Model Web Service Client Sample

This is a short description how to use `model-client-sample` which demonstrates the use of Java client based on JAX-WS. This approach is beneficial because it provides good maintainability of the code on the compiler level. I.e. if the midPoint API changes in an incompatible manner (in future midPoint releases) than the client code will break and indicate the exact place that needs to be fixed. This is saving a lot of testing effort. If the midPoint API changes in a compatible manner then the client code should work without any major change.

MidPoint provides a stand-alone project that is using `model-client` library. The `model-client` library is part of midPoint especially designed to support such clients. It contains a JAX-WS and JAX-B code generated from midPoint schemas. Therefore projects that use this library do not need to generate the code and they can simply use this library. There is no need to build `model-client` library, it is available as a binary JAR in Evolveum Maven Repository. Maven can take advantage of that and he only part that needs to be built is the client code itself. The sample project is configured to do just that. Please inspect the `pom.xml` file in the project for more details.

Using the Sample

Check-out the Source Code

Use one of the following URLs based on your midPoint version:

Version	URL
3.9	https://github.com/Evolveum/midpoint/tree/v3.9/samples/model-client-sample
3.8	https://github.com/Evolveum/midpoint/tree/v3.8/samples/model-client-sample
3.7.2	https://github.com/Evolveum/midpoint/tree/v3.7.2/samples/model-client-sample
3.7.1	https://github.com/Evolveum/midpoint/tree/v3.7.1/samples/model-client-sample
3.7	https://github.com/Evolveum/midpoint/tree/v3.7/samples/model-client-sample
3.6	https://github.com/Evolveum/midpoint/tree/v3.6/samples/model-client-sample
3.5	https://github.com/Evolveum/midpoint/tree/v3.5/samples/model-client-sample
3.4.1	https://github.com/Evolveum/midpoint/tree/v3.4.1/samples/model-client-sample
3.4	https://github.com/Evolveum/midpoint/tree/v3.4/samples/model-client-sample
3.3.1	https://github.com/Evolveum/midpoint/tree/v3.3.1/samples/model-client-sample
3.3	https://github.com/Evolveum/midpoint/tree/v3.3/samples/model-client-sample
3.2	https://github.com/Evolveum/midpoint/tree/v3.2/samples/model-client-sample
3.1.1	https://github.com/Evolveum/midpoint/tree/v3.1.1/samples/model-client-sample
3.1	https://github.com/Evolveum/midpoint/tree/v3.1/samples/model-client-sample
3.0.1	https://github.com/Evolveum/midpoint/tree/v3.0.1/samples/model-client-sample
3.0	https://github.com/Evolveum/midpoint/tree/v3.0/samples/model-client-sample
2.1.1	https://github.com/Evolveum/midpoint/tree/midpoint-2.1.1/samples/model-client-sample
development snapshot	https://github.com/Evolveum/midpoint/tree/master/samples/model-client-sample

Use git to check out the source code:

```
git clone https://github.com/Evolveum/midpoint.git midpoint
cd midpoint/samples/model-client-sample
```

Adjust the Configuration

The sample is very simple. All the configuration is hardcoded into the source code. But it is quite easy to change. Just edit few source files:

Setting	Default	File	Location
Web service URL	http://localhost:8080/midpoint/ws/model-3	`src/main/java/com/evolveum/midpoint/testing/model/client/sample/Main.java`	`createModelPort` method

midPoint user name	administrator	src/main/java/com/evolveum/midpoint/testing /model/client/sample/Main.java	createModelPort method
midPoint user password	5ecr3t	src/main/java/com/evolveum/midpoint/testing /model/client/sample/ClientPasswordHandler. java	handle method
OIDs of roles used in the sample	12345678-d34d-b33f-f00d-987987987988(Pirate), 12345678-d34d-b33f-f00d-987987cccccc(Captain)	src/main/java/com/evolveum/midpoint/testing /model/client/sample/Main.java	constants ROLE_PIRATE_OID, ROLE_CAPTAIN_OID

The scenario that the client sample does in midPoint is hardcoded in the `main` method of the `Main` class. You can adjust it or simple comment out some parts. It goes roughly like this:

1. Getting system configuration object
2. Search for user with username `administrator`
3. Searching for role with name `Sailor`
4. Creating user `Guybrush Threepwood` with username `guybrush`
5. Changing password of user `guybrush`
6. Assing roles `Pirate` and `Captain` to user `guybrush` (roles are identified by fixed OID
7. Unasssign role `Captain`
8. List all roles that have `requestable` flag set to true
9. Delete user `guybrush` (which also deletes all the accounts that the roles assigned to the user)

Build It

Use Maven `mvn` command to build the sample:

```
mvn clean package
```

The resulting JAR should appear in the `target` subdirectory.

Run It

While it is theoretically possible to use `java -jar` command to run the sample this is a bit difficult in practice. The code will need some libraries on the classpath to run, especially the model-client midPoint library. It is much easier to run the sample using Maven. Maven will make sure that the classpath is populated as needed. Using maven this is not a major problem because this just a sample project. The real midPoint client will be most likely integrated into a larger project and executed a any way that is native to that integrated project.

```
mvn exec:java -Dexec.mainClass="com.evolveum.midpoint.testing.model.client.sample.Main"
```

Do not forget that initialized and configued midPoint instance must be running before successfully executing the client sample.