
Adrestia Documentation

Release 0.2.0

Alex Barry

Apr 21, 2019

Contents:

1	Docker	3
2	Building from Source	5
3	Using the Latest Release	7
4	Authenticating with Adrestia	9
5	Adrestia	11
5.1	Overview	11
5.2	Features	11
6	Indices and tables	13

Go Home

Adrestia is primarily a Reverse Proxy server, which means that without it's dependent services (ie. Crazy Ivan and CLyman), it does not do much. For this reason, Docker is the recommended method of experimenting with Adrestia, as it requires minimal effort to get a basic, development cluster standing up.

Regardless of how you start Adrestia and it's dependent services, the easiest way to explore the API currently provided by the cluster is by using [Postman](#). You'll find an environment and collection JSON in the `src/test/json` folder, which can be imported directly into your instance of Postman.

CHAPTER 1

Docker

To stand up all of Adrestia's dependencies, as well as the Adrestia Docker image, run the following from the 'scripts/linux' folder:

```
./start_dev_cluster.sh
```

The following Web UI's will be available in your browser:

Consul - <http://localhost:8500/ui> Neo4j - <http://localhost:7474>

I recommend *MongoDB Compass* <<https://www.mongodb.com/products/compass>>, Community Edition for a Mongo UI, which can connect on localhost:27017

CHAPTER 2

Building from Source

Once you've got the required backend services started, build and execute the tests for the repository. Please note that integration tests will fail unless you have instances of the required backend services running:

```
./gradlew check
```

And, finally, start Adrestia:

```
./gradlew bootRun
```


CHAPTER 3

Using the Latest Release

Adrestia can also be downloaded as a runnable JAR for the latest release from [here](#).

When using a JAR, unzip the downloaded package, move to the main directory from a terminal, and run:

```
java -jar build/libs/adrestia-0.2.0.jar
```

Authenticating with Adrestia

Go Home

Adrestia provides JWT authentication with Spring Security, and can authenticate from a browser with cookies, or from other applications using a Bearer token. To test out authentication, you can visit <http://adrestia-address:8080/login>, making sure to replace ‘adrestia-address’ with the address of your instance.

The Authentication options in `application.properties` expose the ability to turn on/off authentication, as well as provide an initial admin user and password. Currently, only administrator users can create other users, via the `v1/users/sign-up` endpoint.

5.1 Overview

Adrestia is the Gateway Service for the Aesel Distributed Visualization platform.

An Aesel cluster consists of:

- One or more instances of CLyman
- One or more instances of Crazy Ivan
- A Mongo Cluster

Different clusters manage disparate scenes (or groups of renderable objects), and Adrestia abstracts away these clusters into a single transactional plane, which can be accessed by typical transactional services, and end users. It can be scaled horizontally, and runs against a Mongo Cluster itself.

Adrestia also forms a security layer on top of it's dependent services. It provides a number of additional HTTP security features, including a full-scale authentication flow powered by Auth0.

Full Documentation for Adrestia is available [here](#). Repository for Adrestia is available at [here](#).

Full Documentation for the Aesel Distributed Visualization Platform, including the HTTP API for Adrestia, can be found [here](#).

5.2 Features

- Edge Proxy for HTTP communications with end users
- Authentication supported by Auth0 integration
- Cluster-based routing driven by Scenes (data stored in Crazy Ivan)

- Internal Service mesh for transactional services to communicate with Aesel clusters

Adrestia is a part of the AO Aesel Project, along with [Crazy Ivan](#) & [CLyman](#).

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`