GraphDocExplore: Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

A Framework for Graph-based Document Exploration

Project structure

Java packages

Application

- Configs: this package contains the class responsible for parsing the XML configuration file and is a singleton
- Domains: this package contains all domain objects
- Graph: this package contains the interface of the graph generator and one implementation of it
- Logging: this package contains the interface of the logger used in the application and examples of implementation
- Search: this package contains the interface of the searcher used in the application and examples of its implementation

Rest Controllers: contains the controller responsible for dispatching the rest requests

Web Controllers: contains the controller responsible for serving the client application

Resources

Configs: contains the XML configuration files of the application and of the collections list

Data: contains the data that is needed by the server application at runtime

Webapp

The client application, developed mostly with AngularJS

Build process

- The build process is run using the Maven Build Automation Tool. Maven will run the tests, compile the source files, create the war and generate Findbugs and JDepend reports as well as the JavaDoc documentation.
- During the generate-resources phase Maven installs and runs the frontend-maven-plugin, which is responsible for downloading NPM and Bower to take care of the dependency management and the unit-tests of the frontend application (client application)
- To build the project and create a WAR File in the target directory run: `$ mvn install`
- To generate the reports into the target/site directory run: `$ mvn site`

Document indexer

The document indexer is a small CLI java application created by the team to help indexing the documents in Solr server.
This application can be configured using an XML config file to specify the solr server, the indexer and the arguments handler of the CLI application.
The design of the application allows it to be extended and to support more server types (apart from solr) by implementing the already offered interfaces.
Only Solr implementations are currently in use.
Usage:
document−indexer </path/to/config.xml> </path/to/documents> <name (any string)>

Configuration File

The XML config file allows you to customize certain components used by the application. Those components are:

Graph

- You can create your own personal graph generator, add it to the classpath and then enter the fully-qualified class name: <generator>
- You can also specify graph layouts that can be used by the client web application to display the graph: <layout>

Searcher

- Specify the server-url of the search engine
- Specify the fully-qualified name of the class responsible of performing the search operations

Logger

- The fully-qualified name of the class responsible of performing logging operations

Installation process

a) Install Tomcat (or equivalent, only tomcat will be explained here)

b) Configure Tomcat

1. Create a tomcat user with the roles : tomcat, manager, manager-gui, manager-script in `{$TOMCAT_HOME}/conf/tomcat-users.xml`
2. Add or change the Keep-Alive parameter in `{$TOMCAT_HOME}/conf/server.xml` to maxKeepAliveRequests='-1'

c) Install the search engine (this example covers solr as the search engine)

d) Start solr with `bin/solr` start command

e) Create a new solr core (the name used in this tutorial will be documents) with bin/solr create -c documents

f) Use the web admin panel to create the fields used for indexing the documents: Choose documents as the core, then click on 'Schema' -> 'Add field', then add the fields as follows:

```
<field name="dc_text" type="text_general" docValues="false"
        multiValued="false" indexed="true" required="true" stored="true"/>
<field name="groupID" type="string" indexed="true" required="true"
        stored="true"/>
<field name="path" type="string" indexed="false" required="true"
        stored="true"/>
```

g) Use the document indexer to index the documents as described in the document indexer section above

h) Upload the WAR file in tomcat server (use the user account created in step 2)

i) Open your browser, browse to the URL and enjoy !