CommunityMashup2 - Development

Sourcecode can be found at:

https://athene2.informatik.unibw-muenchen.de/MASH/communitymashup2

Development environment - Eclipse

Main class: org.sociotech.cm2.server.CommunityMashupServer

Program arguments:

- -c filename # load config file from resources directory or from other location if full file name is specified
- -i # initialize database

VM arguments

- -Dorg.apache.commons.logging.Log=org.apache.commons.logging.impl.NoOpLog -Dlogback.console.threshold=DEBUG

Basic Sever implementation

DataSetImpl

Stores Item objects in "protected Map<Long,Item> items" - i.e. indexed by the id of the item - ids are Long values!

- getItems()
- getItemsByType(String type)
- getItemById(Long id)
- add(Item item) - add the item to the DataSet - after checking for duplicates (equal) items - return new item or merged item
- addForce(Item item) - add the item to the DataSet without looking for duplicates or merging

For creating new Item objects, the class DataFactory should be used:

- createItem(String type)
- createItem(String type, DataSet dataSet) - just set DataSet in Item - do NOT add to DataSet yet

DataSetImplDB (extends DataSetImpl)

Stores Item objects additionally in relational database - TBD tables

Item

Relevant functions for merging in DataSet

- getTypeString()
- getAttributeTNames()
- getRelationshipTypes()
- isEqualItem(Item item) - check if items are equal (should be merged)
- update(Item item) - merge item to the current one

Logging

The Community Mashup server code is using the Simple Logging Facade for Java (SLF4J) for passing log messages. SLF4J currently is passing the log messages via logback - https://logback.qos.ch/manual/index.html

Configuration file logback.xml in ressources:

```
 <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
   <encoder>
     <pattern>%d{HH:mm:ss} %-5level %logger{36} - %msg%n</pattern>
   </encoder>
   <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
     <level>${logback.console.threshold:-DEBUG}</level>
   </filter>
 </appender>
 <root level="DEBUG">
   <appender-ref ref="STDOUT" />
 </root>
```

You may dynamically configure this file by setting the value of logback.console.treshold to other values than DEBUG.

To change to another configuration you may specify the command line parameter: -Dlogback.configurationFile=/xxxx/logback.xml

(Mashup Server) Configuration

For configuration we are using the Apache Commons Configuration Framework.

https://commons.apache.org/proper/commons-configuration/userguide/user_guide.html

Using the Configuration Framework key-value-pairs are loaded from different sources and can be accessed in the code from a Configuration object.

The following sources are combined:

- resources/META-INF/configuration/*
- resources/mashup.properties (or the file that is specified upon startup via the -c parameter - when not a full path, the file is looked for in the resources folder)
- environment variables (just specify something like "xxx = yyy" in the tab "Environment" in the Run Configuration in Eclipse ...)

Deployment

- Generate a runnable JAR file - e.g. CommunityMashup2.jar - in Eclipse: right-click on project, select Export, select Java/Runnable JAR, select run configuration and select extract required libraries in jar
    - in run configuration: do not select a config file with "-c" - this should be done when executing the JAR file
- copy JAR file to execution environment, e.g. /opt/mashup2 on engelbart.cscwlab.de
- modify configuration file in execution environment if needed
- restart server - e.g. via /etc/init.d/mashup2 restart