator.connection.<connectionId>.auth.<authId>.id`

1.4.3 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, first add the ZooKeeper cluster you want to use as a *Preset Connection* to ZooNavigator.

Then simply set the `zoonavigator.auto-connect.connection-id` configuration option to the id of that preset connection.

For example, if you added your ZooKeeper cluster using `zoonavigator.connection.myzk.connection-string` configuration option, then you would set `zoonavigator.auto-connect.connection-id` to `myzk` like so:

```
snap set zoonavigator zoonavigator.auto-connect.connection-id=myzk
```

That's it. When you open up ZooNavigator, it will automatically connect to your ZooKeeper cluster.

1.5 Development Guide

1.5.1 Repository Structure

The ZooNavigator codebase is organized as a monorepo:

```
zoonavigator/
├── api/                    # Backend (Scala/Play Framework)
├── web/                    # Frontend (Angular/TypeScript)
├── build/                  # Build configurations
│   ├── docker/             # Docker image build files
│   └── snap/               # Snapcraft package configuration
├── docs/                   # Documentation
└── devenv.nix              # Development environment
```

1.5.2 Local Development

By default, the API development server runs on port `9000` while Angular serves the frontend on port `4200`. The frontend includes a proxy configuration in `web/proxy.conf.js` that routes API requests from port `4200` to the backend API server.

Using devenv

For a streamlined development experience, this repository uses devenv which provides a declarative development environment configuration and sets up all necessary dependencies.

Start all development services:

```
devenv up
```

This command starts all development services including the backend API, frontend and documentation server with auto-reloading. To run in the background, use the `-d` option:

```
devenv up -d
```

Enter the development shell:

```
devenv shell
```

This command enters a shell with all development dependencies available (Scala, Node.js, etc.). You can use this to run individual commands like `sbt`, `npm`, or any other tools needed for development.

> **Note**
>
> You can use direnv to automatically load the development environment when entering the project directory. After installing direnv, run `direnv allow` in the project root to enable automatic environment loading.

Port configuration:

If you need to override default ports, create a `devenv.local.nix` file in the project root:

```
_:
{
   env = {
      API_PORT = 9001;
      WEB_PORT = 4201;
      DOCS_PORT = 8001;
   };
}
```

Manual Setup

If you prefer not to use devenv, you can start the dev servers manually:

Start the API server:

```
cd api
sbt play/run
```

Start the Angular server:

```
cd web
npm install
npm run dev
```

1.6 Frequently Asked Questions

1.6.1 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.

1.6.2 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.

1.6.3 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 using ZooNavigator in Docker: If you're trying to access ZooKeeper running locally on host machine (not in Docker container), follow this advice.

1.7 Change Log

1.7.1 2.0.0

October 14, 2025

Changes:

- Update dependencies
- Drop support for ZooKeeper 3.4.x
- Order connection presets by name or id

1.7.2 1.1.3

July 29, 2024

Changes:

- Updated Play Framework to version 2.8.22
- Updated Docker and Snap builds to use OpenJDK 17 and Ubuntu 22.04

Features:

- Provide Docker image for arm64 architecture

1.7.3 1.1.2

March 31, 2022

Changes:

- CVE-2021-4101 - Remove Log4j 1.2 dependency

1.7.4 1.1.1

October 28, 2021

Changes:

- Use custom Authorization header to allow auth in proxies

Fixes:

- Correctly handle HEAD HTTP requests

1.7.5 1.1.0

November 03, 2020

Features:

- Configurable applications base URL

1.7.6 1.0.1

June 07, 2020

Fixes:

- Custom base URL routing
- ZNode import

1.7.7 1.0.0

May 16, 2020

Adds capability to manage multiple ZooKeeper clusters at once for larger deployments

Changes:

- Add support for preset ZooKeeper connections
- Switch between ZooKeeper clusters in Editor UI
- Change Auto connect Docker configuration
- Remove Session info dialog
- Show build version on Connect page

1.7.8 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

1.7.9 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`

1.7.10 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

1.7.11 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*

1.7.12 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*

1.7.13 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

1.7.14 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

1.7.15 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

1.7.16 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

1.7.17 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