

Timeline Analyzer

Social network timeline analysis tool

Installation Guide

Radek Burget



TimelineAnalyzer Installation Guide

TimelineAnalyzer is built around a central RDF repository that stores the event data. For the event data acquisition, several client applications are provided that may run locally (mainly the event data extractors). For browsing and analyzing the gathered events, a client web application is available that is installed on a web server.

RDF Repository

The following RDF repositories are supported by TimelineAnalyzer:

- [Eclipse rdf4j](#) -- the **rdf4j server** must be installed in a web container as described in the [rdf4j documentation](#).
- [Halyard](#) -- a scalable repository that uses rdf4j and Apache HBase for data storage. The Halyard infrastructure must be installed according to the [installation manual](#) including the [web applications](#) that provide the rdf4j server as well.

A typical installation consists of deploying the `rdf4j-server` and `rdf4j-workbench` packages on a Java application server such as Tomcat. This configuration is also useful for deploying the client web application as described below.

In both cases, a separate repository must be created on the running server using the console tool or a web-based *RDF4J Workbench* that is contained in both software packages. The repository is identified by its ID assigned during its creation (e.g. `testRepo`). The rdf4j SPARQL endpoint must be accessible via the HTTP URL (e.g. `http://localhost:8080/rdf4j-server/repositories/testRepo`).

See the corresponding [rdf4j console documentation](#) or [Halyard console documentation](#) for more details.

Local Client Applications

The client applications are written in Java and are controlled via a command-line interface (see the [User guide](#) for invocation details). The executable JAR archives are provided as a part of the TimelineAnalyzer release (see [Releases](#)) or they may be built from source using Maven.

The client applications require the Java 8 Runtime Environment or higher for running. The Java Development Kit and Maven are required for building from source. Building from sources

is tested in Linux environment.

Installing the Pre-Built Applications

Download the `ClientApps.zip` archive from [Releases](#) and unzip it. It contains the runnable JAR archives that may be used as described in the [User guide](#).

Building the Client Applications

The client applications may be build using maven:

- Clone the complete [GitHub repository](#).
- In the root of the repository, install local libraries by running `cd misc; ./install.sh`
- Execute `mvn clean package` in the root of the repository.

After the build completion, the resulting executable JAR files will be available in `timeline-analyzer-fb/target` , `timeline-analyzer-twitter/target` and `timeline-analyzer-local/target` .

Web Client Application

The web client is implemented in TypeScript and Angular. It directly connects the RDF4J repository, retrieves the event data using the SPARQL queries and visualizes the results in an interactive timeline. There are pre-built configured application packages available in the release or the application may be build from sources.

Using a Pre-built Application Package

The easiest way is to deploy the pre-build WAR package on the same application server that is used for running the RDF4J server:

1. Download the `ta.war` package from [releases](#)
2. Deploy the package on your application server (the same way as the `rdf4j-server` package was deployed).

The application should be available at `<server-url>/ta` (e.g. `http://localhost:8080/ta`).

The pre-built application is configured to connect the `rdf4j` endpoint at `<server-url>/rdf4j-server` where `<server-url>` is the URL of the server where the application is deployed. The default repository name is `ta` .

Building the Web Application from Sources

For building the application, [Angular CLI](#) tool must be properly set up in your system. Then, the application may be built in two ways:

- *Using maven* switch to the `timeline-analyzer` root directory and run `mvn package -P webapp`. The created WAR archive will be stored in `timeline-analyzer/timeline-client-angular/target/ta.war`. The package may be directly deployed on the server.
- *Using angular tools* switch to `timeline-analyzer/misc` and run `./build_client.sh`. This does not create the WAR archive.

In both cases the build static application is available in `timeline-analyzer/timeline-client-angular/dist` and it may be deployed to any web server to the `/ta` folder.

Endpoint Configuration

The endpoint configuration can be found in `timeline-client-angular/src/app/config/config.ts`. It defines the endpoint URL and the repository name for obtaining the event data.

Acknowledgements

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.