AE UPM3 Project Creation

Table of Contents

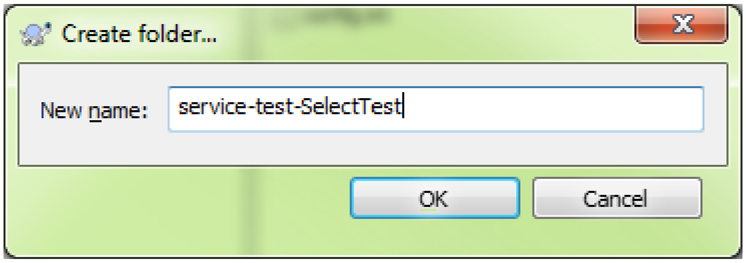
SOAP SERVICE PROJECT CREATION 3
WMB DATA SERVICE PROJECT CREATION 11
WEB DATA SERVICE PROJECT CREATION 19
REST DATA SERVICE PROJECT CREATION 30
INTIAL SETUP.. 40

SOAP Service Project Creation

1. **If this is your first time using maven archetypes and you haven't already set them up for use in Intellij, please follow the instructions for Initial Setup.**

2. Open TortoiseSVN repo browser (right-click anywhere in Windows Explorer and select TortoiseSVN/Repo-browser) to: http://svn.uhc.com/svn/uhgit_upm3

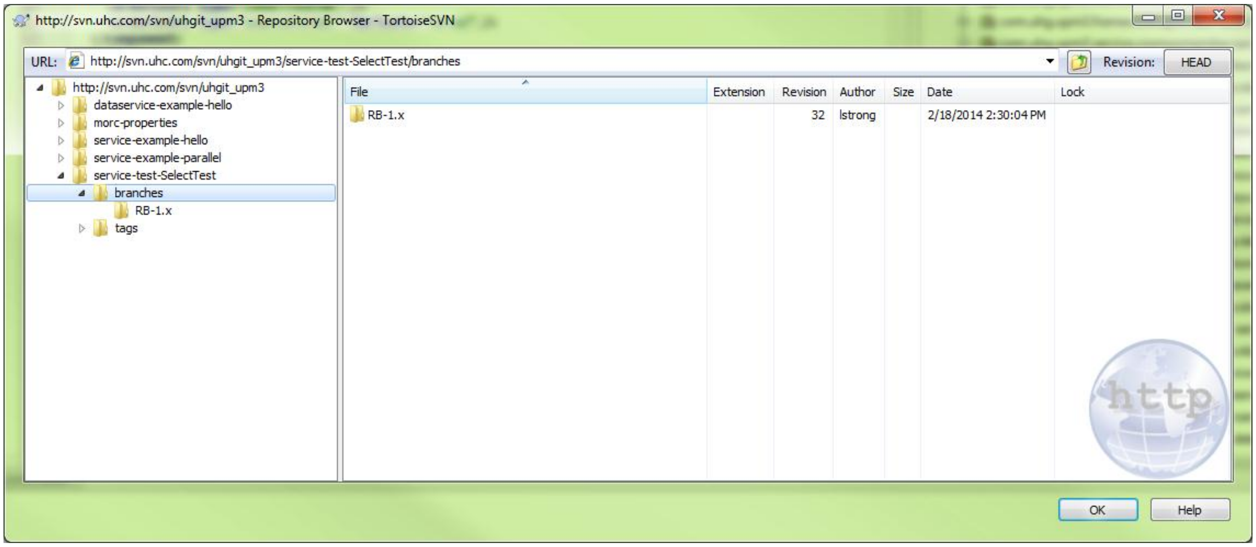
3. Create the new project at the root level of the repository by selecting "Create Folder…" (do not select "Add Folder…"). Name the project without the version (leave off –v1, etc). Naming convention should be service-[category]-[ServiceName]



New name: service-test-SelectTest

4. Under that directory, Create 2 folders: "branches" and "tags".

5. Under "branches", create the release branch for the version(s):
   RB-1.x, RB-2.x, etc. (note the "x" is intentional—don't replace it) Version should correspond to the version of the service.


http://svn.uhc.com/svn/uhgit_upm3 - Repository Browser - TortoiseSVN
http://svn.uhc.com/svn/uhgit_upm3
dataservice-example-hello
morc-properties
service-example-hello
service-example-parallel
service-test-SelectTest
branches
RB-1.x
tags
OK
Help

URL: http://svn.uhc.com/svn/uhgit_upm3/service-test-SelectTest/branches

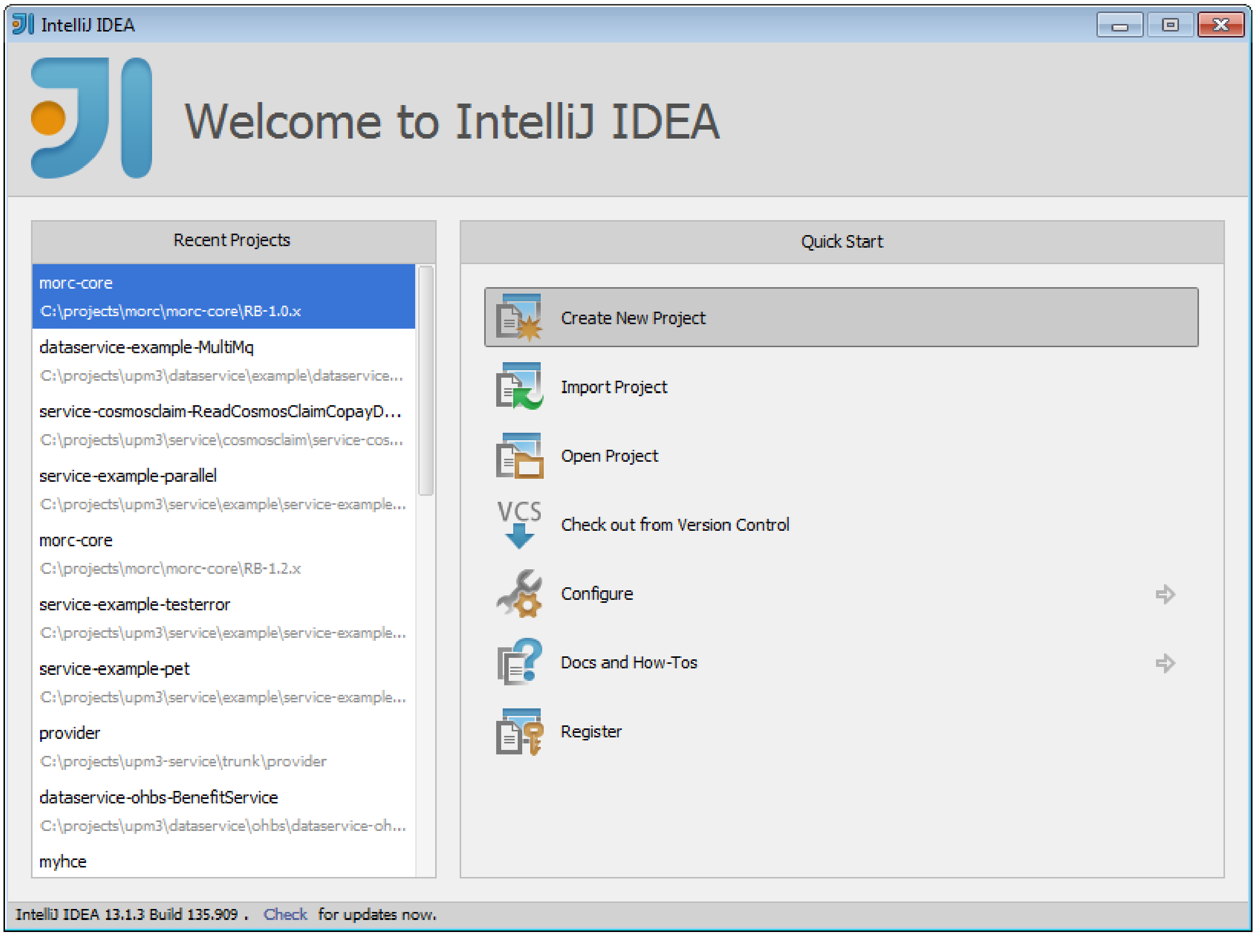
Revision: HEAD

File	Extension	Revision	Author	Size	Date	Lock
RB-1.x		32	lstrong		2/18/2014 2:30:04 PM	

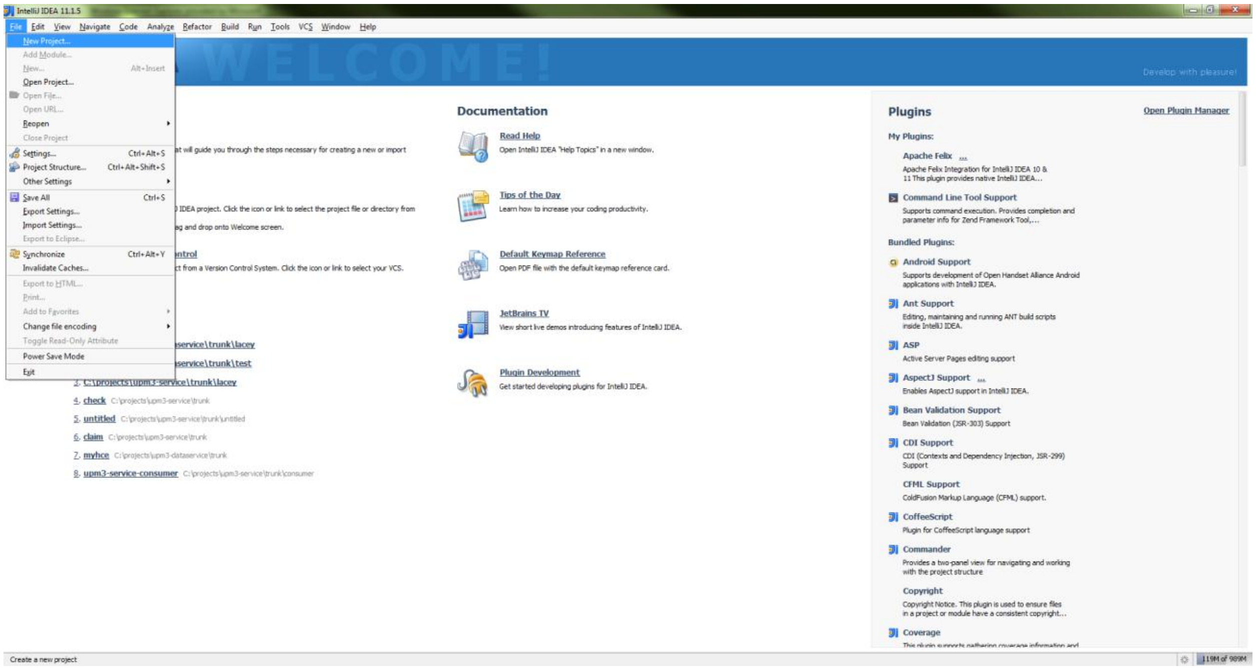
6. On local filesystem, create the project and branch directories:
   C:\projects\upm3\service\test\service-test-SelectTest\RB-1.x

7. Checkout from SVN to that directory
   Repo URL: http://svn.uhc.com/svn/uhgit_upm3/service-test-SelectTest/branches/RB-1.x/
   Local directory: C:\projects\upm3\service\test\service-test-SelectTest\RB-1.x

8. Open Intellij and select Create New Project or File→New Project… if other projects are open


IntelliJ IDEA
Welcome to IntelliJ IDEA
Recent Projects
morc-core
C:\projects\morc\morc-core\RB-1.0.x
dataservice-example-MultiMq
C:\projects\upm3\dataservice\example\dataservice…
service-cosmosclaim-ReadCosmosClaimCopayD…
C:\projects\upm3\service\cosmosclaim\service-cos…
service-example-parallel
C:\projects\upm3\service\example\service-example…
morc-core
C:\projects\morc\morc-core\RB-1.2.x
service-example-testerror
C:\projects\upm3\service\example\service-example…
service-example-pet
C:\projects\upm3\service\example\service-example…
provider
C:\projects\upm3-service\trunk\provider
dataservice-ohbs-BenefitService
C:\projects\upm3\dataservice\ohbs\dataservice-oh…
myhce
Quick Start
Create New Project
Import Project
Open Project
VCS
Check out from Version Control
Configure
Docs and How-Tos
Register

IntelliJ IDEA 13.1.3 Build 135.909 . Check for updates now.



t will guide you through the steps necessary for creating a new or import

IDEA project. Click the icon or link to select the project file or directory from

ag and drop onto Welcome screen.

ntrol

ct from a Version Control System. Click the icon or link to select your VCS.

service\trunk\lacey

service\trunk\test

3. C:\projects\upm3-service\trunk\lacey
4. check C:\projects\upm3-service\trunk
5. untitled C:\projects\upm3-service\trunk\untitled
6. claim C:\projects\upm3-service\trunk
7. myhce C:\projects\upm3-dataservice\trunk
8. upm3-service-consumer C:\projects\upm3-service\trunk\consumer

Open IntelliJ IDEA "Help Topics" in a new window.

Learn how to increase your coding productivity.

Open PDF file with the default keymap reference card.

View short live demos introducing features of IntelliJ IDEA.

Get started developing plugins for IntelliJ IDEA.

Apache Felix Integration for IntelliJ IDEA 10 & 11 This plugin provides native IntelliJ IDEA…

Supports command execution. Provides completion and parameter info for Zend Framework Tool,…

Supports development of Open Handset Alliance Android applications with IntelliJ IDEA.

Editing, maintaining and running ANT build scripts inside IntelliJ IDEA.

Active Server Pages editing support

Enables AspectJ support in IntelliJ IDEA.

Bean Validation (JSR-303) Support

CDI (Contexts and Dependency Injection, JSR-299) Support

ColdFusion Markup Language (CFML) support.

Plugin for CoffeeScript language support

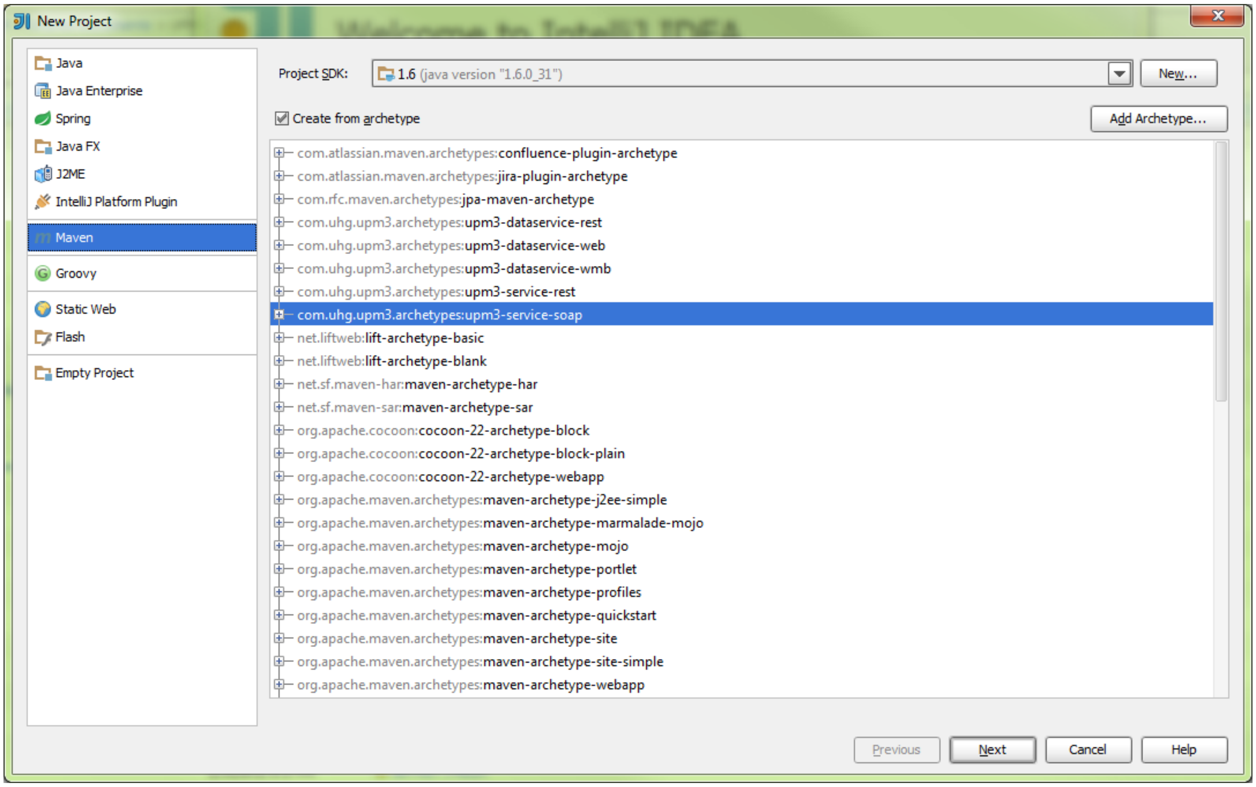
Provides a two-panel view for navigating and working with the project structure

Copyright Notice. This plugin is used to ensure files in a project or module have a consistent copyright…

This plugin supports gathering coverage information and

Create a new project

9. Select "Maven" and check the box for "Create from archetype". Select the com.uhg.upm3.archetypes:upm3-service-soap archetype and click "Next". For the Project SDK: make sure that "1.6" is selected.



Project SDK: 1.6 (java version "1.6.0_31")

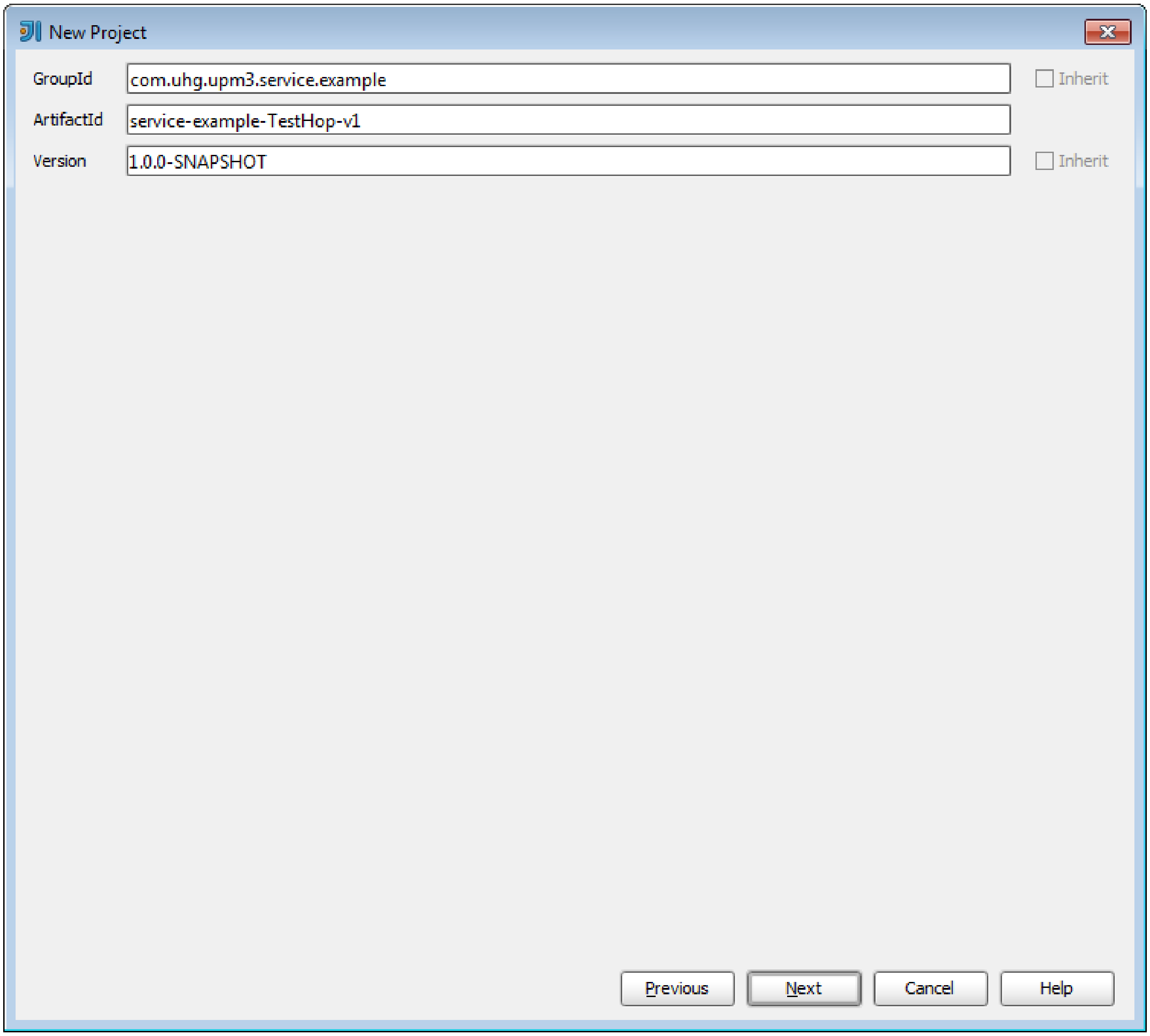
☑ Create from archetype

- com.atlassian.maven.archetypes:confluence-plugin-archetype
- com.atlassian.maven.archetypes:jira-plugin-archetype
- com.rfc.maven.archetypes:jpa-maven-archetype
- com.uhg.upm3.archetypes:upm3-dataservice-rest
- com.uhg.upm3.archetypes:upm3-dataservice-web
- com.uhg.upm3.archetypes:upm3-dataservice-wmb
- com.uhg.upm3.archetypes:upm3-service-rest
- com.uhg.upm3.archetypes:upm3-service-soap
- net.liftweb:lift-archetype-basic
- net.liftweb:lift-archetype-blank
- net.sf.maven-har:maven-archetype-har
- net.sf.maven-sar:maven-archetype-sar
- org.apache.cocoon:cocoon-22-archetype-block
- org.apache.cocoon:cocoon-22-archetype-block-plain
- org.apache.cocoon:cocoon-22-archetype-webapp
- org.apache.maven.archetypes:maven-archetype-j2ee-simple
- org.apache.maven.archetypes:maven-archetype-marmalade-mojo
- org.apache.maven.archetypes:maven-archetype-mojo
- org.apache.maven.archetypes:maven-archetype-portlet
- org.apache.maven.archetypes:maven-archetype-profiles
- org.apache.maven.archetypes:maven-archetype-quickstart
- org.apache.maven.archetypes:maven-archetype-site
- org.apache.maven.archetypes:maven-archetype-site-simple
- org.apache.maven.archetypes:maven-archetype-webapp

10. Set the maven artifact values:
    a. GroupId should be com.uhg.upm3.service.[category] (eg. com.uhg.upm3.service.test)

    b. ArtifactId should be service-[category]-[ServiceName]-v[version] (eg. service-test-SelectTest-v1)

    c. Version should correspond to the version number of the service with SNAPSHOT (eg. 1.0.0-SNAPSHOT)

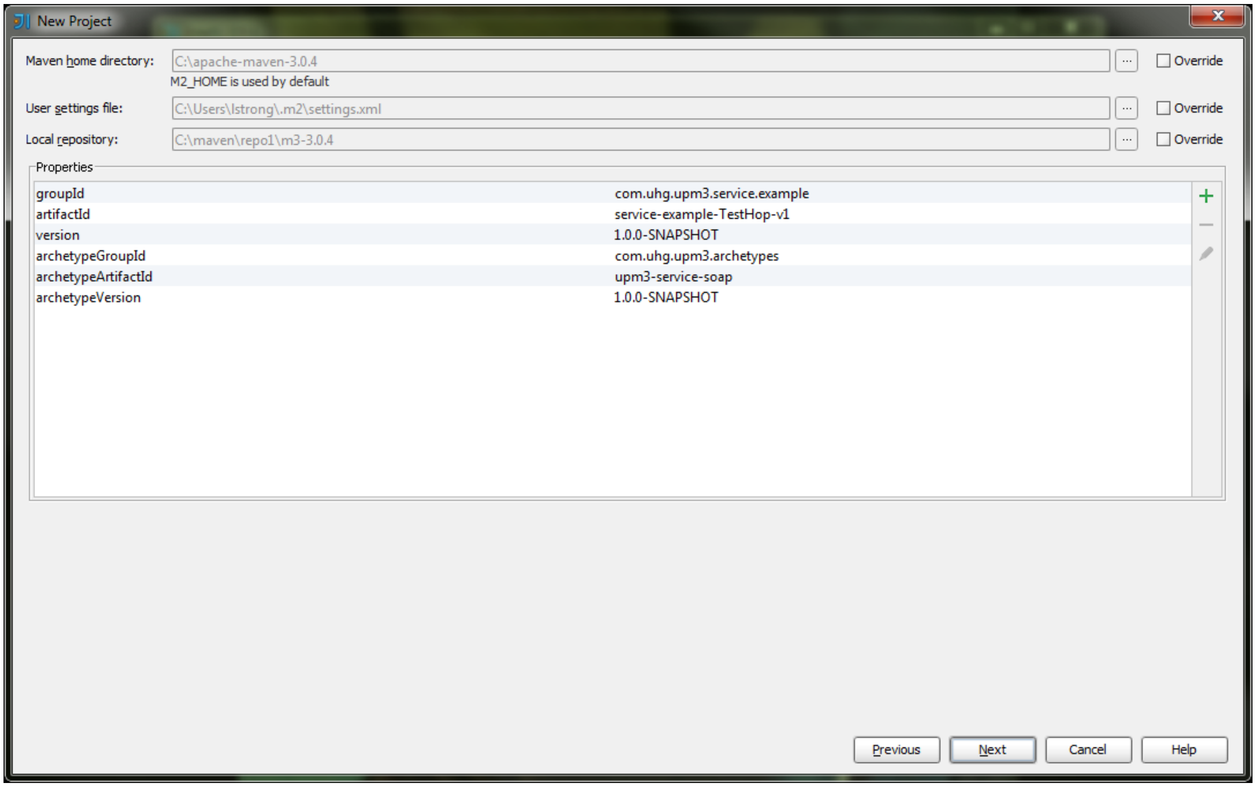


GroupId com.uhg.upm3.service.example

ArtifactId service-example-TestHop-v1

Version 1.0.0-SNAPSHOT

11. Click "Next" and leave the default maven settings on the next screen.



Maven home directory: C:\apache-maven-3.0.4 □ Override
M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml □ Override

Local repository: C:\maven\repo1\m3-3.0.4 □ Override

Properties

groupId	com.uhg.upm3.service.example
artifactId	service-example-TestHop-v1
version	1.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-service-soap
archetypeVersion	1.0.0-SNAPSHOT

12. Click "Next" again and enter the following values:

    a. For Project name: Enter the name of the project that you just created on your local filesystem:
    service-test-SelectTest

    b. For Project location: Click on the … and navigate to the folder that you just created.

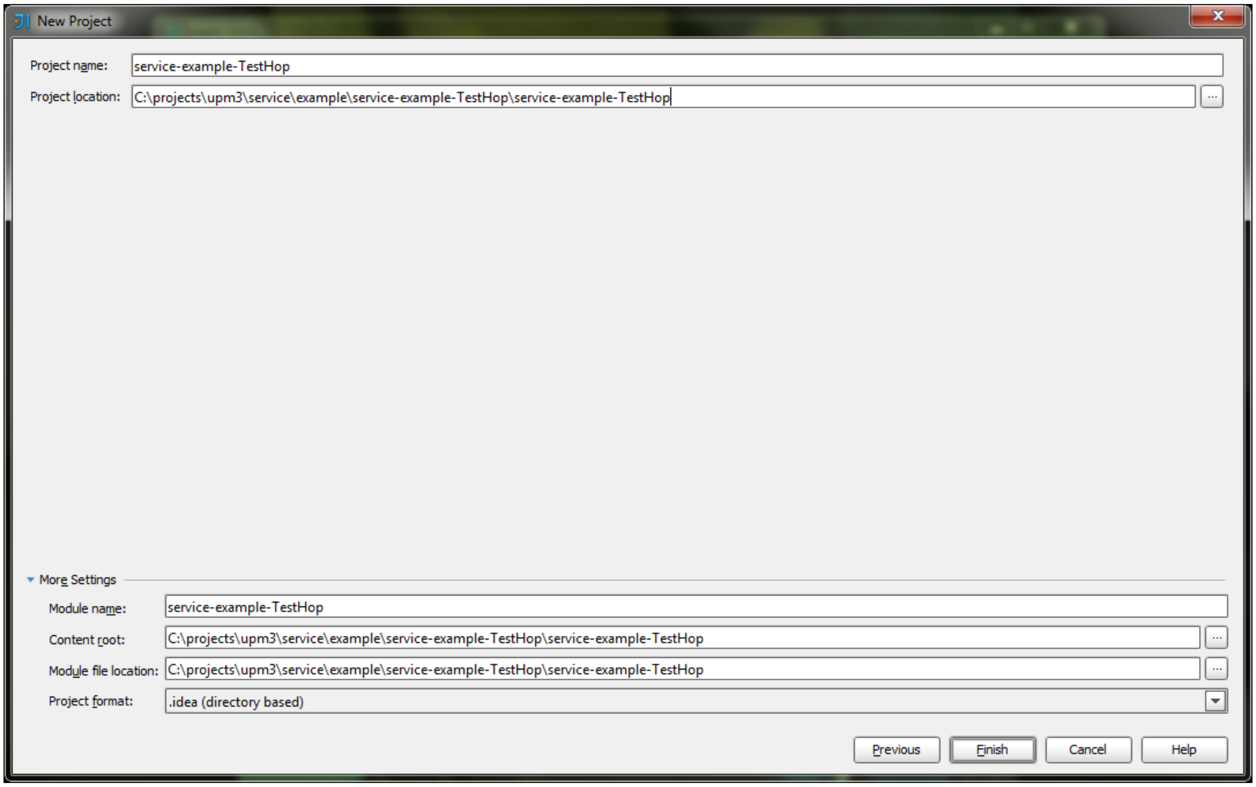
13. Expand the "More Settings" section at the bottom of the window and fill in the following values:

    a. Module name: Copy/Paste the Project name: service-test-SelectTest

    b. Content Root: Should match what is in Project files location. If not, then Copy/Paste it here:
    C:\projects\upm3\service\test\service-test-SelectTest\RB-1.x

    c. Module file location: Should match what is in Project files location. If not, then Copy/Paste it here:
    C:\projects\upm3\service\test\service-test-SelectTest\RB-1.x

    d. Project file format: .idea (directory based) should be selected.



New Project

Project name: service-example-TestHop

Project location: C:\projects\upm3\service\example\service-example-TestHop\service-example-TestHop

More Settings

Module name: service-example-TestHop

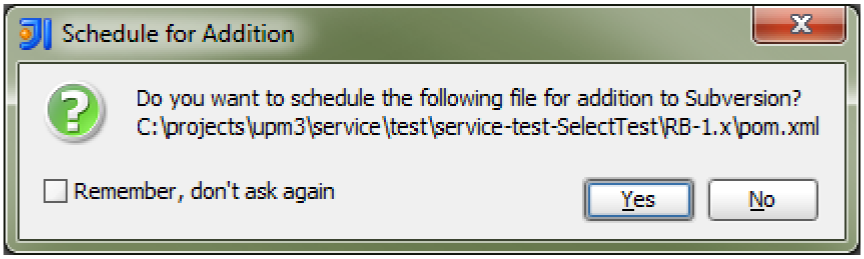
Content root: C:\projects\upm3\service\example\service-example-TestHop\service-example-TestHop

Module file location: C:\projects\upm3\service\example\service-example-TestHop\service-example-TestHop

Project format: .idea (directory based)

14. Click Finish.

15. Intellij will ask if you would like to add the pom.xml to Subversion. Click Yes.



Schedule for Addition

Do you want to schedule the following file for addition to Subversion?
C:\projects\upm3\service\test\service-test-SelectTest\RB-1.x\pom.xml

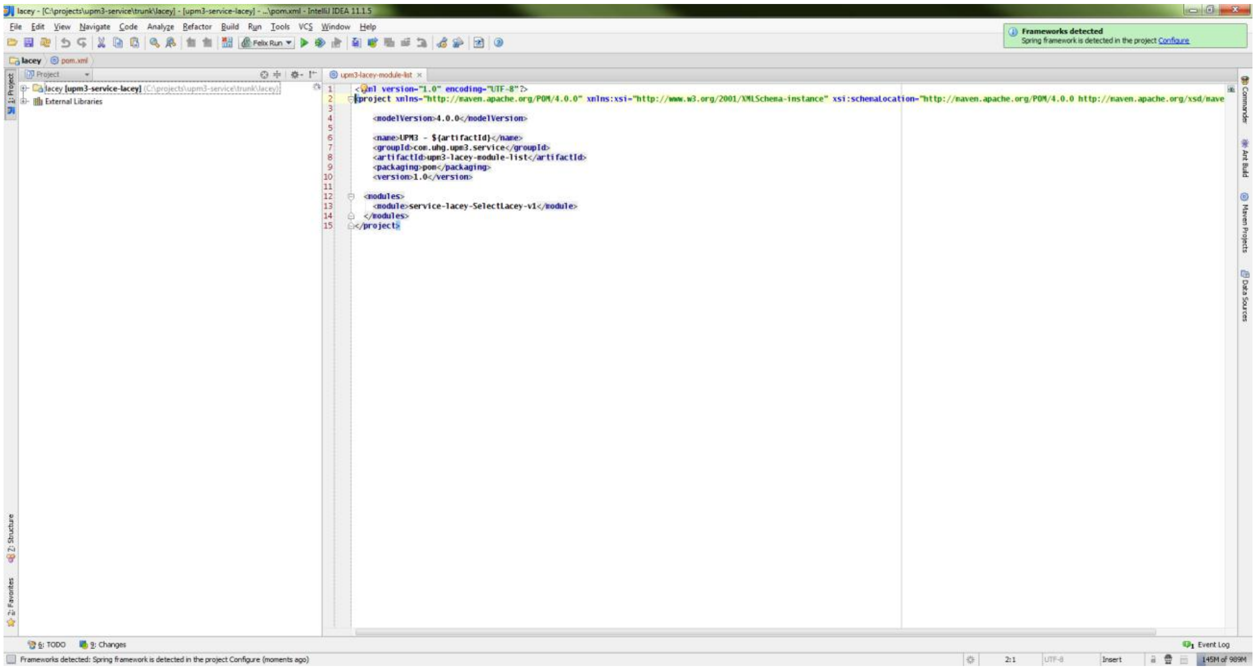
□ Remember, don't ask again

16. Intellij will then create the project from the archetype. If everything was successful you will see a message in the Messages window in Intellij.

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 12.138s
[INFO] Finished at: Tue Feb 18 14:50:33 EST 2014
[INFO] Final Memory: 6M/15M
[INFO] ------------------------------------------------------------------------
[INFO] Maven execution finished
```

17. Close and reopen the project – You won’t be able see/use the Felix Run until you do this.

18. Once you have reopened the project, you will briefly get a popup in the top right corner of your Intellij window that shows a Spring Framework has been detected. Click configure and accept the defaults.



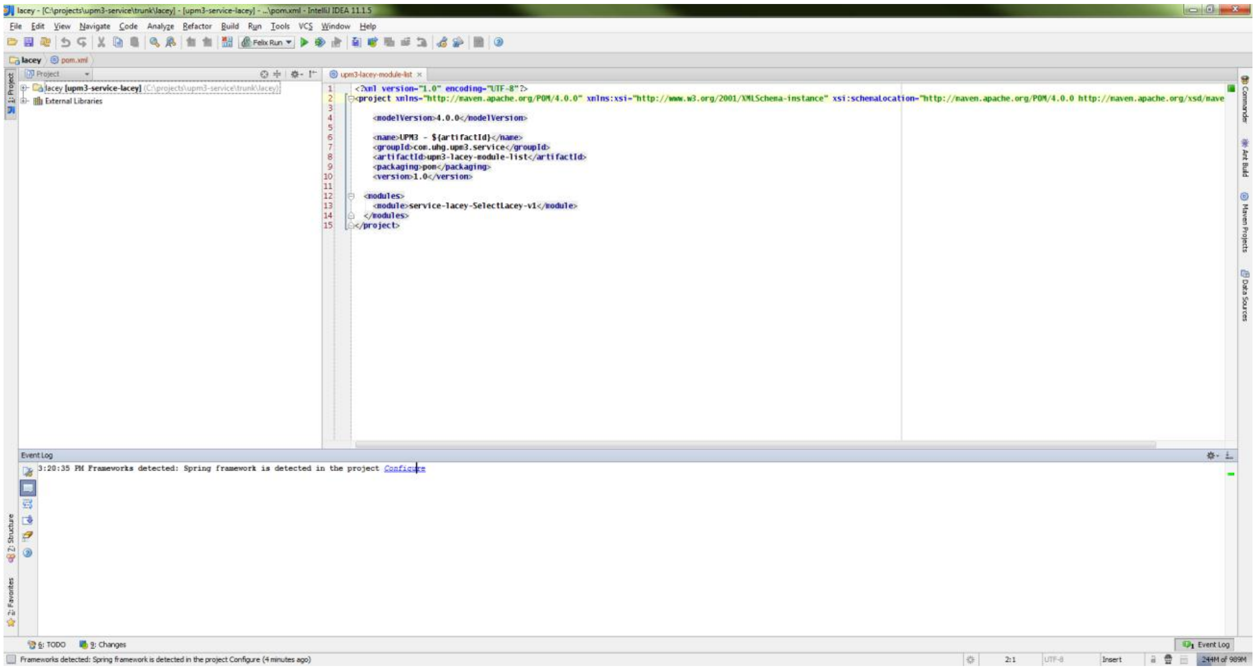
lacey - [C:\projects\upm3-service\trunk\lacey] - [upm3-service-lacey] - ...\pom.xml - IntelliJ IDEA 11.1.5

Frameworks detected
Spring framework is detected in the project Configure

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.service</groupId>
    <artifactId>upm3-lacey-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>service-lacey-SelectLacey-v1</module>
  </modules>
</project>
```

Frameworks detected: Spring framework is detected in the project Configure (moments ago)

If the popup disappears before you are able to click it then open the Event Log in the bottom right corner of Intellij and click on Configure. It will bring up a Setup Frameworks dialog box. Click OK.



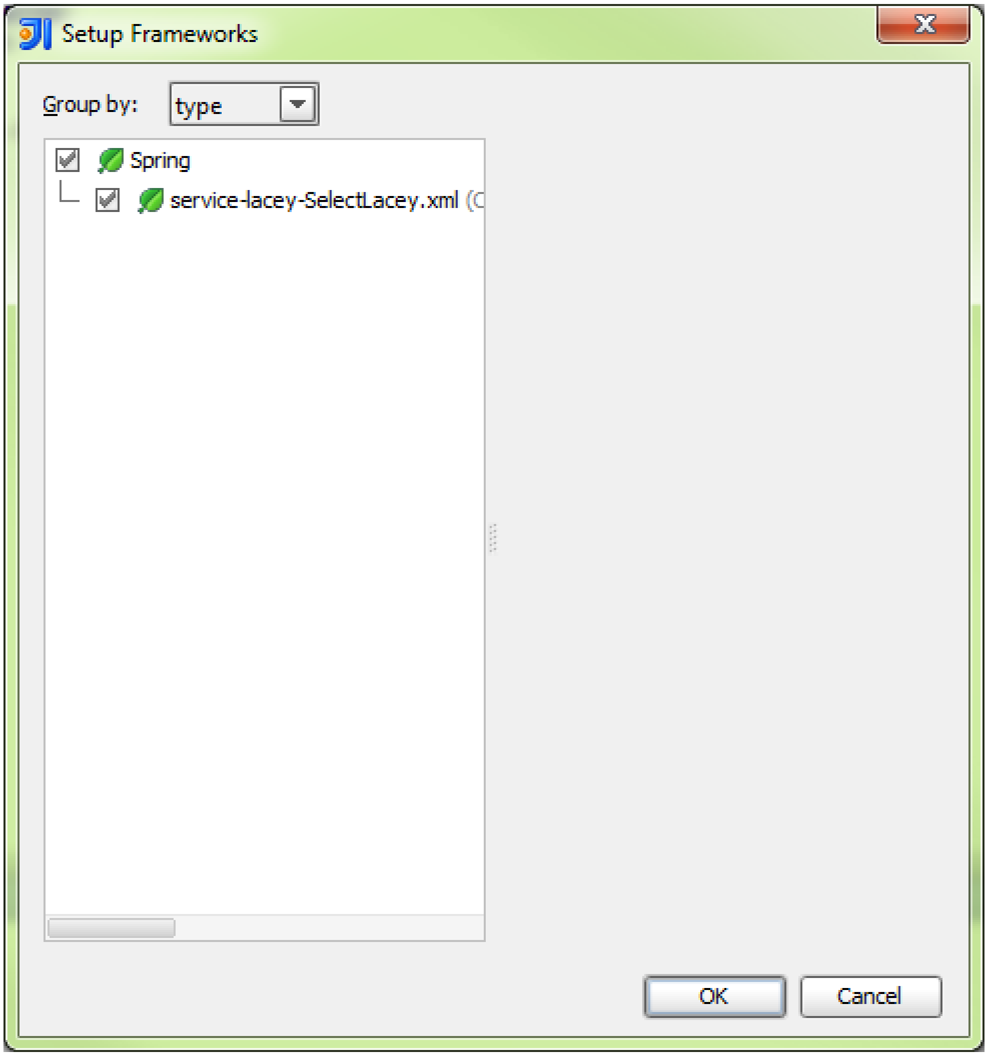
lacey - [C:\projects\upm3-service\trunk\lacey] - [upm3-service-lacey] - ...\pom.xml - IntelliJ IDEA 11.1.5

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.service</groupId>
    <artifactId>upm3-lacey-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>service-lacey-SelectLacey-v1</module>
  </modules>
</project>
```

Event Log

```
3:20:35 PM Frameworks detected: Spring framework is detected in the project Configure
```

Frameworks detected: Spring framework is detected in the project Configure (4 minutes ago)



Group by: type

☑ Spring

☑ service-lacey-SelectLacey.xml (C

****A note about Tricare services or services to be deployed to the Enclave****

- Modify the spring config file to change the interceptor from SamlInboundInterceptor to SignatureInboundInterceptor:

  Change the bolded line from this:

  ```
            <entry key="org.apache.cxf.ws.in.interceptors">
               <array>
                  <osgi:reference
  interface="com.uhg.upm.core.interceptor.ServerInboundCxfInterceptor"/>
                 <osgi:reference
  interface="com.uhg.upm.core.interceptor.SamlInboundInterceptor"/>
               </array>
            </entry>
  ```

  To this:

```
            <entry key="org.apache.cxf.ws.in.interceptors">
                <array>
                    <osgi:reference
    interface="com.uhg.upm.core.interceptor.ServerInboundCxfInterceptor"/>
                    <osgi:reference
    interface="com.uhg.upm.core.interceptor.SignatureInboundInterceptor"/>
                </array>
            </entry>
```

19. Add the Unversioned files to Subversion.
    - If svn brings up any ./out* or ./target* directories DO NOT commit them. Please add them to .svnignore (TortoiseSVN->"Unversion and add to ignore list").
      These directories are used by Maven and Felix as working directories and have the compiled classes and bundle jar files. These directories are not needed and will just add a lot of overhead to commits and updates as they are constantly changing.
    - In the directory ./.idea/ add the file "workspace.xml to .svnignore.
    - In the directory ./.idea/ add the "./libraries" directory to .svnignore.

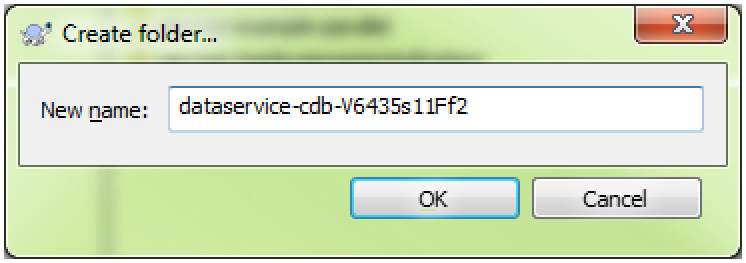
20. Once you have committed your new project it will need to be added to the build farm setup. If you have access, add your project, otherwise send an email to your dev lead or dev manager with the details for your new module, make sure to make note of any dependencies your module has so the build farm can be configured correctly.

WMB Data Service Project Creation

1. **If this is your first time using maven archetypes and you haven't already set them up for use in Intellij, please follow the instructions for Initial Setup.**

2. Open TortoiseSVN repo browser (right-click anywhere in Windows Explorer and select TortoiseSVN/Repo-browser) to: http://svn.uhc.com/svn/uhgit_upm3

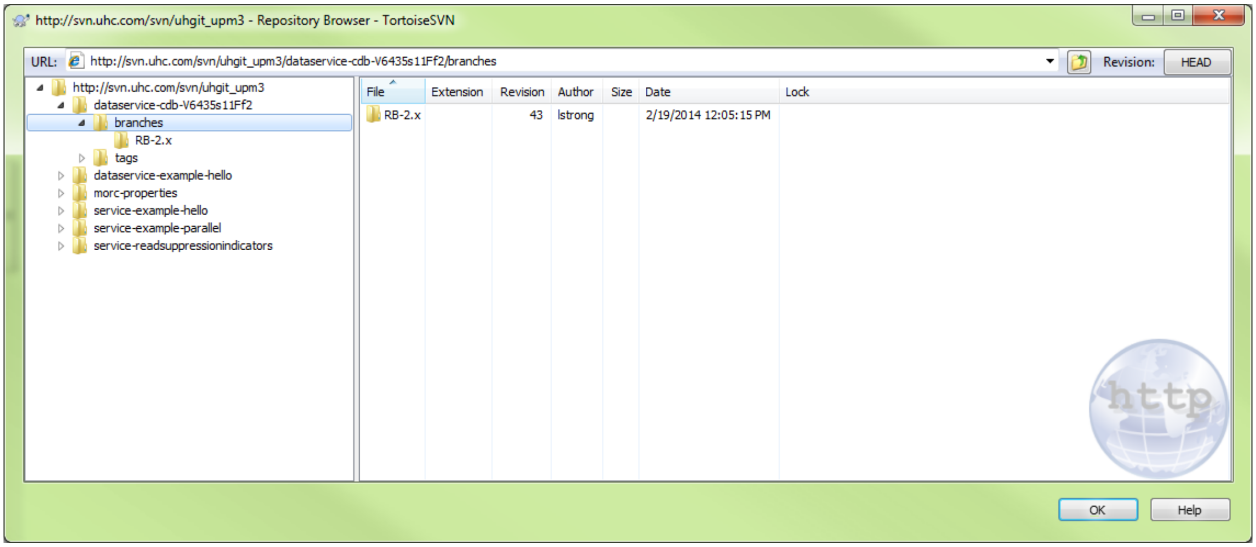
3. Create the new project at the root level of the repository by selecting "Create Folder…" (do not select "Add Folder…"). Name the project without the version (leave off –v1, etc). Naming convention should be dataservice-[backend]-[MessageBrokerServiceName]



New name: dataservice-cdb-V6435s11Ff2

4. Under that directory, Create 2 folders: "branches" and "tags".

5. Under "branches", create the release branch for the version(s):
   RB-1.x, RB-2.x, etc. (note the "x" is intentional—don't replace it) Version should correspond to the MessageBroker version number.


http://svn.uhc.com/svn/uhgit_upm3 - Repository Browser - TortoiseSVN
URL:
Revision:
HEAD
OK
Help
http

URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-cdb-V6435s11Ff2/branches

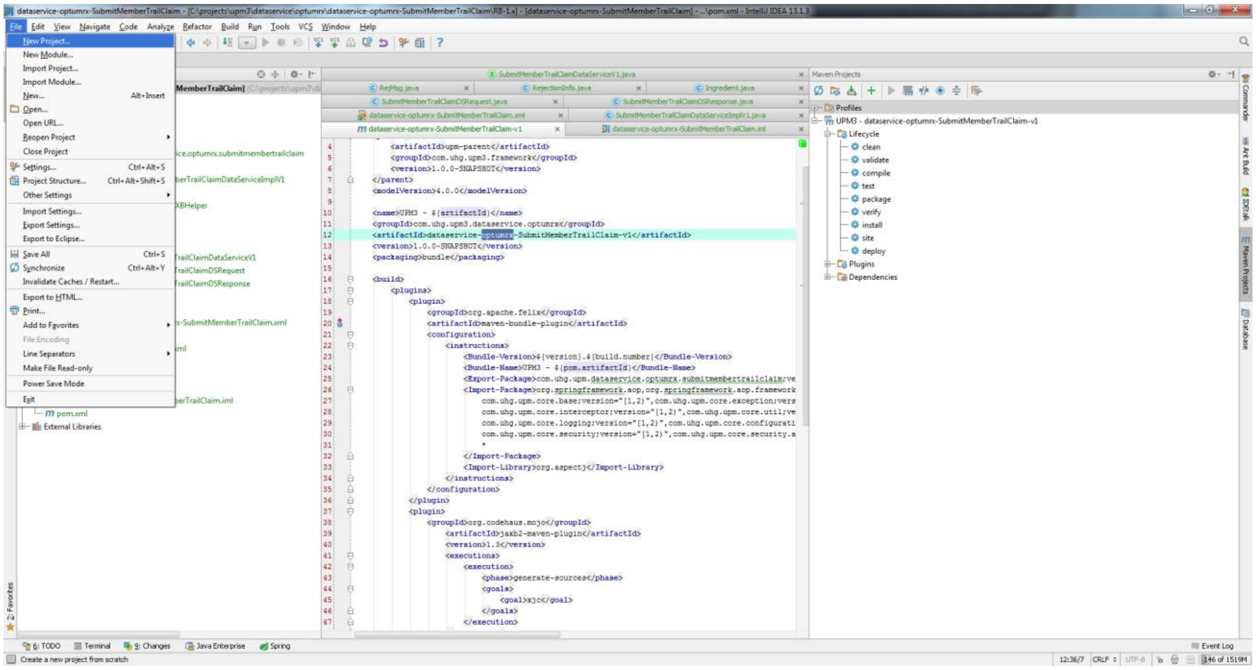
```
http://svn.uhc.com/svn/uhgit_upm3
  dataservice-cdb-V6435s11Ff2
    branches
      RB-2.x
    tags
  dataservice-example-hello
  morc-properties
  service-example-hello
  service-example-parallel
  service-readsuppressionindicators
```

File	Extension	Revision	Author	Size	Date	Lock
RB-2.x		43	lstrong		2/19/2014 12:05:15 PM	

6. On local filesystem, create the project and branch directories:
   C:\projects\upm3\dataservice\cdb\dataservice-cdb-V6435s11Ff2\RB-2.x

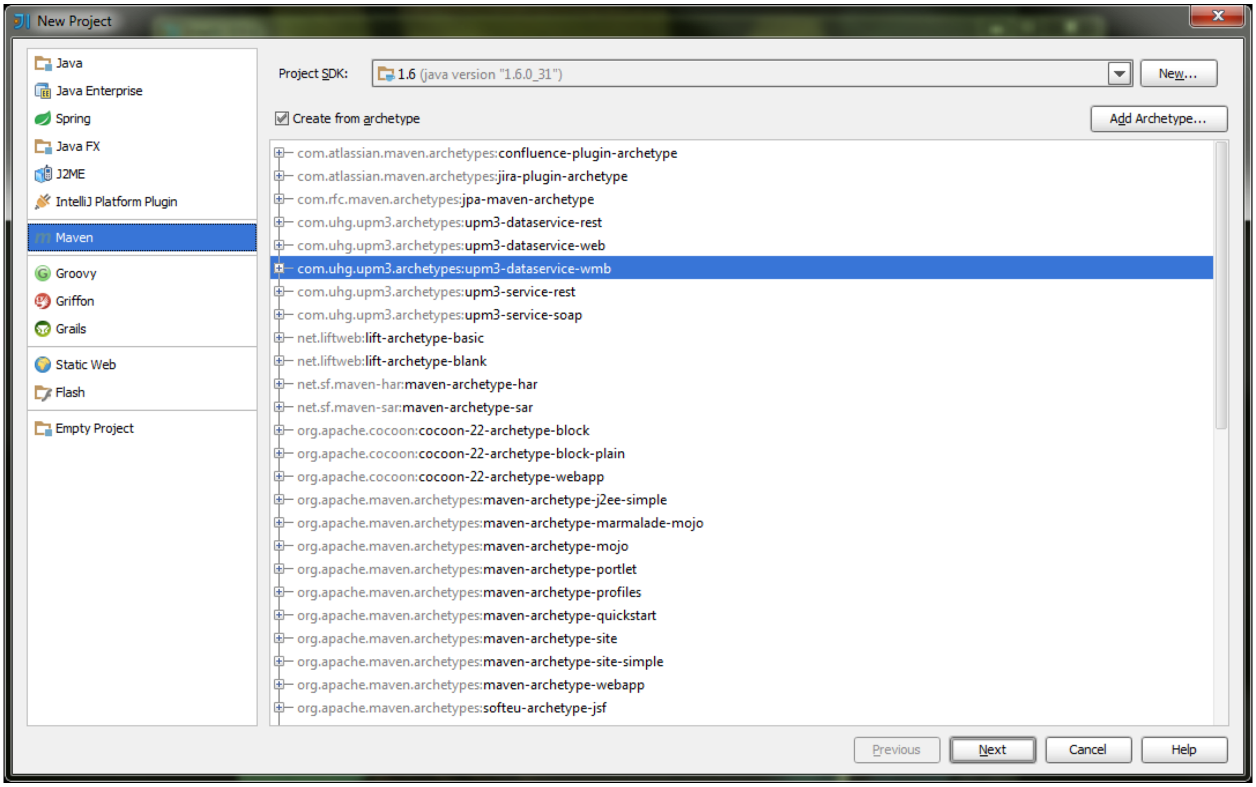
7. Checkout from SVN to that directory
   Repo URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-cdb-V6435s11Ff2/branches/RB-2.x/
   Local directory: C:\projects\upm3\dataservice\cdb\dataservice-cdb-V6435s11Ff2\RB-2.x

8. Open Intellij and select File→New Project…



```
        <artifactId>upm-parent</artifactId>
        <groupId>com.uhg.upm3.framework</groupId>
        <version>1.0.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice.optumrx</groupId>
    <artifactId>dataservice-optumrx-SubmitMemberTrailClaim-v1</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>bundle</packaging>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.felix</groupId>
                <artifactId>maven-bundle-plugin</artifactId>
                <configuration>
                    <instructions>
                        <Bundle-Version>${version}.${build.number}</Bundle-Version>
                        <Bundle-Name>UPM3 - ${pom.artifactId}</Bundle-Name>
                        <Export-Package>com.uhg.upm.dataservice.optumrx.submitmembertrailclaim;ve
                        <Import-Package>org.springframework.aop,org.springframework.aop.framework
                            com.uhg.upm.core.base;version="[1,2)",com.uhg.upm.core.exception;vers
                            com.uhg.upm.core.interceptor;version="[1,2)",com.uhg.upm.core.util;ve
                            com.uhg.upm.core.logging;version="[1,2)",com.uhg.upm.core.configurati
                            com.uhg.upm.core.security;version="[1,2)",com.uhg.upm.core.security.a
                            *
                        </Import-Package>
                        <Import-Library>org.aspectj</Import-Library>
                    </instructions>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                    <artifactId>jaxb2-maven-plugin</artifactId>
                    <version>1.3</version>
                    <executions>
                        <execution>
                            <phase>generate-sources</phase>
                            <goals>
                                <goal>xjc</goal>
                            </goals>
                        </execution>
```

9. Select "Maven" and check the box for "Create from archetype". Select the com.uhg.upm3.archetypes:upm3-dataservice-wmb archetype and click "Next". For the Project SDK: make sure that "1.6" is selected.



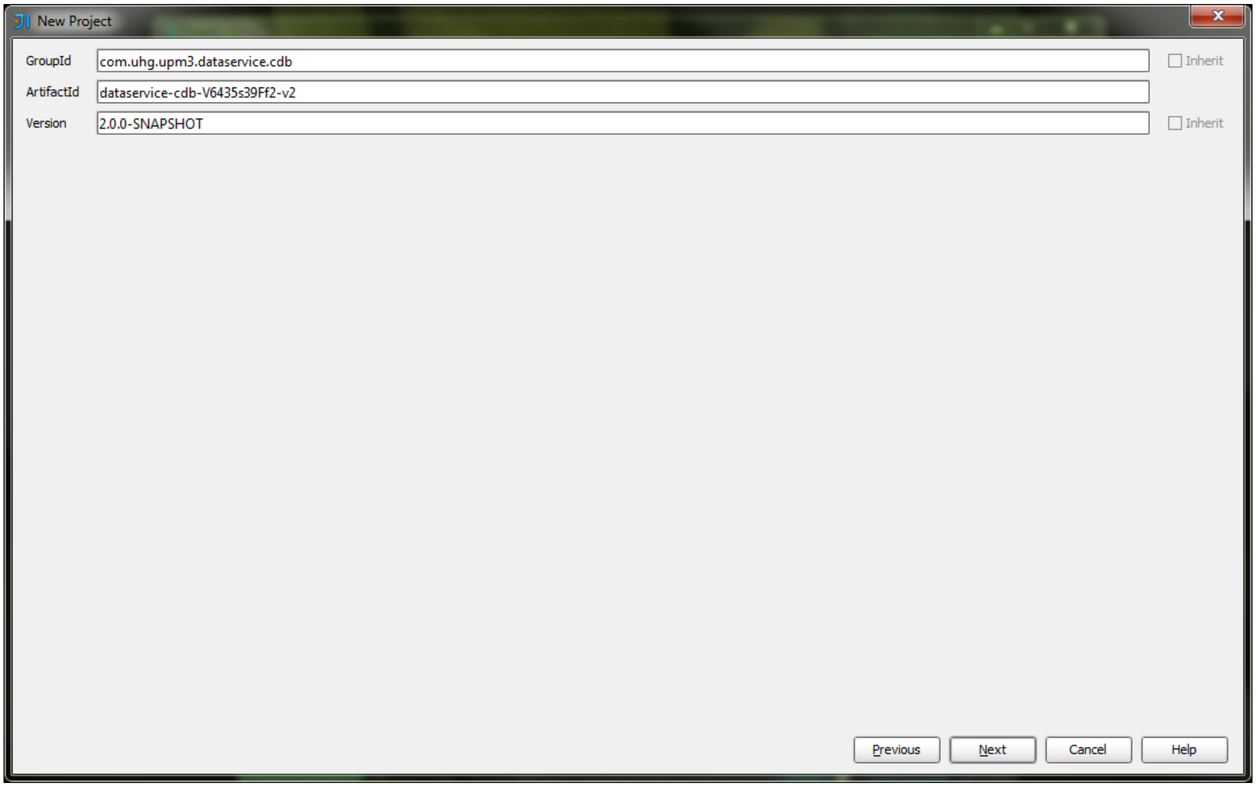
Project SDK: 1.6 (java version "1.6.0_31")

☑ Create from archetype

- com.atlassian.maven.archetypes:confluence-plugin-archetype
- com.atlassian.maven.archetypes:jira-plugin-archetype
- com.rfc.maven.archetypes:jpa-maven-archetype
- com.uhg.upm3.archetypes:upm3-dataservice-rest
- com.uhg.upm3.archetypes:upm3-dataservice-web
- com.uhg.upm3.archetypes:upm3-dataservice-wmb
- com.uhg.upm3.archetypes:upm3-service-rest
- com.uhg.upm3.archetypes:upm3-service-soap
- net.liftweb:lift-archetype-basic
- net.liftweb:lift-archetype-blank
- net.sf.maven-har:maven-archetype-har
- net.sf.maven-sar:maven-archetype-sar
- org.apache.cocoon:cocoon-22-archetype-block
- org.apache.cocoon:cocoon-22-archetype-block-plain
- org.apache.cocoon:cocoon-22-archetype-webapp
- org.apache.maven.archetypes:maven-archetype-j2ee-simple
- org.apache.maven.archetypes:maven-archetype-marmalade-mojo
- org.apache.maven.archetypes:maven-archetype-mojo
- org.apache.maven.archetypes:maven-archetype-portlet
- org.apache.maven.archetypes:maven-archetype-profiles
- org.apache.maven.archetypes:maven-archetype-quickstart
- org.apache.maven.archetypes:maven-archetype-site
- org.apache.maven.archetypes:maven-archetype-site-simple
- org.apache.maven.archetypes:maven-archetype-webapp
- org.apache.maven.archetypes:softeu-archetype-jsf

10. Set the maven artifact values:

a. GroupId should be com.uhg.upm3.dataservice.[backend] (eg. com.uhg.upm3.dataservice.cdb)
b. *ArtifactId should be* dataservice-[backend]-[MessageBrokerServiceName]-v[MessageBrokerVersion] (eg. dataservice-cdb-V6435s39Ff2-v2)
c. Version should correspond to the Message Broker version number with SNAPSHOT (eg. 1.0.0-SNAPSHOT)



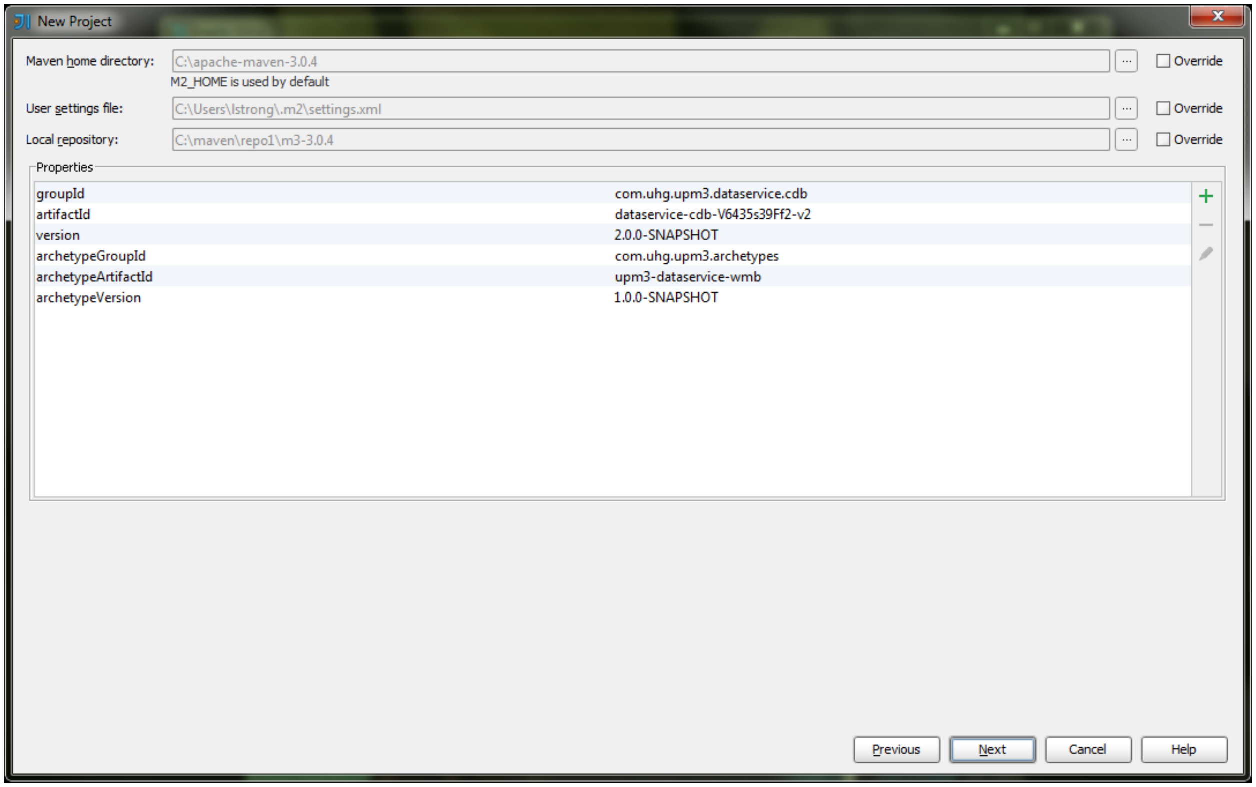
GroupId: com.uhg.upm3.dataservice.cdb ☐ Inherit

ArtifactId: dataservice-cdb-V6435s39Ff2-v2

Version: 2.0.0-SNAPSHOT ☐ Inherit

11. Click Next.

12. You will see a summary of the properties being used to create the project/module.



Maven home directory: C:\apache-maven-3.0.4 □ Override

M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml □ Override

Local repository: C:\maven\repo1\m3-3.0.4 □ Override

Properties

groupId	com.uhg.upm3.dataservice.cdb
artifactId	dataservice-cdb-V6435s39Ff2-v2
version	2.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-dataservice-wmb
archetypeVersion	1.0.0-SNAPSHOT

13. Click “Next” again and enter the following values:

    a. For Project name: Enter the name of the project that you just created on your local filesystem:
    dataservice-cdb-V6435s11Ff2

    b. For Project location: Click on the … and navigate to the folder that you just created.

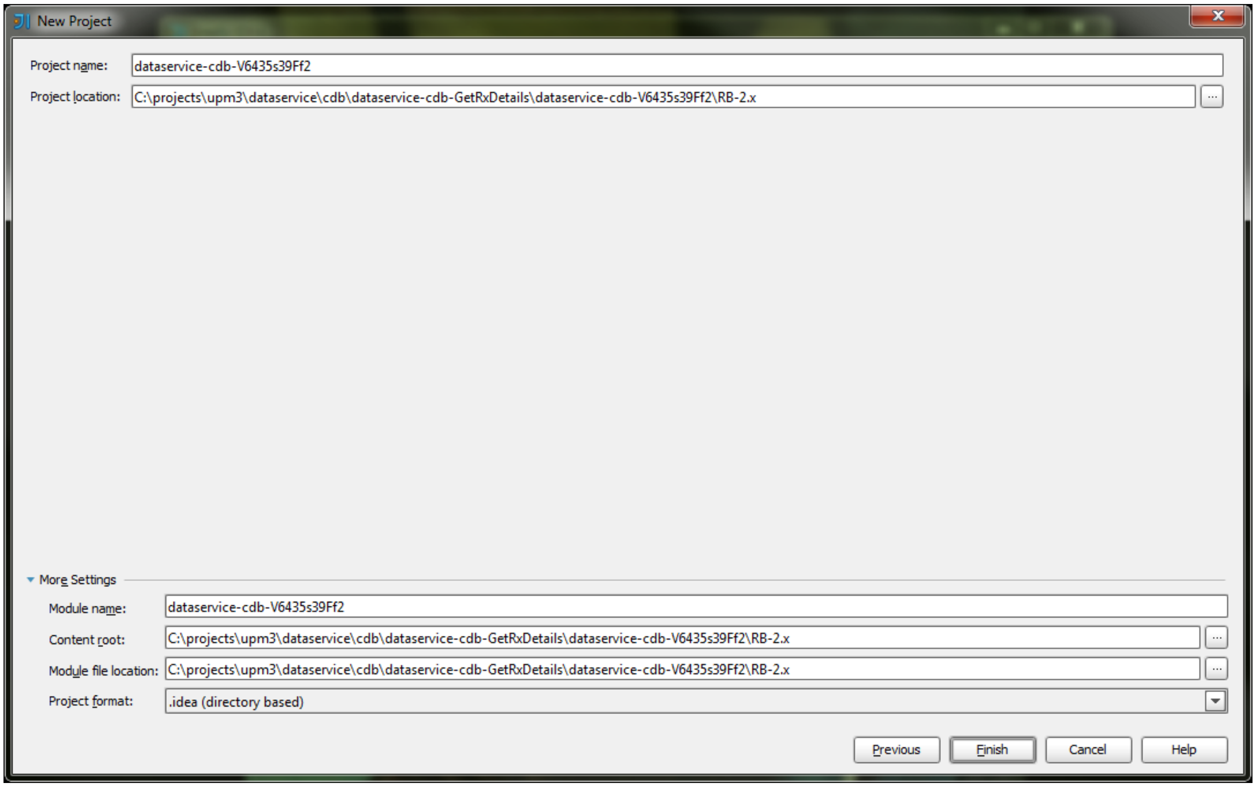
14. Expand the “More Settings” section at the bottom of the window and fill in the following values:

    a. Module name: Copy/Paste the Project name: dataservice-cdb-V6435s11Ff2

    b. Content Root: Should match what is in Project files location. If not, then Copy/Paste it here:
    C:\projects\upm3\dataservice\cdb\dataservice-cdb-V6435s11Ff2\RB-1.x

    c. Module file location: Should match what is in Project files location. If not, then Copy/Paste it here:
    C:\projects\upm3\dataservice\cdb\dataservice-cdb-V6435s11Ff2\RB-1.x

    d. Project file format: .idea (directory based) should be selected.



New Project

Project name: dataservice-cdb-V6435s39Ff2

Project location: C:\projects\upm3\dataservice\cdb\dataservice-cdb-GetRxDetails\dataservice-cdb-V6435s39Ff2\RB-2.x

More Settings

Module name: dataservice-cdb-V6435s39Ff2

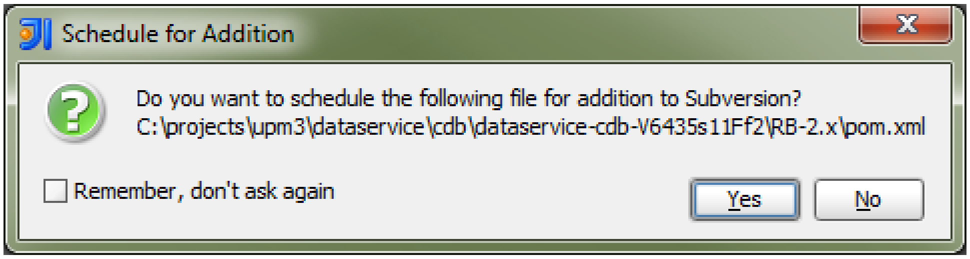
Content root: C:\projects\upm3\dataservice\cdb\dataservice-cdb-GetRxDetails\dataservice-cdb-V6435s39Ff2\RB-2.x

Module file location: C:\projects\upm3\dataservice\cdb\dataservice-cdb-GetRxDetails\dataservice-cdb-V6435s39Ff2\RB-2.x

Project format: .idea (directory based)

15. Click Finish.

16. Intellij will ask if you would like to add the pom.xml to Subversion. Click Yes.



Schedule for Addition

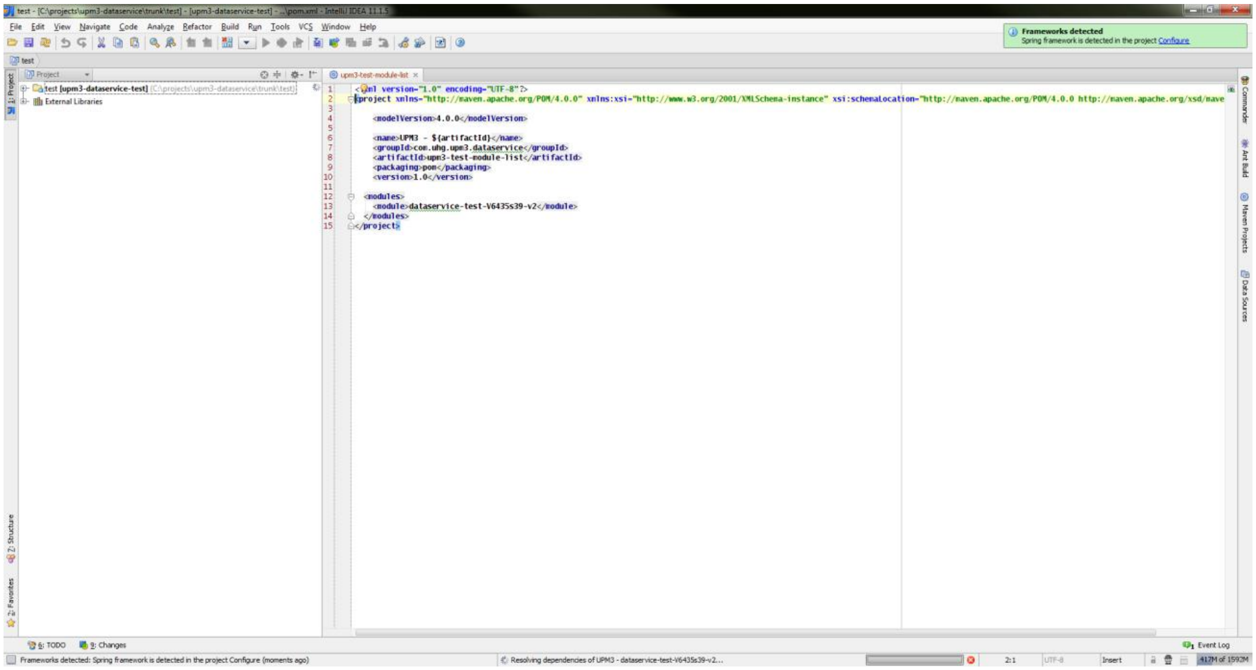
Do you want to schedule the following file for addition to Subversion?
C:\projects\upm3\dataservice\cdb\dataservice-cdb-V6435s11Ff2\RB-2.x\pom.xml

☐ Remember, don't ask again

17. Intellij will then create the project from the archetype. If everything was successful you will see a message in the Messages window in Intellij.

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.919s
[INFO] Finished at: Wed Feb 19 12:42:11 EST 2014
[INFO] Final Memory: 6M/15M
[INFO] ------------------------------------------------------------------------
[INFO] Maven execution finished
```

18. You will briefly get a popup in the top right corner of your Intellij window that shows a Spring Framework has been detected. Click configure and accept the defaults.


test - [C:\projects\upm3-dataservice\trunk\test] - [upm3-dataservice-test] - ...\pom.xml - IntelliJ IDEA 11.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
test
Project
test [upm3-dataservice-test] (C:\projects\upm3-dataservice\trunk\test)
External Libraries
upm3-test-module-list
1: Project
7: Structure
2: Favorites
Commander
Ant Build
Maven Projects
Data Sources
6: TODO
9: Changes
Event Log
2:1
UTF-8
Insert
417M of 1592M

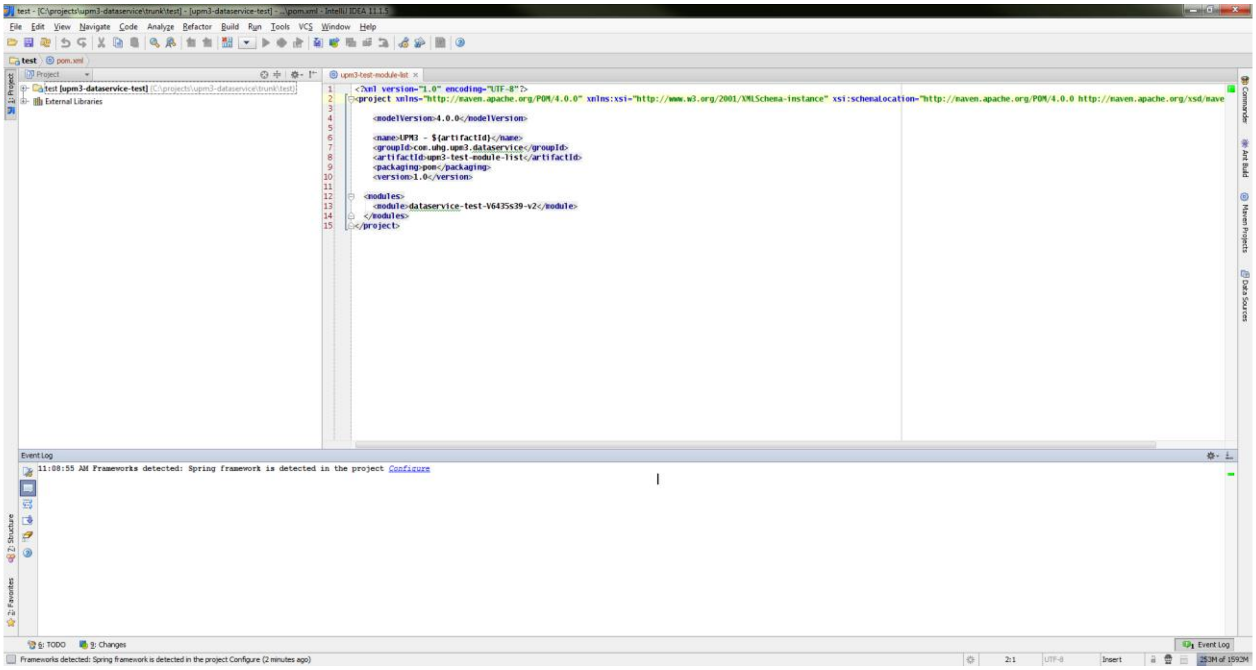
Frameworks detected
Spring framework is detected in the project Configure

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

Frameworks detected: Spring framework is detected in the project Configure (moments ago)

Resolving dependencies of UPM3 - dataservice-test-V6435s39-v2...

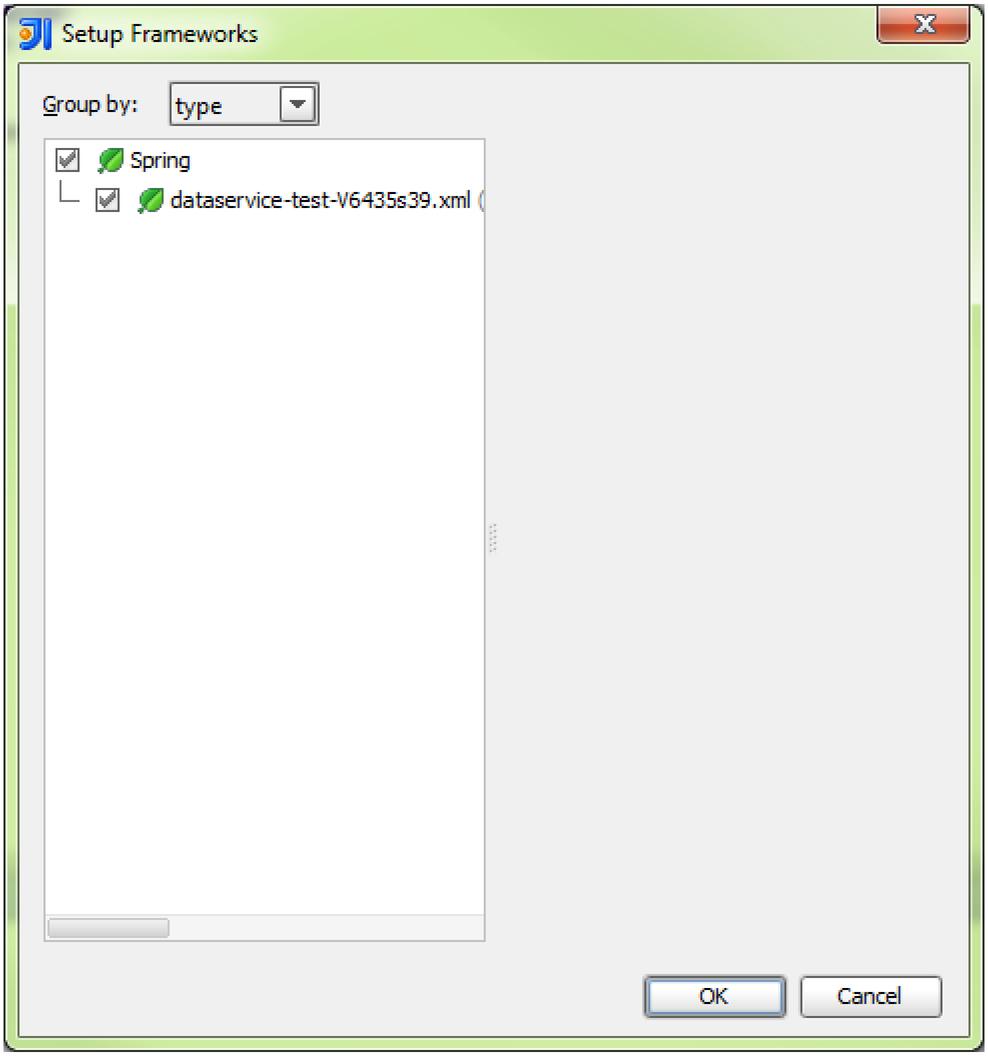
If the popup disappears before you are able to click it then open the Event Log in the bottom right corner of Intellij and click on Configure. It will bring up a Setup Frameworks dialog box. Click OK.


test - [C:\projects\upm3-dataservice\trunk\test] - [upm3-dataservice-test] - ...\pom.xml - IntelliJ IDEA 11.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
test
pom.xml
Project
test [upm3-dataservice-test] (C:\projects\upm3-dataservice\trunk\test)
External Libraries
upm3-test-module-list
Event Log
6: TODO
9: Changes
Event Log
2:1
UTF-8
Insert
253M of 1592M

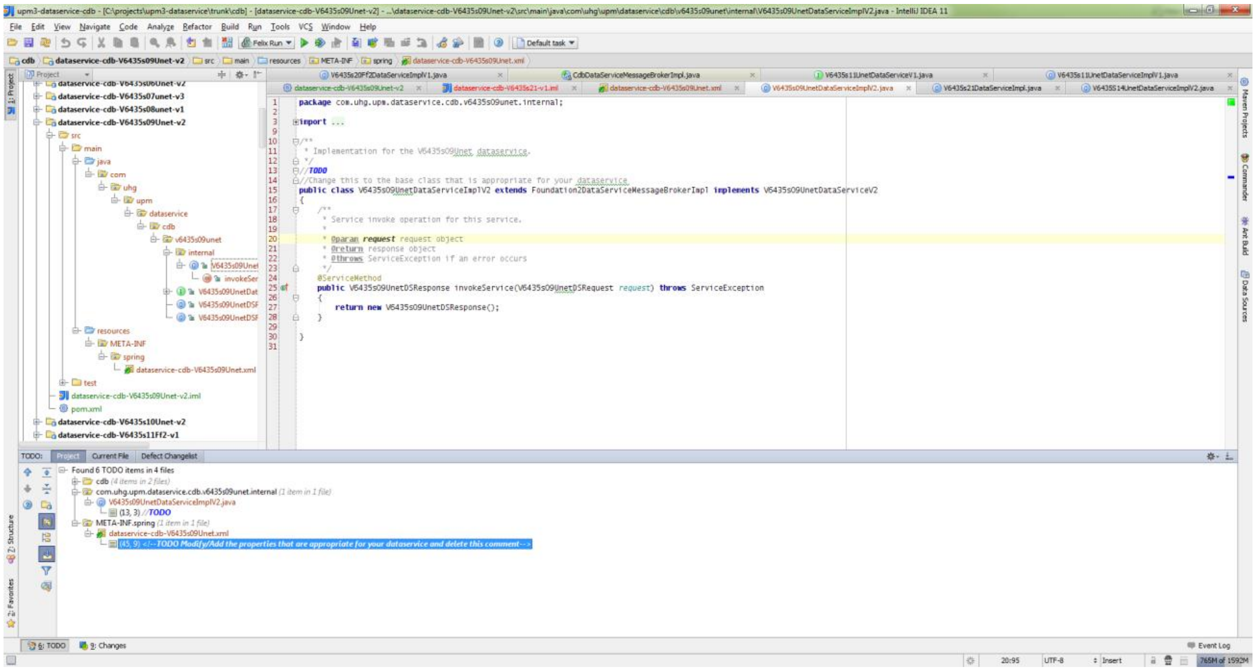
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

```
11:08:55 AM Frameworks detected: Spring framework is detected in the project Configure
```

Frameworks detected: Spring framework is detected in the project Configure (2 minutes ago)



19. You will find TODO statements in the data service impl class and the spring xml file. You should modify the impl class to extend the base class that is appropriate for your dataservice and modify the spring properties accordingly. You can also find these TODOs by clicking on the TODO tab in the bottom left corner of Intellij.



```
package com.uhg.upm.dataservice.cdb.v6435s09unet.internal;

import ...

/**
 * Implementation for the V6435s09Unet dataservice.
 */
//TODO
//Change this to the base class that is appropriate for your dataservice
public class V6435s09UnetDataServiceImplV2 extends Foundation2DataServiceMessageBrokerImpl implements V6435s09UnetDataServiceV2
{
    /**
     * Service invoke operation for this service.
     *
     * @param request request object
     * @return response object
     * @throws ServiceException if an error occurs
     */
    @ServiceMethod
    public V6435s09UnetDSResponse invokeService(V6435s09UnetDSRequest request) throws ServiceException
    {
        return new V6435s09UnetDSResponse();
    }

}
```

```
Found 6 TODO items in 4 files
  cdb (4 items in 2 files)
  com.uhg.upm.dataservice.cdb.v6435s09unet.internal (1 item in 1 file)
    V6435s09UnetDataServiceImplV2.java
      (13, 3) //TODO
  META-INF.spring (1 item in 1 file)
    dataservice-cdb-V6435s09Unet.xml
      (45, 9) <!--TODO Modify/Add the properties that are appropriate for your dataservice and delete this comment-->
```

20. Add the Unversioned files to Subversion.
    - If svn brings up any ./out* or ./target* directories DO NOT commit them. Please add them to .svnignore (TortoiseSVN->"Unversion and add to ignore list").
      These directories are used by Maven and Felix as working directories and have the compiled classes and bundle jar files. These directories are not needed and will just add a lot of overhead to commits and updates as they are constantly changing.
    - In the directory ./.idea/ add the file "workspace.xml to .svnignore.
    - In the directory ./.idea/ add the "./libraries" directory to .svnignore.

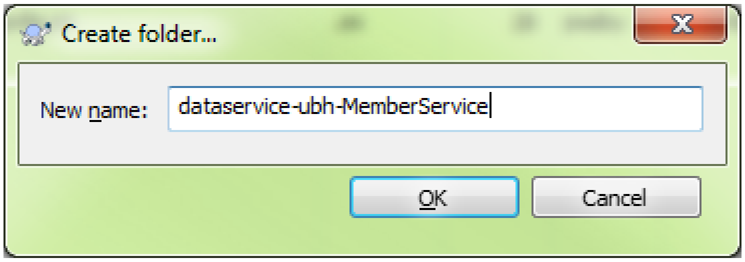
21. Once you have committed your new project it will need to be added to the build farm setup. If you have access, add your project, otherwise send an email to your dev lead or dev manager with the details for your new module, make sure to make note of any dependencies your module has so the build farm can be configured correctly.

Web Data Service Project Creation

1. **If this is your first time using maven archetypes and you haven't already set them up for use in Intellij, please follow the instructions for Initial Setup.**

2. Open TortoiseSVN repo browser (right-click anywhere in Windows Explorer and select TortoiseSVN/Repo-browser) to: http://svn.uhc.com/svn/uhgit_upm3

3. Create the new project at the root level of the repository by selecting "Create Folder…" (do not select "Add Folder…"). Name the project without the version (leave off –v1, etc). Naming convention should be dataservice-[backend]-[WSDLName]

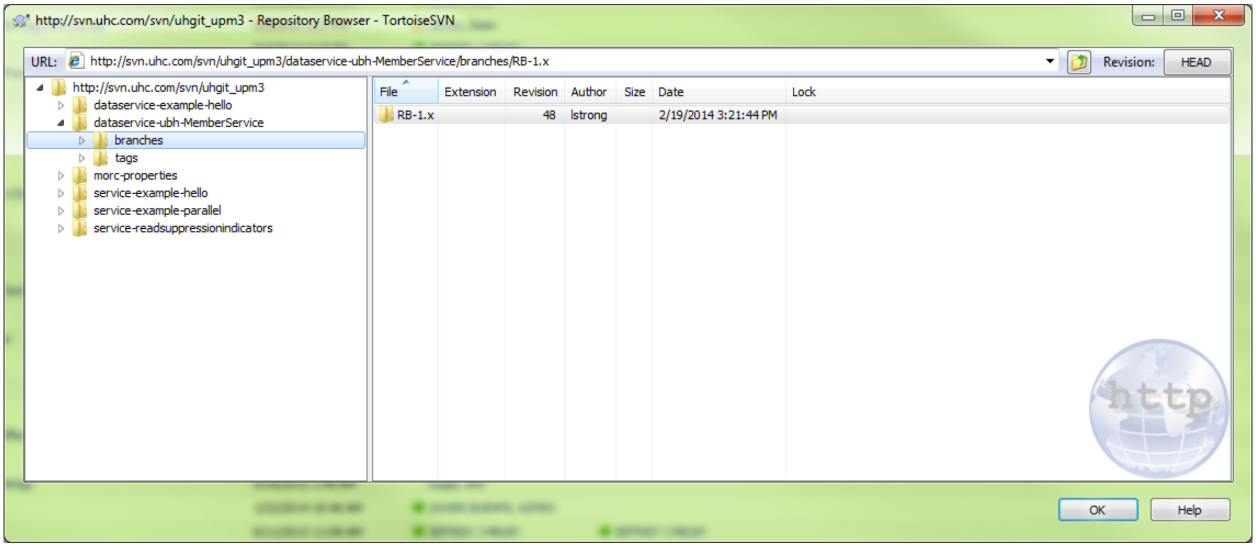


Create folder...

New name: dataservice-ubh-MemberService

4. Under that directory, Create 2 folders: “branches” and “tags”.

5. Under “branches”, create the release branch for the version(s):
   RB-1.x, RB-2.x, etc. (note the “x” is intentional—don’t replace it) Version should correspond to the backend WSDL version number.



http://svn.uhc.com/svn/uhgit_upm3 - Repository Browser - TortoiseSVN

URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-ubh-MemberService/branches/RB-1.x

Revision: HEAD

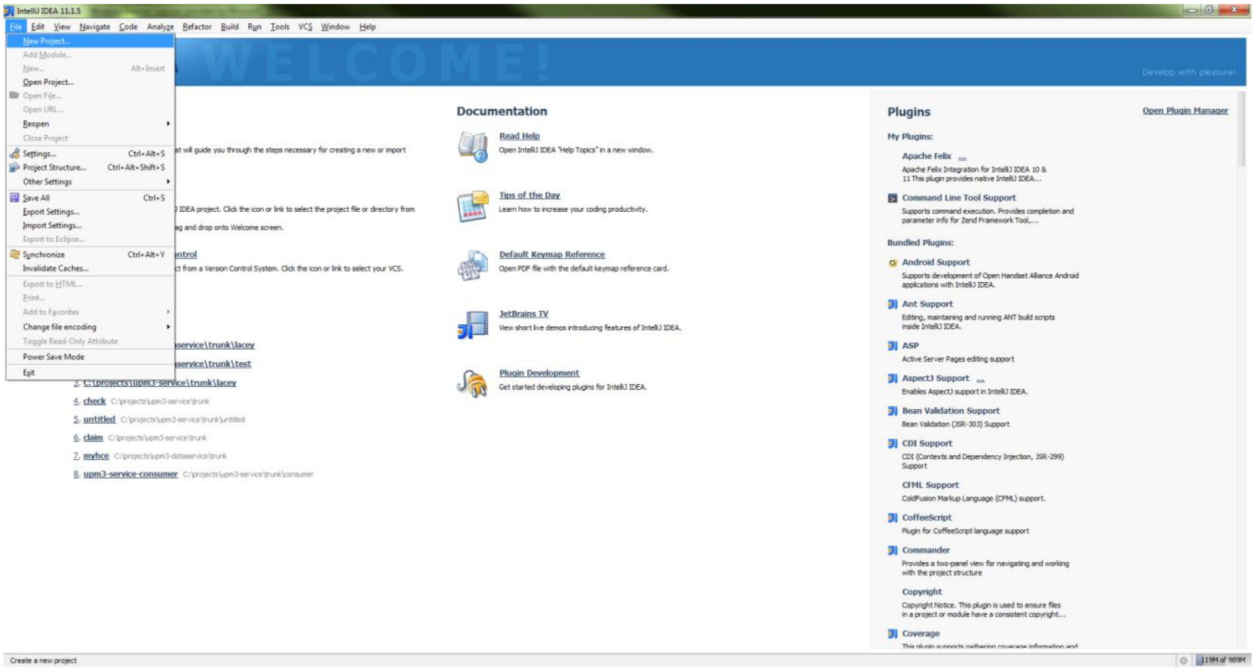
- http://svn.uhc.com/svn/uhgit_upm3
  - dataservice-example-hello
  - dataservice-ubh-MemberService
    - branches
    - tags
  - morc-properties
  - service-example-hello
  - service-example-parallel
  - service-readsuppressionindicators

File	Extension	Revision	Author	Size	Date	Lock
RB-1.x		48	lstrong		2/19/2014 3:21:44 PM	

6. On local filesystem, create the project and branch directories:
   C:\projects\upm3\dataservice\ubh\dataservice-ubh-MemberService\RB-1.x

7. Checkout from SVN to that directory
   Repo URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-ubh-MemberService/branches/RB-1.x/
   Local directory: C:\projects\upm3\dataservice\ubh\dataservice-ubh-MemberService\RB-1.x

8. Open Intellij and select File→New Project…



at will guide you through the steps necessary for creating a new or import

) IDEA project. Click the icon or link to select the project file or directory from

ag and drop onto Welcome screen.

ntrol

ct from a Version Control System. Click the icon or link to select your VCS.

service\trunk\lacey

service\trunk\test

3. C:\projects\upm3-service\trunk\lacey

4. check C:\projects\upm3-service\trunk

5. untitled C:\projects\upm3-service\trunk\untitled

6. claim C:\projects\upm3-service\trunk

7. myhce C:\projects\upm3-dataservice\trunk

8. upm3-service-consumer C:\projects\upm3-service\trunk\consumer

Read Help: Open IntelliJ IDEA "Help Topics" in a new window.

Tips of the Day: Learn how to increase your coding productivity.

Default Keymap Reference: Open PDF file with the default keymap reference card.

JetBrains TV: View short live demos introducing features of IntelliJ IDEA.

Plugin Development: Get started developing plugins for IntelliJ IDEA.

Apache Felix ...
Apache Felix Integration for IntelliJ IDEA 10 & 11 This plugin provides native IntelliJ IDEA...

Command Line Tool Support
Supports command execution. Provides completion and parameter info for Zend Framework Tool,...

Android Support
Supports development of Open Handset Alliance Android applications with IntelliJ IDEA.

Ant Support
Editing, maintaining and running ANT build scripts inside IntelliJ IDEA.

ASP
Active Server Pages editing support

AspectJ Support ...
Enables AspectJ support in IntelliJ IDEA.

Bean Validation Support
Bean Validation (JSR-303) Support

CDI Support
CDI (Contexts and Dependency Injection, JSR-299) Support

CFML Support
ColdFusion Markup Language (CFML) support.

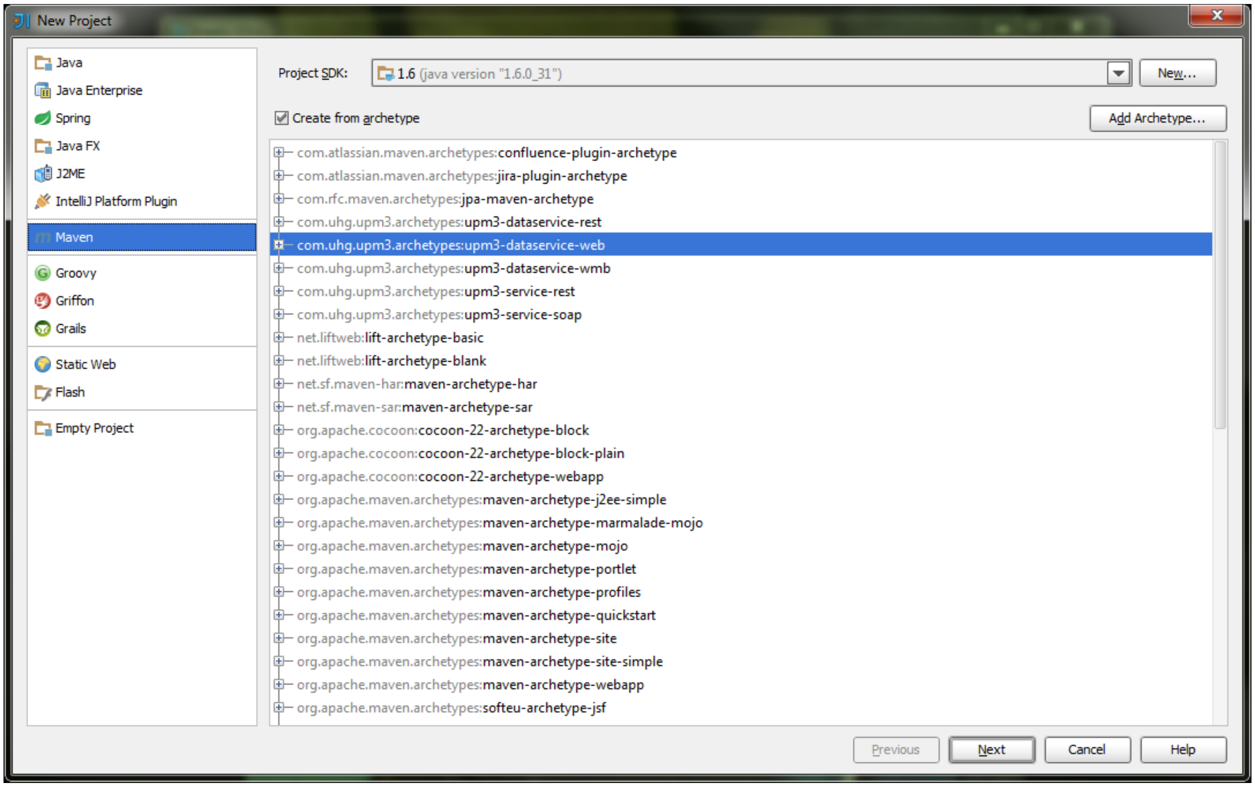
CoffeeScript
Plugin for CoffeeScript language support

Commander
Provides a two-panel view for navigating and working with the project structure

Copyright
Copyright Notice. This plugin is used to ensure files in a project or module have a consistent copyright...

Coverage
This plugin supports gathering coverage information and

9. Select "Maven" and check the box for "Create from archetype". Select the com.uhg.upm3.archetypes:upm3-dataservice-web archetype and click "Next". For the Project SDK: make sure that "1.6" is selected.



Project SDK: 1.6 (java version "1.6.0_31")

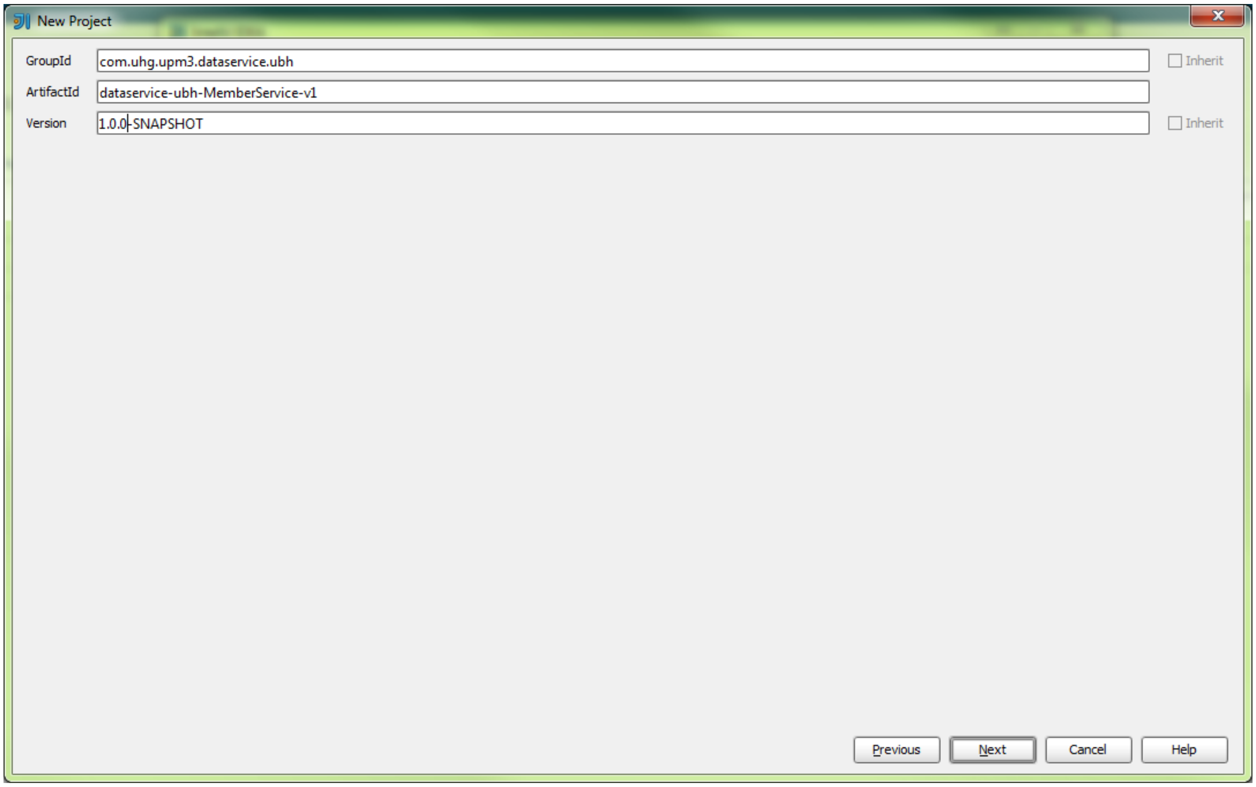
☑ Create from archetype

- com.atlassian.maven.archetypes:confluence-plugin-archetype
- com.atlassian.maven.archetypes:jira-plugin-archetype
- com.rfc.maven.archetypes:jpa-maven-archetype
- com.uhg.upm3.archetypes:upm3-dataservice-rest
- com.uhg.upm3.archetypes:upm3-dataservice-web
- com.uhg.upm3.archetypes:upm3-dataservice-wmb
- com.uhg.upm3.archetypes:upm3-service-rest
- com.uhg.upm3.archetypes:upm3-service-soap
- net.liftweb:lift-archetype-basic
- net.liftweb:lift-archetype-blank
- net.sf.maven-har:maven-archetype-har
- net.sf.maven-sar:maven-archetype-sar
- org.apache.cocoon:cocoon-22-archetype-block
- org.apache.cocoon:cocoon-22-archetype-block-plain
- org.apache.cocoon:cocoon-22-archetype-webapp
- org.apache.maven.archetypes:maven-archetype-j2ee-simple
- org.apache.maven.archetypes:maven-archetype-marmalade-mojo
- org.apache.maven.archetypes:maven-archetype-mojo
- org.apache.maven.archetypes:maven-archetype-portlet
- org.apache.maven.archetypes:maven-archetype-profiles
- org.apache.maven.archetypes:maven-archetype-quickstart
- org.apache.maven.archetypes:maven-archetype-site
- org.apache.maven.archetypes:maven-archetype-site-simple
- org.apache.maven.archetypes:maven-archetype-webapp
- org.apache.maven.archetypes:softeu-archetype-jsf

10. Set the maven artifact values:
    a. GroupId should be com.uhg.upm3.dataservice.[backend] (eg. com.uhg.upm3.dataservice.ubh)

    b. ArtifactId should be dataservice-[backend]-[WSDL Name]-v[WSDL Version] (eg. dataservice-ubh-MemberService-v1)

    c. Version should correspond to the WSDL version number of the service with SNAPSHOT (eg. 1.0.0-SNAPSHOT)



New Project

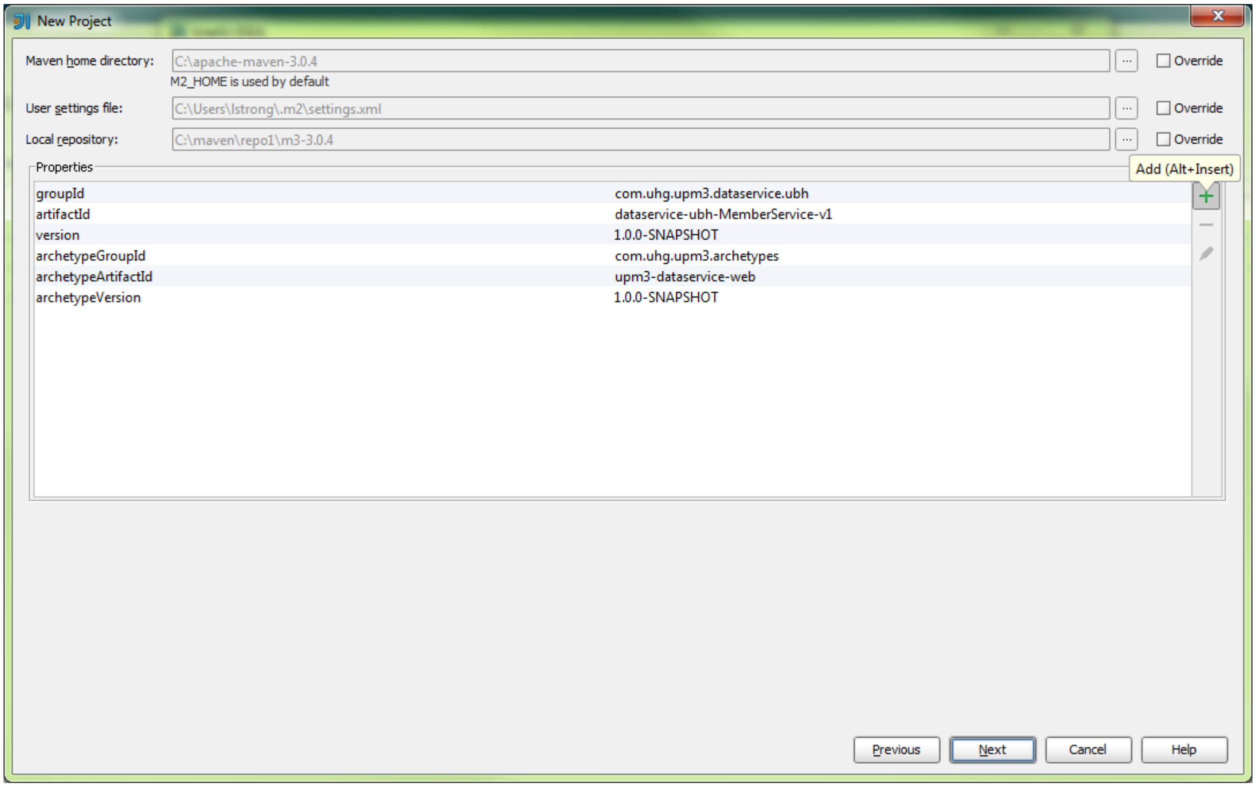
GroupId: com.uhg.upm3.dataservice.ubh □ Inherit

ArtifactId: dataservice-ubh-MemberService-v1

Version: 1.0.0-SNAPSHOT □ Inherit

11. Click next.

12. Click on the + button in the properties box. This will bring up the Edit Maven Property dialog box.



Maven home directory: C:\apache-maven-3.0.4 ☐ Override

M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml ☐ Override

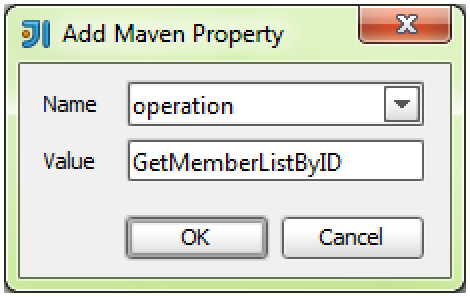
Local repository: C:\maven\repo1\m3-3.0.4 ☐ Override

Properties

groupId	com.uhg.upm3.dataservice.ubh
artifactId	dataservice-ubh-MemberService-v1
version	1.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-dataservice-web
archetypeVersion	1.0.0-SNAPSHOT

13. In the Name box type ‘operation’

14. In the Value box type the name of the WSDL operation.

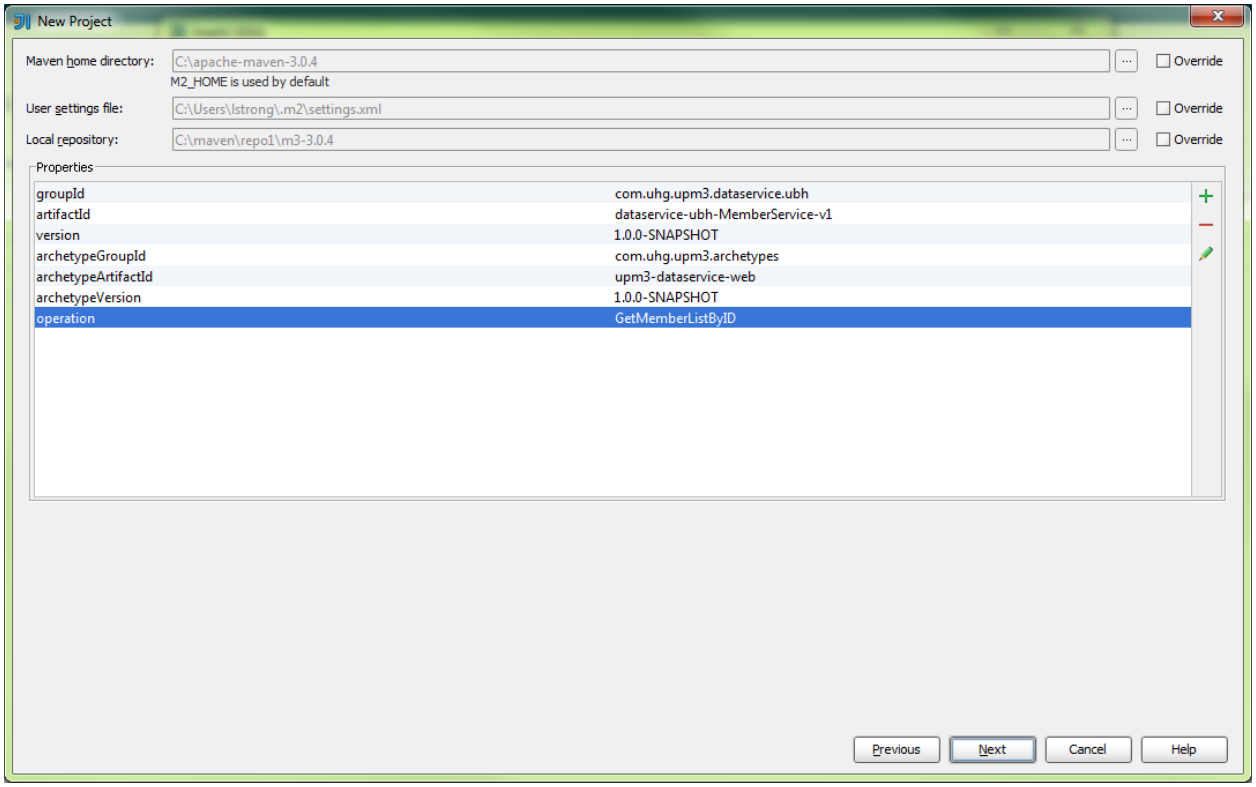


Name: operation

Value: GetMemberListByID

15. Click OK

16. You will see a summary of the properties being used to create the project/module.



New Project

Maven home directory: C:\apache-maven-3.0.4 ☐ Override

M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml ☐ Override

Local repository: C:\maven\repo1\m3-3.0.4 ☐ Override

Properties

groupId	com.uhg.upm3.dataservice.ubh
artifactId	dataservice-ubh-MemberService-v1
version	1.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-dataservice-web
archetypeVersion	1.0.0-SNAPSHOT
operation	GetMemberListByID

17. Click "Next" again and enter the following values:

    a. For Project name: Enter the name of the project that you just created on your local filesystem:
       dataservice-ubh-MemberService

    b. For Project location: Click on the … and navigate to the folder that you just created.

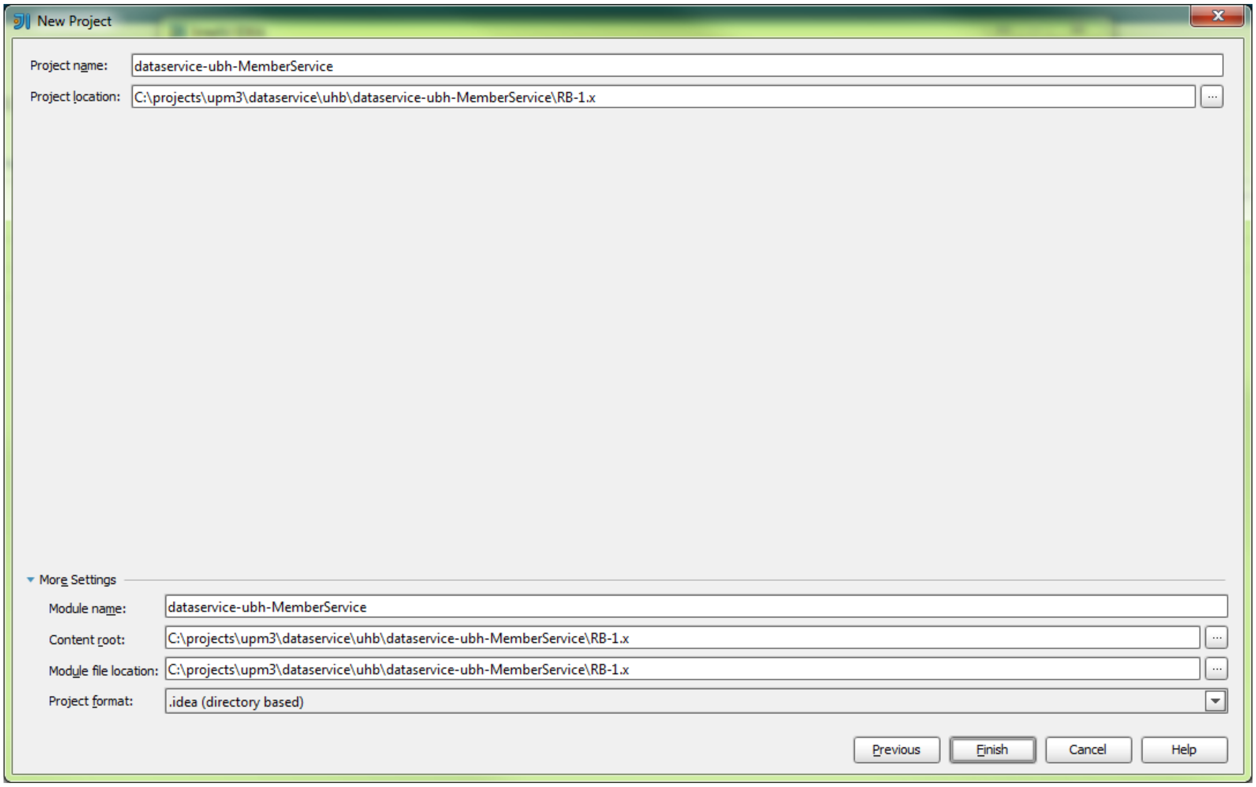
18. Expand the "More Settings" section at the bottom of the window and fill in the following values:

    a. Module name: Copy/Paste the Project name: dataservice-ubh-MemberService

    b. Content Root: Should match what is in Project files location. If not, then Copy/Paste it here:
       C:\projects\upm3\dataservice\ubh\dataservice-ubh-MemberService\RB-1.x

       Module file location: Should match what is in Project files location. If not, then Copy/Paste it here:
       C:\projects\upm3\dataservice\ubh\dataservice-ubh-MemberService\RB-1.x

    c. Project file format: .idea (directory based) should be selected.



Project name: dataservice-ubh-MemberService

Project location: C:\projects\upm3\dataservice\uhb\dataservice-ubh-MemberService\RB-1.x

More Settings

Module name: dataservice-ubh-MemberService

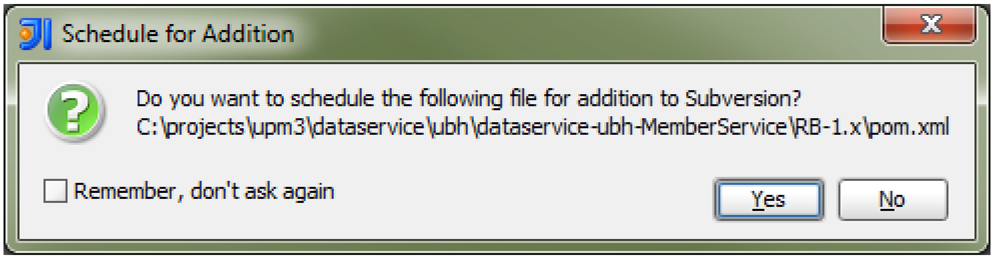
Content root: C:\projects\upm3\dataservice\uhb\dataservice-ubh-MemberService\RB-1.x

Module file location: C:\projects\upm3\dataservice\uhb\dataservice-ubh-MemberService\RB-1.x

Project format: .idea (directory based)

19. Click Finish.

20. Intellij will ask if you would like to add the pom.xml to Subversion. Click Yes.



Schedule for Addition

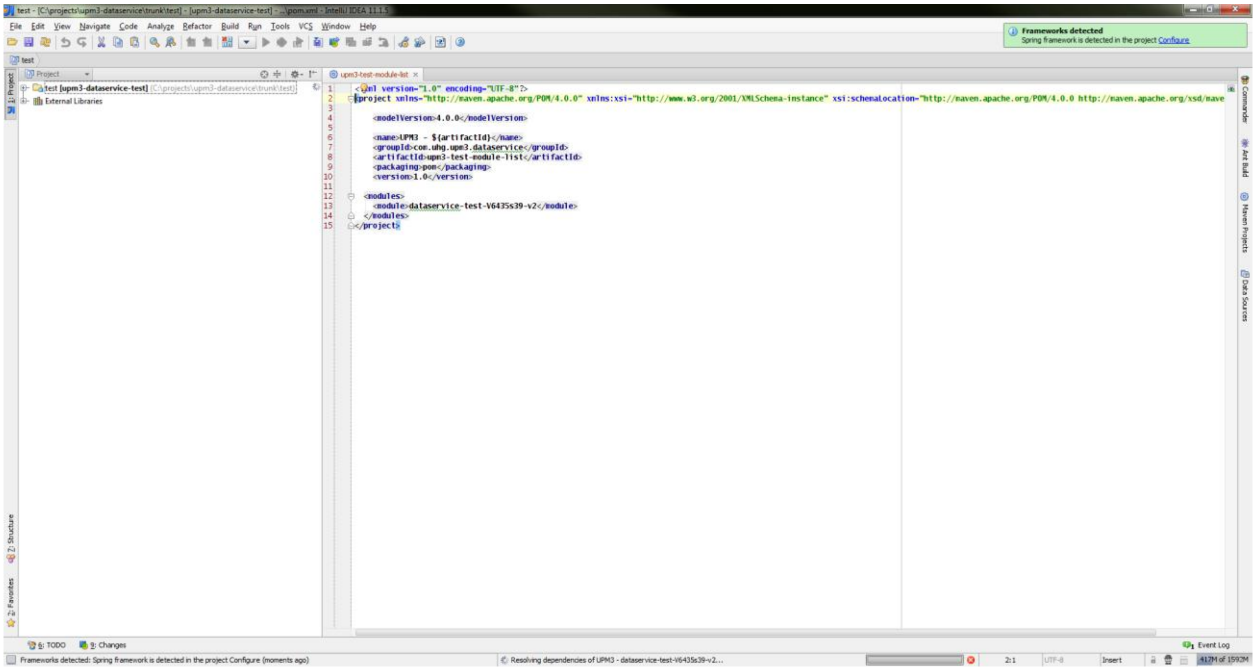
Do you want to schedule the following file for addition to Subversion?
C:\projects\upm3\dataservice\ubh\dataservice-ubh-MemberService\RB-1.x\pom.xml

☐ Remember, don't ask again

21. Intellij will then create the project from the archetype. If everything was successful you will see a message in the Messages window in Intellij.

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.919s
[INFO] Finished at: Wed Feb 19 12:42:11 EST 2014
[INFO] Final Memory: 6M/15M
[INFO] ------------------------------------------------------------------------
[INFO] Maven execution finished
```

22. You will briefly get a popup in the top right corner of your Intellij window that shows a Spring Framework has been detected. Click configure and accept the defaults.



test - [C:\projects\upm3-dataservice\trunk\test] - [upm3-dataservice-test] - ...\pom.xml - IntelliJ IDEA 11.1.5

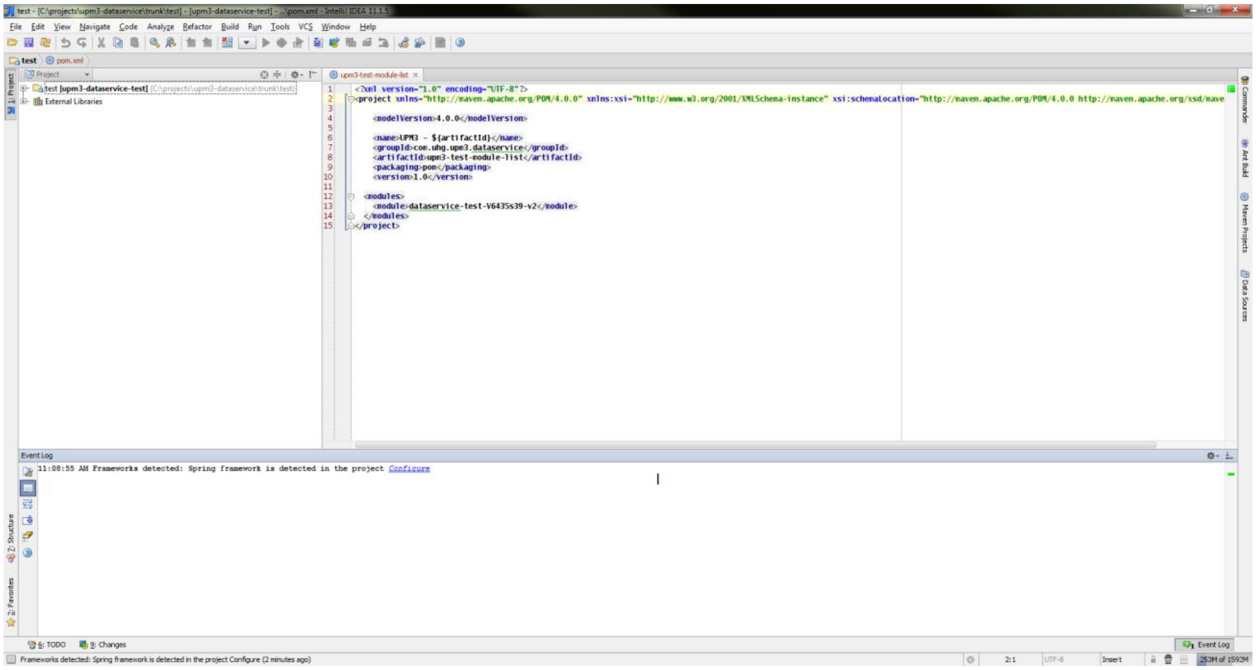
Frameworks detected
Spring framework is detected in the project Configure

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

Frameworks detected: Spring framework is detected in the project Configure (moments ago)

Resolving dependencies of UPM3 - dataservice-test-V6435s39-v2...

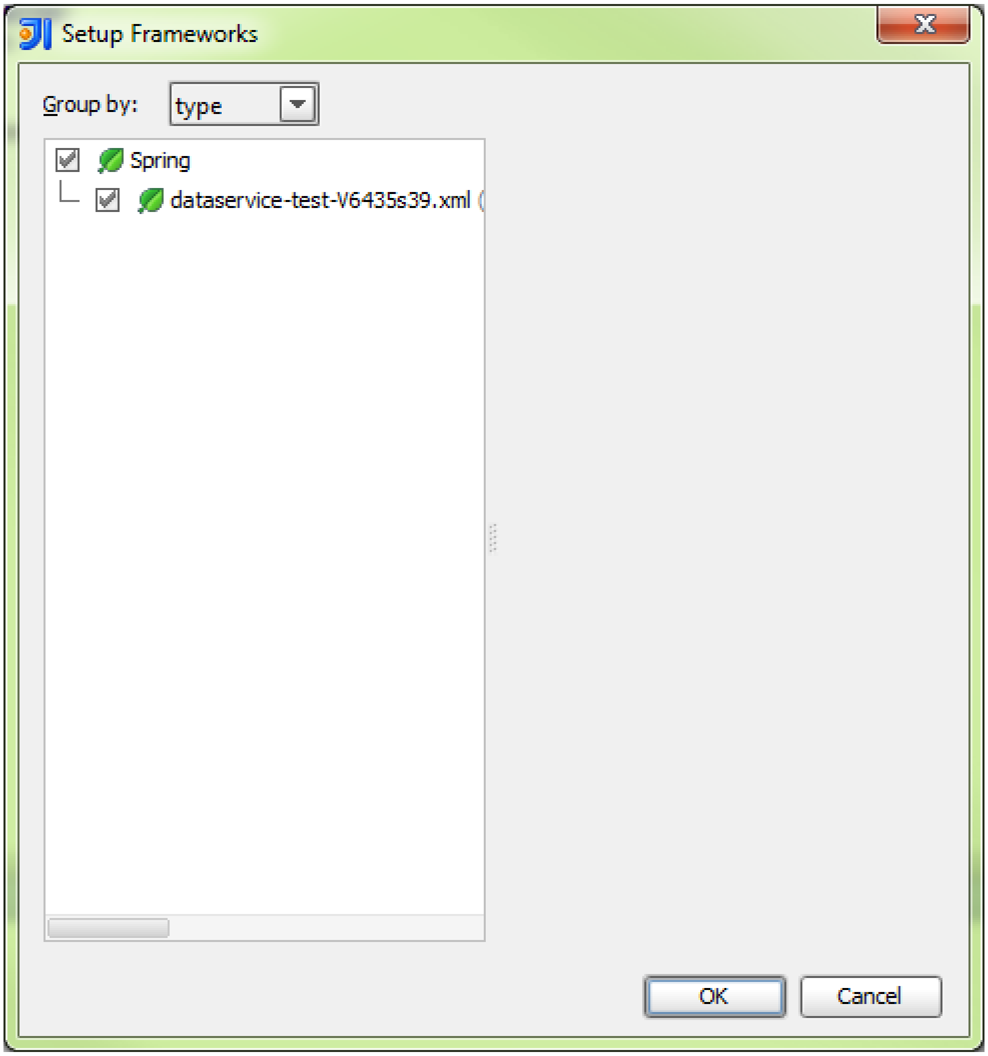
If the popup disappears before you are able to click it then open the Event Log in the bottom right corner of Intellij and click on Configure. It will bring up a Setup Frameworks dialog box. Click OK.


test - [C:\projects\upm3-dataservice\trunk\test] - [upm3-dataservice-test] - ...\pom.xml - IntelliJ IDEA 11.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
test
pom.xml
Project
test [upm3-dataservice-test] (C:\projects\upm3-dataservice\trunk\test)
External Libraries
upm3-test-module-list
1: Project
7: Structure
2: Favorites
Commander
Ant Build
Maven Projects
Data Sources
Event Log
6: TODO
9: Changes
Event Log
2:1
UTF-8
Insert
253M of 1592M

```
 <?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

```
11:08:55 AM Frameworks detected: Spring framework is detected in the project Configure
```

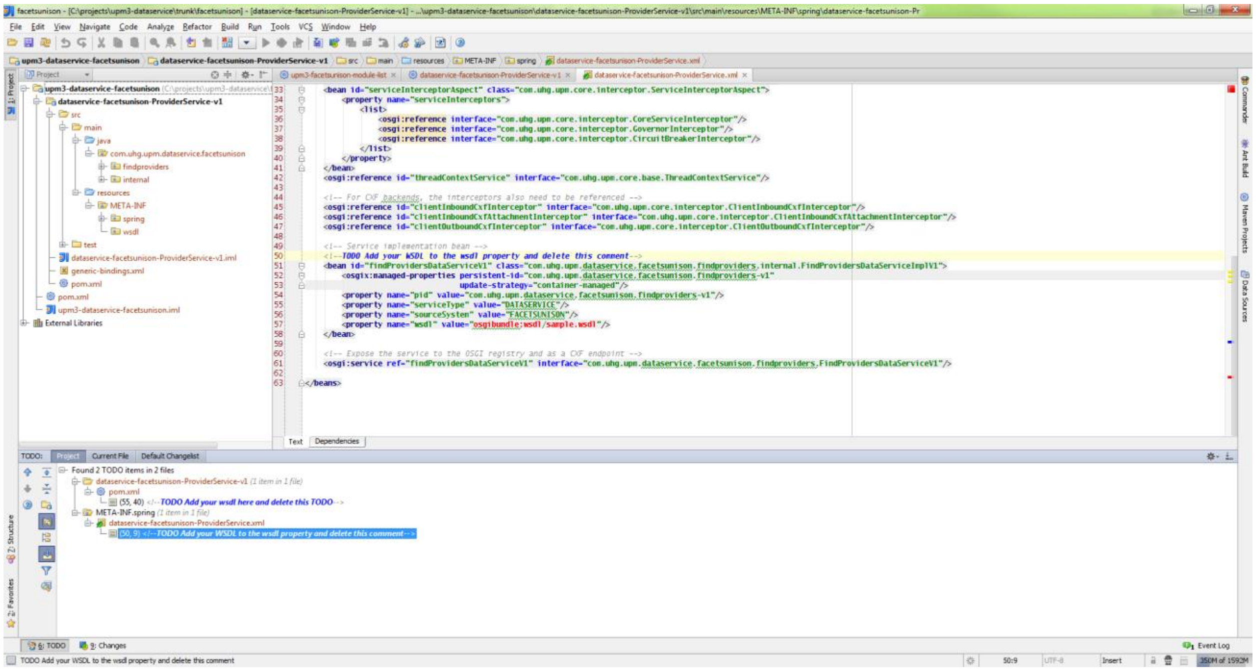
Frameworks detected: Spring framework is detected in the project Configure (2 minutes ago)



☑ Spring

☑ dataservice-test-V6435s39.xml (

23. You will find TODO statements in the module pom and the spring xml file. You should modify these files by adding your WSDL. You can also find these TODOs by clicking on the TODO tab in the bottom left corner of Intellij. You'll also notice that the base class in the internal folder has some compilation errors. Once you've added your WSDL and generated your stubs, you will need to add your port object to this class.


facetsunison - [C:\projects\upm3-dataservice\trunk\facetsunison] - [dataservice-facetsunison-ProviderService-v1] - ...\upm3-dataservice-facetsunison\dataservice-facetsunison-ProviderService-v1\src\main\resources\META-INF\spring\dataservice-facetsunison-Pr
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
upm3-dataservice-facetsunison
dataservice-facetsunison-ProviderService-v1
src
main
resources
META-INF
spring
dataservice-facetsunison-ProviderService.xml
Project
upm3-facetsunison-module-list
dataservice-facetsunison-ProviderService-v1
dataservice-facetsunison-ProviderService.xml
upm3-dataservice-facetsunison (C:\projects\upm3-dataservice\
dataservice-facetsunison-ProviderService-v1
src
main
java
com.uhg.upm.dataservice.facetsunison
findproviders
internal
resources
META-INF
spring
wsdl
test
dataservice-facetsunison-ProviderService-v1.iml
generic-bindings.xml
pom.xml
pom.xml
upm3-dataservice-facetsunison.iml
External Libraries
Text
Dependencies
TODO:
Project
Current File
Default Changelist
6: TODO
9: Changes
Event Log
1: Project
7: Structure
2: Favorites
Commander
Ant Build
Maven Projects
Data Sources
50:9
UTF-8
Insert
350M of 1592M

```
    <bean id="serviceInterceptorAspect" class="com.uhg.upm.core.interceptor.ServiceInterceptorAspect">
        <property name="serviceInterceptors">
            <list>
                <osgi:reference interface="com.uhg.upm.core.interceptor.CoreServiceInterceptor"/>
                <osgi:reference interface="com.uhg.upm.core.interceptor.GovernorInterceptor"/>
                <osgi:reference interface="com.uhg.upm.core.interceptor.CircuitBreakerInterceptor"/>
            </list>
        </property>
    </bean>
    <osgi:reference id="threadContextService" interface="com.uhg.upm.core.base.ThreadContextService"/>

    <!-- For CXF backends, the interceptors also need to be referenced -->
    <osgi:reference id="clientInboundCxfInterceptor" interface="com.uhg.upm.core.interceptor.ClientInboundCxfInterceptor"/>
    <osgi:reference id="clientInboundCxfAttachmentInterceptor" interface="com.uhg.upm.core.interceptor.ClientInboundCxfAttachmentInterceptor"/>
    <osgi:reference id="clientOutboundCxfInterceptor" interface="com.uhg.upm.core.interceptor.ClientOutboundCxfInterceptor"/>

    <!-- Service implementation bean -->
    <!--TODO Add your WSDL to the wsdl property and delete this comment-->
    <bean id="findProvidersDataServiceV1" class="com.uhg.upm.dataservice.facetsunison.findproviders.internal.FindProvidersDataServiceImplV1">
        <osgix:managed-properties persistent-id="com.uhg.upm.dataservice.facetsunison.findproviders-v1"
                                  update-strategy="container-managed"/>
        <property name="pid" value="com.uhg.upm.dataservice.facetsunison.findproviders-v1"/>
        <property name="serviceType" value="DATASERVICE"/>
        <property name="sourceSystem" value="FACETSUNISON"/>
        <property name="wsdl" value="osgibundle:wsdl/sample.wsdl"/>
    </bean>

    <!-- Expose the service to the OSGI registry and as a CXF endpoint -->
    <osgi:service ref="findProvidersDataServiceV1" interface="com.uhg.upm.dataservice.facetsunison.findproviders.FindProvidersDataServiceV1"/>

</beans>
```

```
Found 2 TODO items in 2 files
  dataservice-facetsunison-ProviderService-v1 (1 item in 1 file)
    pom.xml
      (55, 40) <!--TODO Add your wsdl here and delete this TODO-->
  META-INF.spring (1 item in 1 file)
    dataservice-facetsunison-ProviderService.xml
      (50, 9) <!--TODO Add your WSDL to the wsdl property and delete this comment-->
```

TODO Add your WSDL to the wsdl property and delete this comment

24. Add the Unversioned files to Subversion.
    - If svn brings up any ./out* or ./target* directories DO NOT commit them. Please add them to .svnignore (TortoiseSVN->"Unversion and add to ignore list").
      These directories are used by Maven and Felix as working directories and have the compiled classes and bundle jar files. These directories are not needed and will just add a lot of overhead to commits and updates as they are constantly changing.
    - In the directory ./.idea/ add the file "workspace.xml to .svnignore.
    - In the directory ./.idea/ add the "./libraries" directory to .svnignore.

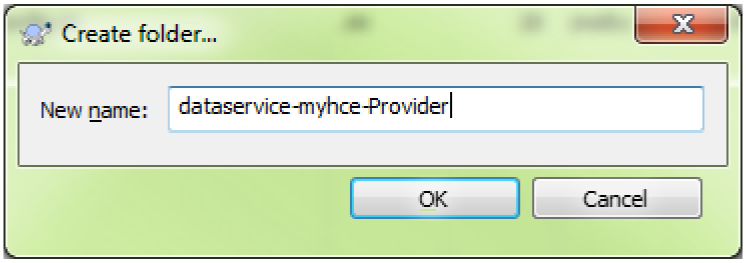
25. Once you have committed your new project it will need to be added to the build farm setup. If you have access, add your project, otherwise send an email to your dev lead or dev manager with the details for your new module, make sure to make note of any dependencies your module has so the build farm can be configured correctly.

REST Data Service Project Creation

1. **If this is your first time using maven archetypes and you haven't already set them up for use in Intellij, please follow the instructions for Initial Setup.**

2. Open TortoiseSVN repo browser (right-click anywhere in Windows Explorer and select TortoiseSVN/Repo-browser) to: http://svn.uhc.com/svn/uhgit_upm3

3. Create the new project at the root level of the repository by selecting "Create Folder…" (do not select "Add Folder…"). Name the project without the version (leave off –v1, etc). Naming convention should be dataservice-[backend]-[WADLName]

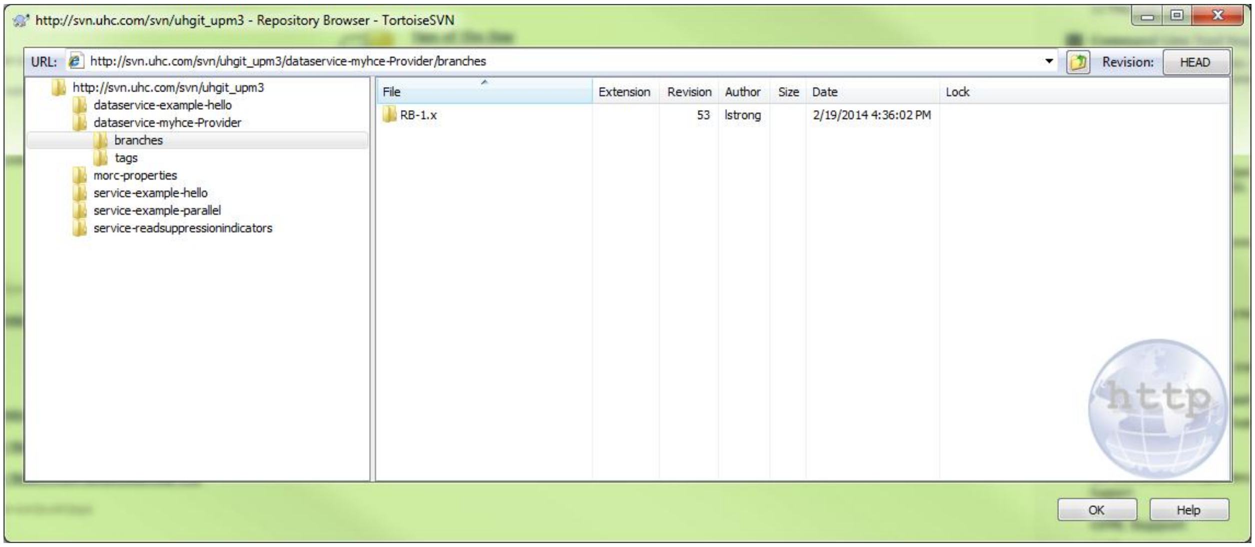


Create folder...

New name: dataservice-myhce-Provider

4. Under that directory, Create 2 folders: "branches" and "tags".

5. Under "branches", create the release branch for the version(s): RB-1.x, RB-2.x, etc. (note the "x" is intentional—don't replace it) Version should correspond to the backend WADL version number.



http://svn.uhc.com/svn/uhgit_upm3 - Repository Browser - TortoiseSVN

URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-myhce-Provider/branches

Revision: HEAD

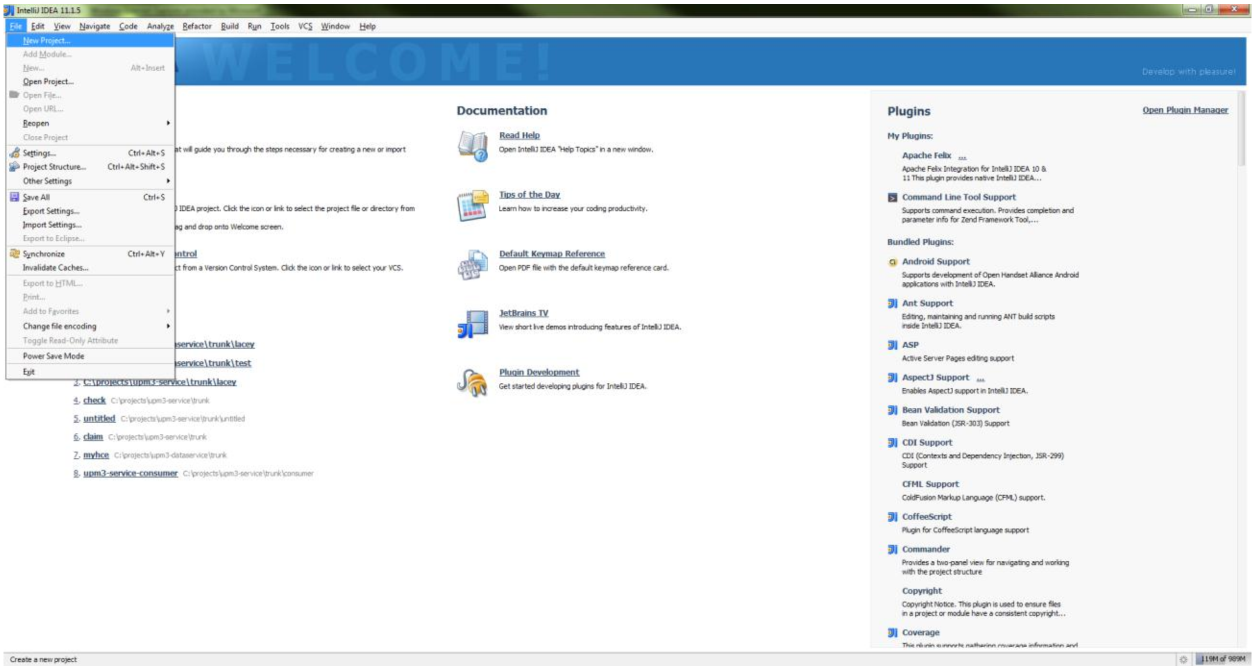
- http://svn.uhc.com/svn/uhgit_upm3
  - dataservice-example-hello
  - dataservice-myhce-Provider
    - branches
    - tags
  - morc-properties
  - service-example-hello
  - service-example-parallel
  - service-readsuppressionindicators

File	Extension	Revision	Author	Size	Date	Lock
RB-1.x		53	lstrong		2/19/2014 4:36:02 PM	

6. On local filesystem, create the project and branch directories: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

7. Checkout from SVN to that directory
   Repo URL: http://svn.uhc.com/svn/uhgit_upm3/dataservice-myhce-Provider/branches/RB-1.x/
   Local directory: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

8. Open Intellij and select File→New Project…



at will guide you through the steps necessary for creating a new or import

IDEA project. Click the icon or link to select the project file or directory from

ag and drop onto Welcome screen.

ntrol

ct from a Version Control System. Click the icon or link to select your VCS.

service\trunk\lacey

service\trunk\test

3. C:\projects\upm3-service\trunk\lacey

4. check C:\projects\upm3-service\trunk

5. untitled C:\projects\upm3-service\trunk\untitled

6. claim C:\projects\upm3-service\trunk

7. myhce C:\projects\upm3-dataservice\trunk

8. upm3-service-consumer C:\projects\upm3-service\trunk\consumer

Documentation

Read Help
Open IntelliJ IDEA "Help Topics" in a new window.

Tips of the Day
Learn how to increase your coding productivity.

Default Keymap Reference
Open PDF file with the default keymap reference card.

JetBrains TV
View short live demos introducing features of IntelliJ IDEA.

Plugin Development
Get started developing plugins for IntelliJ IDEA.

Plugins

My Plugins:

Apache Felix ...
Apache Felix Integration for IntelliJ IDEA 10 & 11 This plugin provides native IntelliJ IDEA...

Command Line Tool Support
Supports command execution. Provides completion and parameter info for Zend Framework Tool,...

Bundled Plugins:

Android Support
Supports development of Open Handset Alliance Android applications with IntelliJ IDEA.

Ant Support
Editing, maintaining and running ANT build scripts inside IntelliJ IDEA.

ASP
Active Server Pages editing support

AspectJ Support ...
Enables AspectJ support in IntelliJ IDEA.

Bean Validation Support
Bean Validation (JSR-303) Support

CDI Support
CDI (Contexts and Dependency Injection, JSR-299) Support

CFML Support
ColdFusion Markup Language (CFML) support.

CoffeeScript
Plugin for CoffeeScript language support

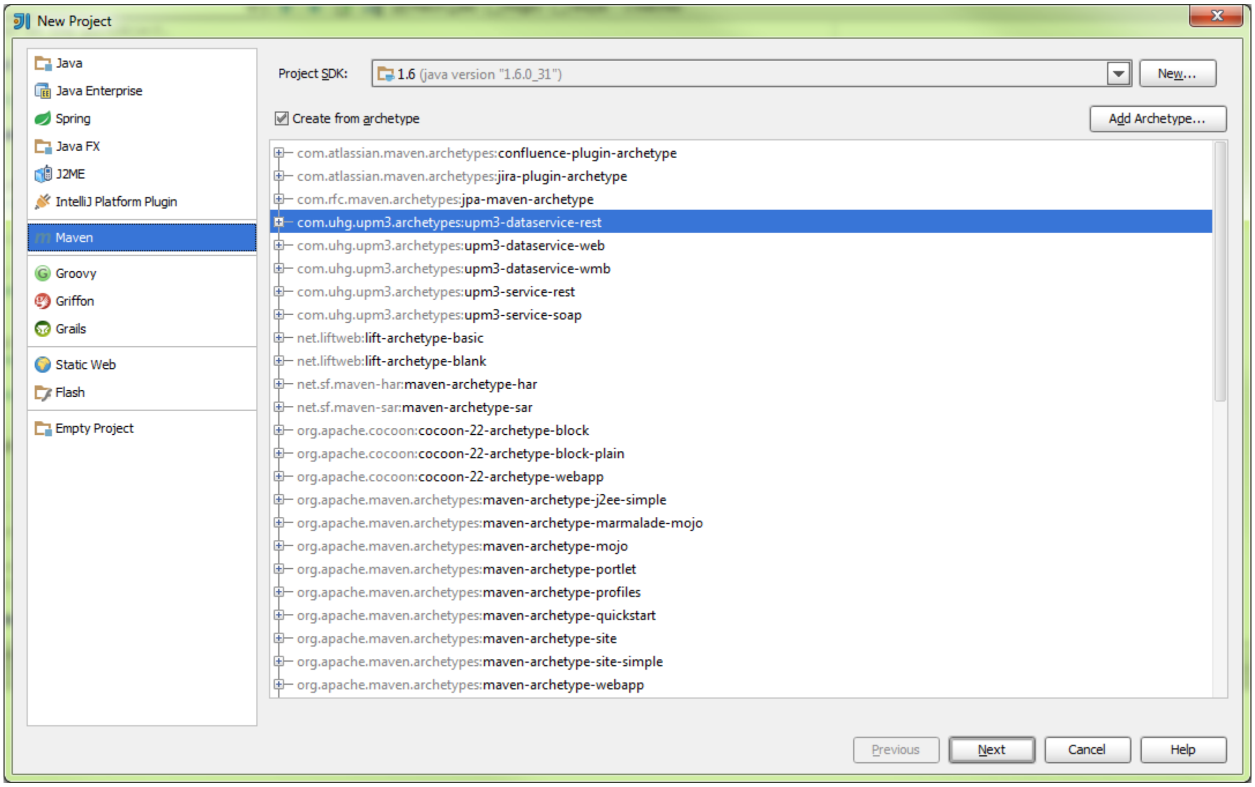
Commander
Provides a two-panel view for navigating and working with the project structure

Copyright
Copyright Notice. This plugin is used to ensure files in a project or module have a consistent copyright...

Coverage
This plugin supports gathering coverage information and

Create a new project

9. Select "Maven" and check the box for "Create from archetype". Select the com.uhg.upm3.archetypes:upm3-service-soap archetype and click "Next". For the Project SDK: make sure that "1.6" is selected.



Project SDK: 1.6 (java version "1.6.0_31")

☑ Create from archetype

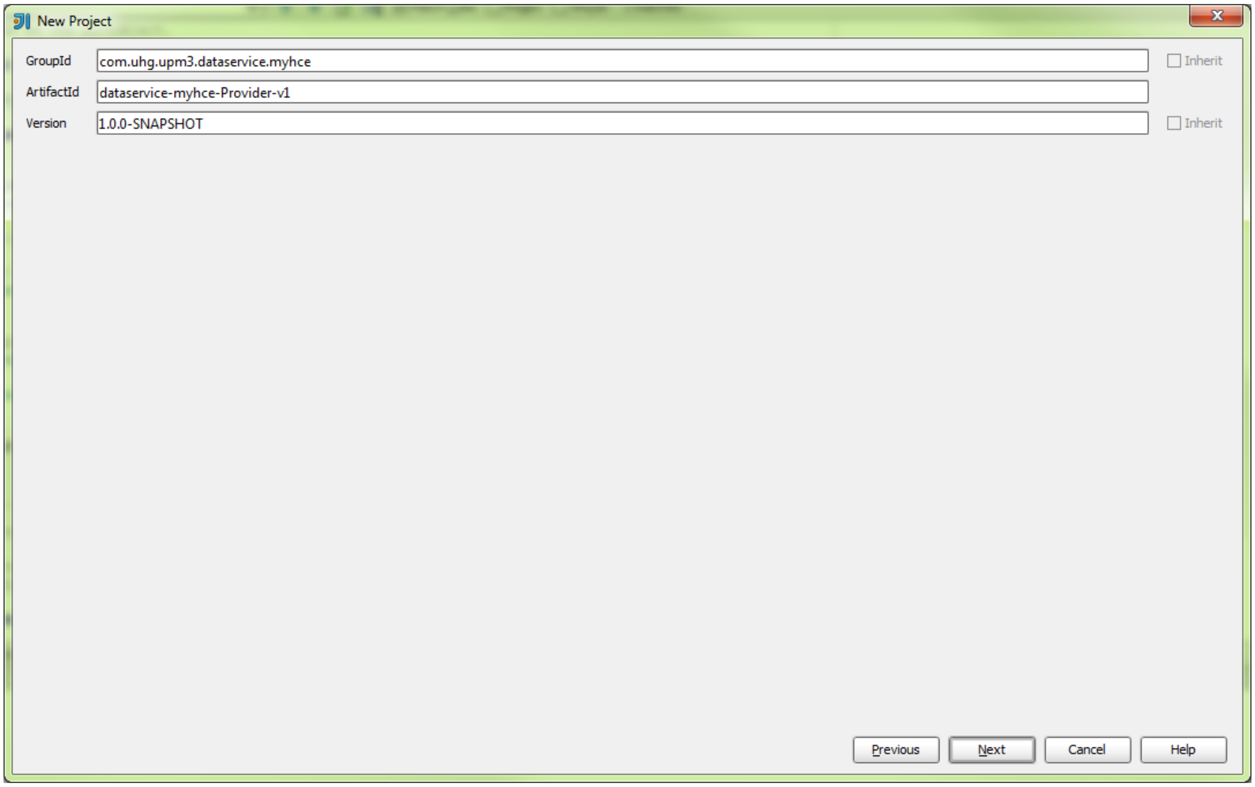
- com.atlassian.maven.archetypes:confluence-plugin-archetype
- com.atlassian.maven.archetypes:jira-plugin-archetype
- com.rfc.maven.archetypes:jpa-maven-archetype
- com.uhg.upm3.archetypes:upm3-dataservice-rest
- com.uhg.upm3.archetypes:upm3-dataservice-web
- com.uhg.upm3.archetypes:upm3-dataservice-wmb
- com.uhg.upm3.archetypes:upm3-service-rest
- com.uhg.upm3.archetypes:upm3-service-soap
- net.liftweb:lift-archetype-basic
- net.liftweb:lift-archetype-blank
- net.sf.maven-har:maven-archetype-har
- net.sf.maven-sar:maven-archetype-sar
- org.apache.cocoon:cocoon-22-archetype-block
- org.apache.cocoon:cocoon-22-archetype-block-plain
- org.apache.cocoon:cocoon-22-archetype-webapp
- org.apache.maven.archetypes:maven-archetype-j2ee-simple
- org.apache.maven.archetypes:maven-archetype-marmalade-mojo
- org.apache.maven.archetypes:maven-archetype-mojo
- org.apache.maven.archetypes:maven-archetype-portlet
- org.apache.maven.archetypes:maven-archetype-profiles
- org.apache.maven.archetypes:maven-archetype-quickstart
- org.apache.maven.archetypes:maven-archetype-site
- org.apache.maven.archetypes:maven-archetype-site-simple
- org.apache.maven.archetypes:maven-archetype-webapp

10. Click next.

11. Set the maven artifact values:
    a. GroupId should be com.uhg.upm3.dataservice.[backend] (eg. com.uhg.upm3.dataservice.myhce)

    b. ArtifactId should be dataservice-[backend]-[WADL Name]-v[WADL Version] (eg. dataservice-myhce-Provider-v1)

c. Version should correspond to the WADL version number of the service with SNAPSHOT (eg. 1.0.0-SNAPSHOT)



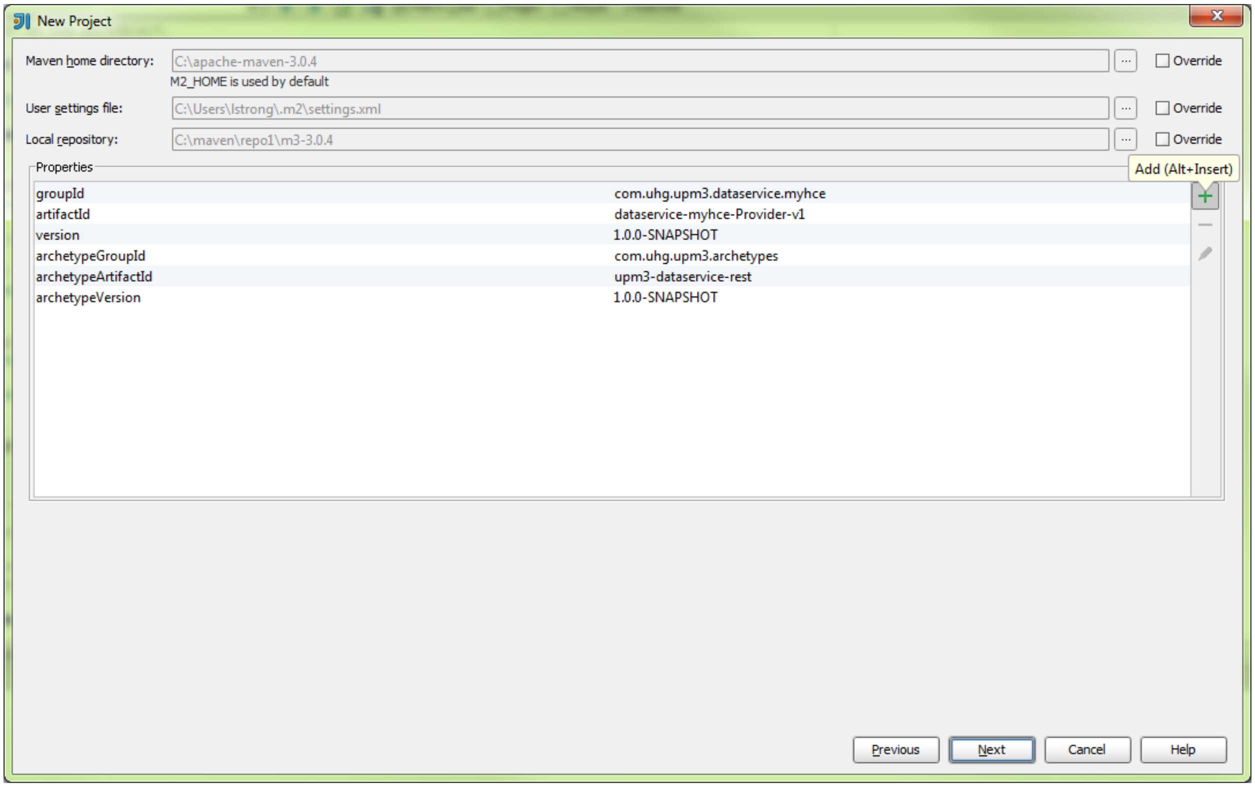
GroupId: com.uhg.upm3.dataservice.myhce □ Inherit

ArtifactId: dataservice-myhce-Provider-v1

Version: 1.0.0-SNAPSHOT □ Inherit

12. Click next.

13. Click on the + button in the properties box. This will bring up the Edit Maven Property dialog box.



Maven home directory: C:\apache-maven-3.0.4 □ Override
M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml □ Override

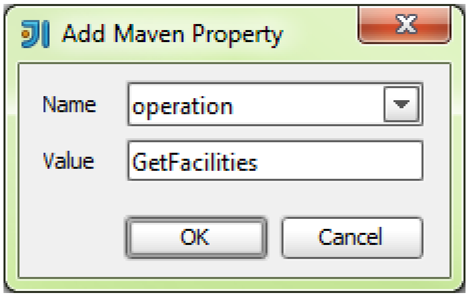
Local repository: C:\maven\repo1\m3-3.0.4 □ Override

Properties

groupId	com.uhg.upm3.dataservice.myhce
artifactId	dataservice-myhce-Provider-v1
version	1.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-dataservice-rest
archetypeVersion	1.0.0-SNAPSHOT

14. In the Name box type ‘operation’

15. In the Value box type the name of the WADL operation.

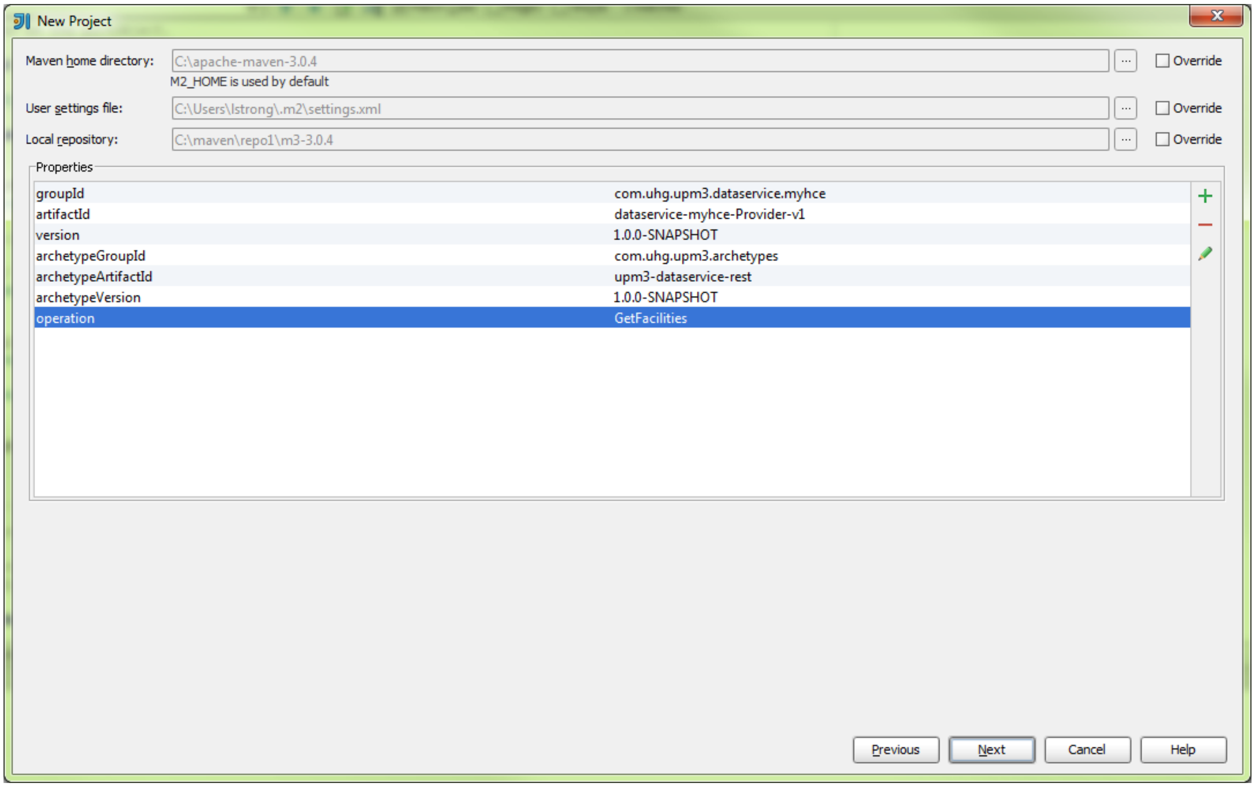


Name: operation

Value: GetFacilities

16. Click OK

17. You will see a summary of the properties being used to create the project/module.



Maven home directory: C:\apache-maven-3.0.4 □ Override

M2_HOME is used by default

User settings file: C:\Users\lstrong\.m2\settings.xml □ Override

Local repository: C:\maven\repo1\m3-3.0.4 □ Override

Properties

groupId	com.uhg.upm3.dataservice.myhce
artifactId	dataservice-myhce-Provider-v1
version	1.0.0-SNAPSHOT
archetypeGroupId	com.uhg.upm3.archetypes
archetypeArtifactId	upm3-dataservice-rest
archetypeVersion	1.0.0-SNAPSHOT
operation	GetFacilities

18. Click “Next” again and enter the following values:

    a. For Project name: Enter the name of the project that you just created on your local filesystem: dataservice-myhce-Provider

    b. For Project location: Click on the … and navigate to the folder that you just created.

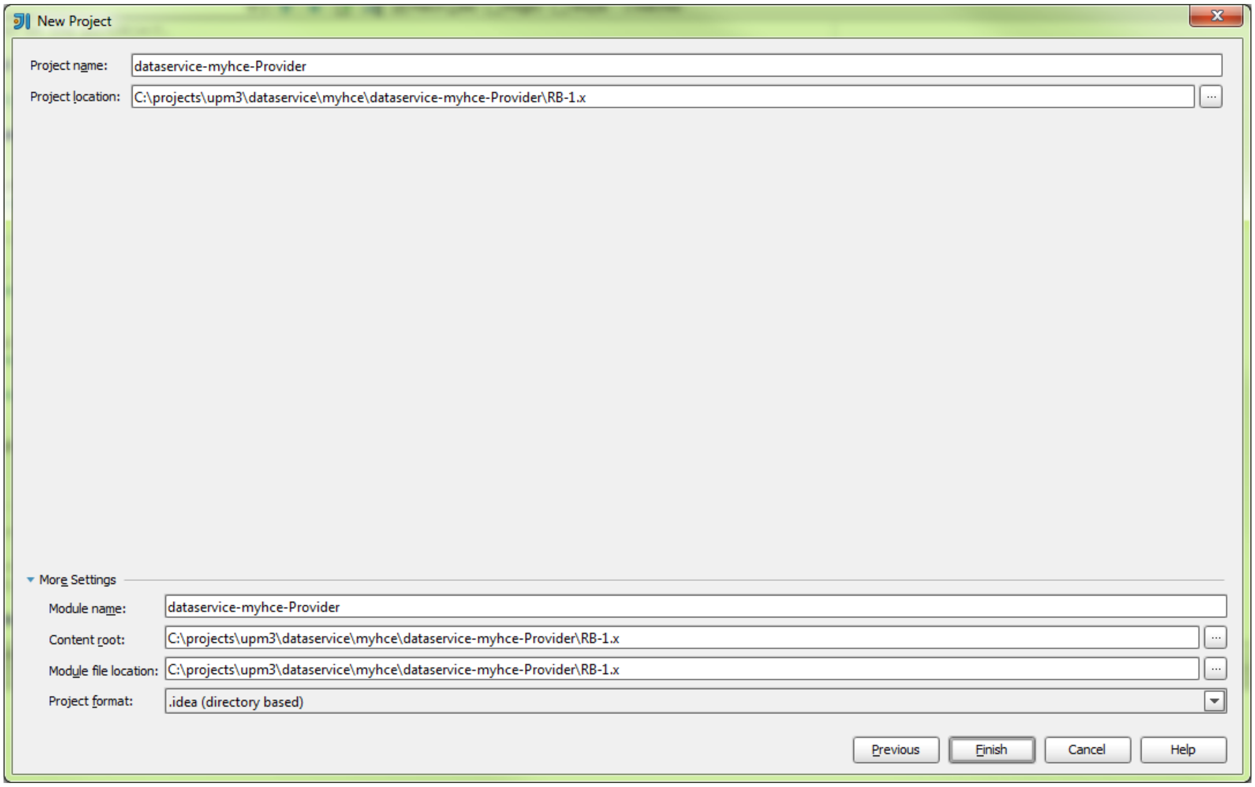
19. Expand the “More Settings” section at the bottom of the window and fill in the following values:

    a. Module name: Copy/Paste the Project name: dataservice-myhce-Provider

    b. Content Root: Should match what is in Project files location. If not, then Copy/Paste it here: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

    c. Module file location: Should match what is in Project files location. If not, then Copy/Paste it here: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

    d. Project file format: .idea (directory based) should be selected.



New Project

Project name: dataservice-myhce-Provider

Project location: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

More Settings

Module name: dataservice-myhce-Provider

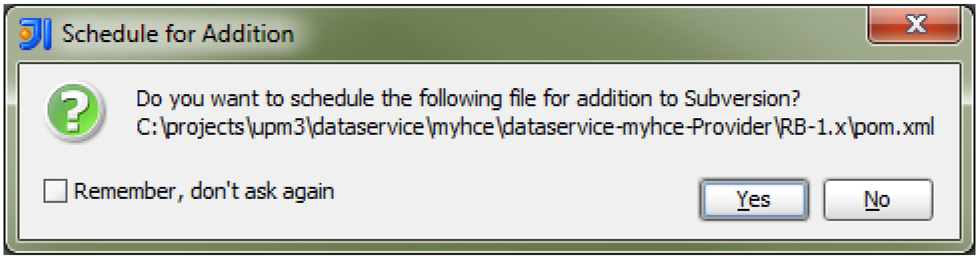
Content root: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

Module file location: C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x

Project format: .idea (directory based)

20. Click Finish.

21. Intellij will ask if you would like to add the pom.xml to Subversion. Click Yes.



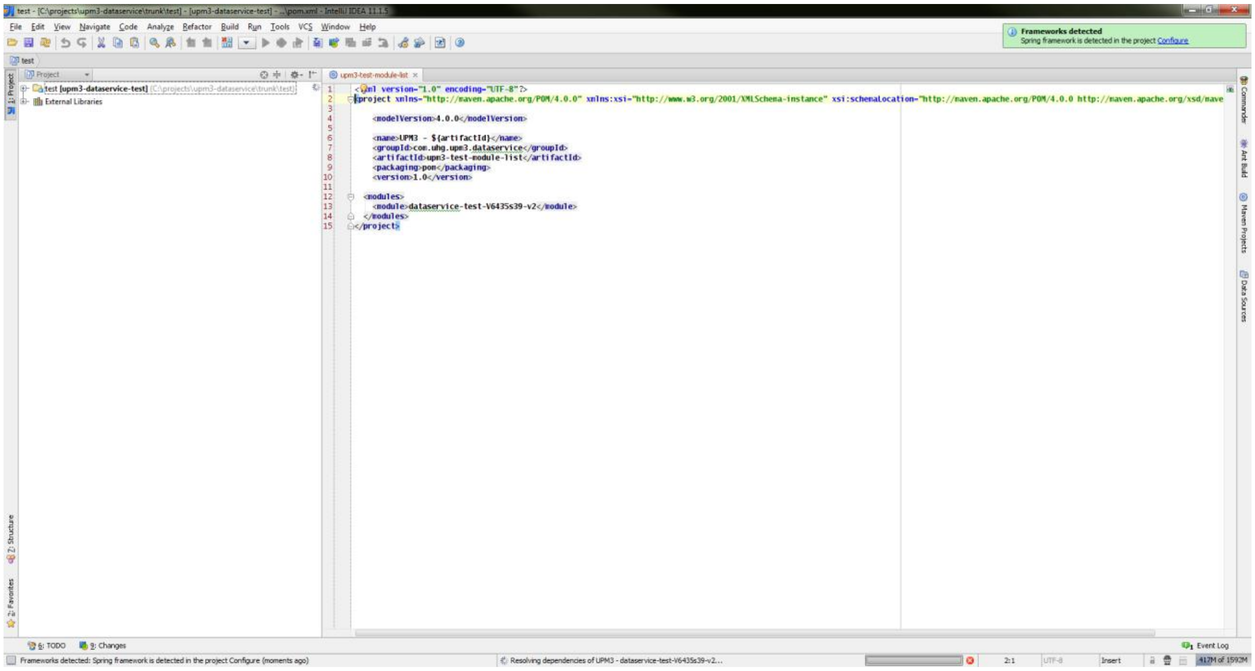
Do you want to schedule the following file for addition to Subversion?
C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x\pom.xml

☐ Remember, don't ask again

22. Intellij will then create the project from the archetype. If everything was successful you will see a message in the Messages window in Intellij.

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.919s
[INFO] Finished at: Wed Feb 19 12:42:11 EST 2014
[INFO] Final Memory: 6M/15M
[INFO] ------------------------------------------------------------------------
[INFO] Maven execution finished
```

23. You will briefly get a popup in the top right corner of your Intellij window that shows a Spring Framework has been detected. Click configure and accept the defaults.



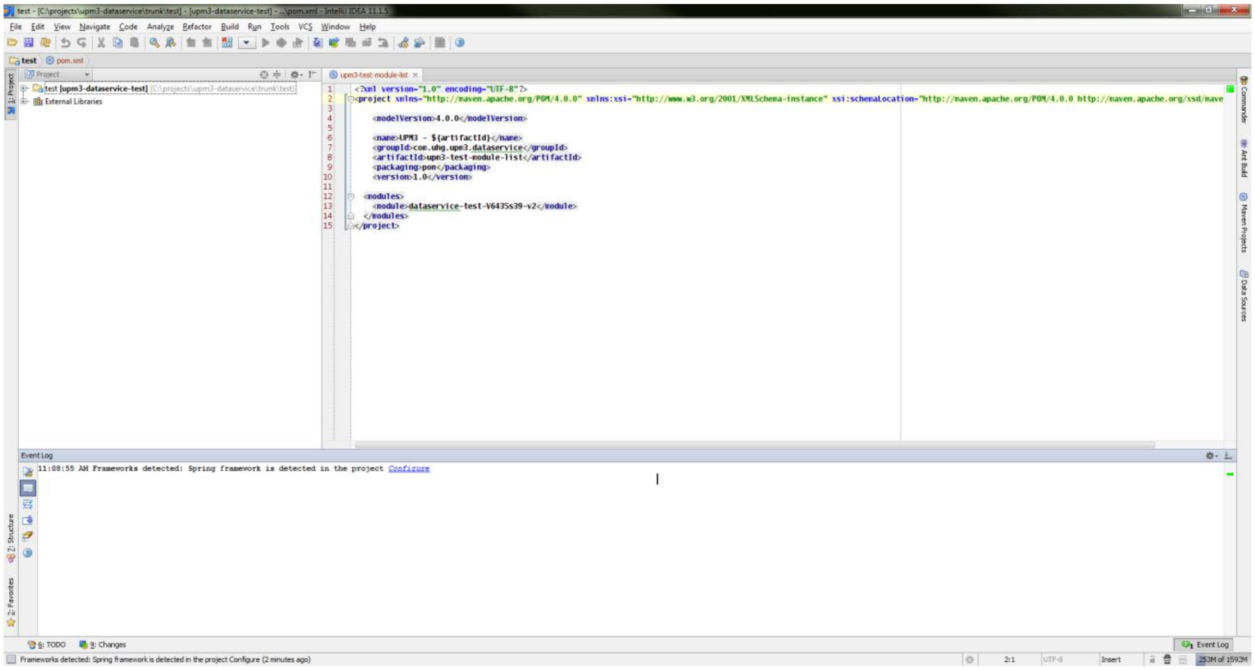
Frameworks detected
Spring framework is detected in the project Configure

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

Frameworks detected: Spring framework is detected in the project Configure (moments ago)

Resolving dependencies of UPM3 - dataservice-test-V6435s39-v2...

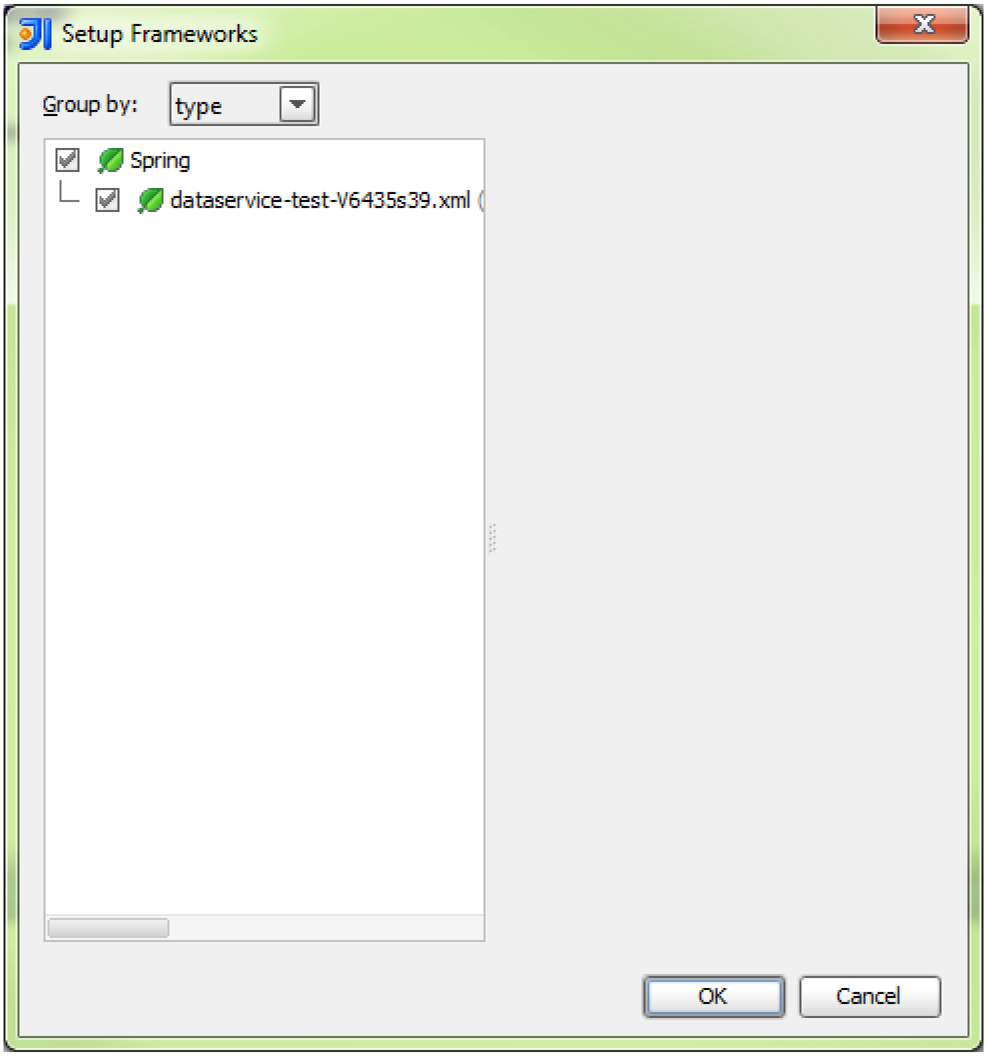
If the popup disappears before you are able to click it then open the Event Log in the bottom right corner of Intellij and click on Configure. It will bring up a Setup Frameworks dialog box. Click OK.


test - [C:\projects\upm3-dataservice\trunk\test] - [upm3-dataservice-test] - ...\pom.xml - IntelliJ IDEA 11.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
test
pom.xml
Project
test [upm3-dataservice-test] (C:\projects\upm3-dataservice\trunk\test)
External Libraries
upm3-test-module-list
Commander
Ant Build
Maven Projects
Data Sources
1: Project
7: Structure
2: Favorites
Event Log
6: TODO
9: Changes
Event Log
2:1
UTF-8
Insert
253M of 1593M

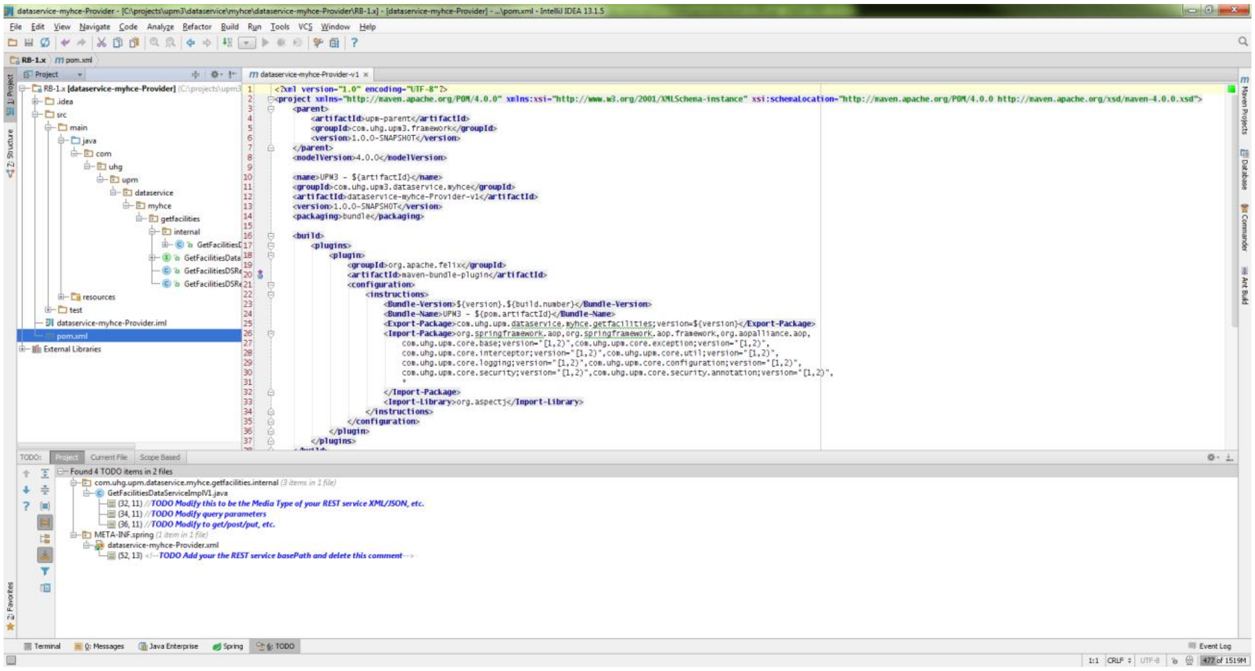
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave

    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice</groupId>
    <artifactId>upm3-test-module-list</artifactId>
    <packaging>pom</packaging>
    <version>1.0</version>

  <modules>
    <module>dataservice-test-V6435s39-v2</module>
  </modules>
</project>
```

```
11:08:55 AM Frameworks detected: Spring framework is detected in the project Configure
```

```
Frameworks detected: Spring framework is detected in the project Configure (2 minutes ago)
```



24. You will find TODO statements in the Impl class, where you will need to update the path of your REST service, Media Type, etc. You can also find these TODOs by clicking on the TODO tab in the bottom left corner of Intellij.


dataservice-myhce-Provider - [C:\projects\upm3\dataservice\myhce\dataservice-myhce-Provider\RB-1.x] - [dataservice-myhce-Provider] - ...\pom.xml - IntelliJ IDEA 13.1.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
RB-1.x
pom.xml
Project
RB-1.x [dataservice-myhce-Provider] (C:\projects\upm3
.idea
src
main
java
com
uhg
upm
dataservice
myhce
getfacilities
internal
GetFacilitiesD
GetFacilitiesData
GetFacilitiesDSRe
GetFacilitiesDSRe
resources
test
dataservice-myhce-Provider.iml
pom.xml
External Libraries
dataservice-myhce-Provider-v1
Maven Projects
Database
Commander
Ant Build
1: Project
Z: Structure
2: Favorites
Terminal
0: Messages
Java Enterprise
Spring
6: TODO
Event Log
1:1
CRLF
UTF-8
477 of 1519M

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <parent>
        <artifactId>upm-parent</artifactId>
        <groupId>com.uhg.upm3.framework</groupId>
        <version>1.0.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <name>UPM3 - ${artifactId}</name>
    <groupId>com.uhg.upm3.dataservice.myhce</groupId>
    <artifactId>dataservice-myhce-Provider-v1</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>bundle</packaging>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.felix</groupId>
                <artifactId>maven-bundle-plugin</artifactId>
                <configuration>
                    <instructions>
                        <Bundle-Version>${version}.${build.number}</Bundle-Version>
                        <Bundle-Name>UPM3 - ${pom.artifactId}</Bundle-Name>
                        <Export-Package>com.uhg.upm.dataservice.myhce.getfacilities;version=${version}</Export-Package>
                        <Import-Package>org.springframework.aop,org.springframework.aop.framework,org.aopalliance.aop,
                            com.uhg.upm.core.base;version="[1,2)",com.uhg.upm.core.exception;version="[1,2)",
                            com.uhg.upm.core.interceptor;version="[1,2)",com.uhg.upm.core.util;version="[1,2)",
                            com.uhg.upm.core.logging;version="[1,2)",com.uhg.upm.core.configuration;version="[1,2)",
                            com.uhg.upm.core.security;version="[1,2)",com.uhg.upm.core.security.annotation;version="[1,2)",
                            *
                        </Import-Package>
                        <Import-Library>org.aspectj</Import-Library>
                    </instructions>
                </configuration>
            </plugin>
        </plugins>
```

```
TODO:  Project  Current File  Scope Based
Found 4 TODO items in 2 files
  com.uhg.upm.dataservice.myhce.getfacilities.internal (3 items in 1 file)
    GetFacilitiesDataServiceImplV1.java
      (32, 11) //TODO Modify this to be the Media Type of your REST service XML/JSON, etc.
      (34, 11) //TODO Modify query parameters
      (36, 11) //TODO Modify to get/post/put, etc.
  META-INF.spring (1 item in 1 file)
    dataservice-myhce-Provider.xml
      (52, 13) <!--TODO Add your the REST service basePath and delete this comment-->
```

25. Add the Unversioned files to Subversion.
    - If svn brings up any ./out* or ./target* directories DO NOT commit them. Please add them to .svnignore (TortoiseSVN->"Unversion and add to ignore list").

      These directories are used by Maven and Felix as working directories and have the compiled classes and bundle jar files. These directories are not needed and will just add a lot of overhead to commits and updates as they are constantly changing.
    - In the directory ./.idea/ add the file "workspace.xml to .svnignore.
    - In the directory ./.idea/ add the "./libraries" directory to .svnignore.

26. Once you have committed your new project it will need to be added to the build farm setup. If you have access, add your project, otherwise send an email to your dev lead or dev manager with the details for your new module, make sure to make note of any dependencies your module has so the build farm can be configured correctly.

Intial Setup

1. To add the latest UPM3 Maven Archetypes to Intellij, please copy the UserArchetypes.xml file located here to this directory and restart Intellij.

   C:\Users\[msid]\.IntelliJIdea13\system\Maven\Indices

If you have added the file to the proper directory, and you still are unable to see the archetypes in Intellij, then please make sure that your settings.xml file in your maven/conf directory have the correct login information in them. As a reminder, this file must be updated every time you change your Windows password.

Log in to Artifactory to get your encrypted password:

i. Go to http://repo1.uhc.com/artifactory/webapp/login.html
ii. Log in using your Windows login
iii. Click on your username in the upper right corner
iv. Type in your Current Password and click Unlock
v. This will give you what you should put in the "server" section of the maven settings.xml file. Replace username/password in your local settings.xml file (NOTE—only change the username/password tags, leave the id as "artifactory" even though it gives you something else to paste that has ${server-id}—this will not work):

```
<server>
    <id>artifactory</id>
    <username>[userId]</username>
    <password>[encrypted.pwd.from.artifactory]</password>
</server>
```