SOA Installation
Java & Maven & IntelliJ & Wildfly

Jacek Oleś[1]

[1]AGH University of Science and Technology

Faculty of Electrical Engineering,
Automatics, Computer Science
and Biomedical Engineering

ul. Mickiewicza 30
30-059 Kraków
Poland

11.03.2019

Part I

Installation

Table of Contents

Installing Java

Installing Java 8 (recommended)

- sudo add-apt-repository ppa:webupd8team/java
- sudo apt update
- sudo apt install oracle-java8-installer
- sudo apt install oracle-java8-set-default
- sudo add-apt-repository ppa:openjdk-r/ppa
- sudo apt-get update
- sudo apt-get install openjdk-8-jdk
- sudo update-alternatives –config java

Selection	Path	Priority	Status
0	.../java-11-openjdk-amd64/...	1101	auto mode
1	.../java-11-openjdk-amd64/...	1101	manual mode
* 2	.../java-8-openjdk-amd64/jre/...	1081	manual mode
3	.../java-8-oracle/jre/...	1081	manual mode

- This is Java's real location

- java -version & javac -version
- whereis java

```
java: /usr/bin/java /usr/share/java
/usr/lib/jvm/java-8-oracle/bin/java
/usr/lib/jvm/java-8-oracle/jre/bin/java
/usr/share/man/man1/java.1.gz
```

- ls -l /usr/bin/java

```
lrwxrwxrwx 1 root root 22 lut 22 15:18 /usr/bin/java ->
/etc/alternatives/java
```

- ls -l /etc/alternatives/java

```
lrwxrwxrwx 1 root root 46 lut 25 09:59 /etc/alternatives/java ->
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

- ls -l /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

```
-rwxr-xr-x 1 root root 6280 sty 14 22:02
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Installing Java

Installing Java 11 (not recommended)

- sudo add-apt-repository ppa:webupd8team/java
- sudo apt-get update
- sudo apt install default-jdk
- Item sudo apt install default-jre
- Item javac -version
- Item java -version

Installing Maven

- sudo apt-get install maven
- mvn -version

```
Apache Maven 3.5.2
Maven home: /usr/share/maven
Java version: 1.8.0_201, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-oracle/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.18.0-15-generic", arch: "amd64", family: "unix"
```

- which mvn

```
/usr/bin/mvn
```

- ls -l /usr/bin/mvn

```
lrwxrwxrwx 1 root root 21 lut 22 16:32 /usr/bin/mvn -> /etc/alternatives/mvn
```

- ls -l /etc/alternatives/mvn

```
lrwxrwxrwx 1 root root 24 lut 22 16:32 /etc/alternatives/mvn -> /usr/share/maven/bin/mvn
```

- ls -l /usr/share/maven/bin/mvn

```
-rwxr-xr-x 1 root root 5969 paź 18 2017 /usr/share/maven/bin/mvn
```

- This is Maven's real location

Installing IntelliJ

- Get student licence at:

https://www.jetbrains.com/shop/eform/students
Using name@student.agh.edu.pl mail

- Download at:

https://www.jetbrains.com/idea/download/

- Extract -> Run ./idea.sh:
- To create shortcut
  Open a terminal and go to

/extracted_to_path/idea/bin

- On my computer it is in

~/home/IdeaUI-install/bin

- Start IntelliJ with ./idea.sh.
- Tools -> Generate Desktop Entry.
- Close IntelliJ.
- In the terminal, start nautilus as admin

sudo nautilus

- Go to /usr/share/applications.
- Drag IntelliJ's icon to your desktop and/or launcher
- If you've installed by mistake
  Vim emulator and don't want it
- Launch IntelliJ and
  Go to:

File -> Settings\Preferences (Ctrl+Alt+S) -> Plugins -> Installed -> IdeaVim *uncheck*

- If you were using other means of installation here are Default IntelliJ's locations:

Windows: %LOCALAPPDATA%\JetBrains\Toolbox\apps
macOS: ~/Library/Application Support/JetBrains/Toolbox/apps
Linux: ~/.local/share/JetBrains/Toolbox/apps

Installing Wildfly

- Download Wildfly at:

`http://wildfly.org/downloads/`

- I recommend version 14.0.1.Final (2018-09-05)

Works with latest Red Hat (JBoss) Developer Studio
and I've tested it with previously mentioned Java version

- Open terminal and go to

```
cd /extracted_to_path/
```

- Change file permission

```
chmod -R 777 /wildfly-14.0.1.Final
```

- Add new users (management or application):

```
.../wildfly-14.0.1.Final/bin/add-user.sh
```

- Be sure to remember users' passwords
those are kept in encoded form

Login	Passwd	Group	XBool
admin	admin	admin	yes
user	user	user	yes

XBool - is this new user going to be used for one AS process to connect to another AS process

- Users' info is kept at:

```
.../wildfly-14.0.1.Final/standalone/configuration/application-roles.properties
.../wildfly-14.0.1.Final/standalone/configuration/application-users.properties
```

- Run Wildfly using:

```
.../wildfly-14.0.1.Final/bin/standalone.sh .../wildfly-14.0.1.Final/bin/standalone.sh -c standalone-full.xml
.../wildfly-14.0.1.Final/bin/standalone.sh -c standalone-full-ha.xml
```

- Killing Wildfly
By pressing Ctrl+C in terminal running Wildfly

- Or by running one those of commends in other terminal:

```
pgrep -d" " -f "wildfly" — xargs kill;
pgrep -d" " -f "jboss" — xargs kill;
.../wildfly-14.0.1.Final/bin/jboss-cli.sh –connect controller=127.0.0.1:9990 command=:shutdown
.../wildfly-14.0.1.Final/bin/jboss-cli.sh –connect command=:shutdown
```

- Add environment variable JBOSS_HOME to PATH

```
gedit ~/.profile
```

- Add line:

```
export JBOSS_HOME="$HOME"/wildfly-14.0.1.Final
```

- Save and close
- Refresh terminal sources

```
source ~/.profile
echo $JBOSS_HOME
```

- You should see variable $JBOSS_HOME like so

```
/home/jackdaeel/wildfly-14.0.1.Final
```

- For Windows OS use .bat files instead .sh to run Wildfly scripts
- Basic info:

Default settings:

Management console: `http://127.0.0.1:9990/`
Application access endpoint: `http://127.0.0.1:8080/`

You can find additional info and change configuration in:

```
.../wildfly-14.0.1.Final/standalone/configuration/standalone(...).xml
```

Try searching for:

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
```

Installing SOAP UI

- Download SOAP UI at:

urlhttps://www.soapui.org/downloads/soapui.html
(Open Source, there's no student licence for Pro version)

- ls ~/Downloads

SoapUI-x64-5.5.0.sh

- chmod 777 ~/Downloads/SoapUI-x64-5.5.0.sh
- ~/Downloads/SoapUI-x64-5.5.0.sh
- The rest is straight forward installation

Next -> Next -> Finish

Part II

Working with Wildfly

Table of Contents

Maven - New project - usual project generation

- To generate new EJB project open terminal
- Enter Idea's (IntelliJ's) projects' directory and create separate directory for Maven projects

```
cd IdeaProjects
mkdir MavenProjects
cd MavenProjects
```

- Now to generate project using Maven
- mvn archetype:generate

```
Choose a number or apply filter (format: [groupId:]artifactId,
case sensitive contains): 1327:
```

- wildfly

```
...
12: remote ->
org.wildfly.archetype:wildfly-javaee7-webapp-ear-blank-archetype
(An archetype that generates a starter Java EE 7 webapp project for
JBoss Wildfly. The project is an EAR, with an EJB-JAR and WAR)
...
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains):
```

- 12

```
Choose org.wildfly.archetype:wildfly-javaee7-webapp-ear-blank-archetype version:
1: 8.1.0.Final
2: 8.2.0.Final
Choose a number: 2:
```

- 2

```
Define value for property 'groupId': my.first.group.id
Define value for property 'artifactId': MyFirstProjectName
Define value for property 'version' 1.0-SNAPSHOT: : 1.0.0.Final
Define value for property 'package' my.first.group.id: : my.first.projects.first.package
Confirm properties configuration:
(...)
Y: :
```

- Confirm by typing in 'y'
- This is what you should see after correctly generating EJB project

```
(...)
Using following parameters for creating project from Archetype:
wildfly-javaee7-webapp-ear-blank-archetype:8.2.0.Final
(...)
[INFO] Parameter: groupId, Value: my.first.group.id
[INFO] Parameter: artifactId, Value: MyFirstProjectName
[INFO] Parameter: version, Value: 1.0.0.Final
[INFO] Parameter: package, Value: my.first.projects.first.package
[INFO] Parameter: packageInPathFormat, Value: my/first/projects/first/package
[INFO] Parameter: package, Value: my.first.projects.first.package
[INFO] Parameter: version, Value: 1.0.0.Final
[INFO] Parameter: groupId, Value: my.first.group.id
[INFO] Parameter: artifactId, Value: MyFirstProjectName
[INFO] Parent element not overwritten in
/home/jackdaeel/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-ejb/pom.xml
[INFO] Parent element not overwritten in
/home/jackdaeel/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-web/pom.xml
[INFO] Parent element not overwritten in
/home/jackdaeel/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-ear/pom.xml
[INFO] Project created from Archetype in dir: /home/jackdaeel/IdeaProjects/MavenProjects/MyFirstProjectName
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
(...)
```

- groupId - Project group that this project belongs to
  artifactId - Project name
  version - Project version (1.2.3 Snapshot / Release / Final)
  package - ???
- ls

```
MyFirstProjectName
wildfly
```

Maven - New project - 1st time generation

- On 1st generation of project using Maven you may encounter different generating process
- mvn archetype:generate

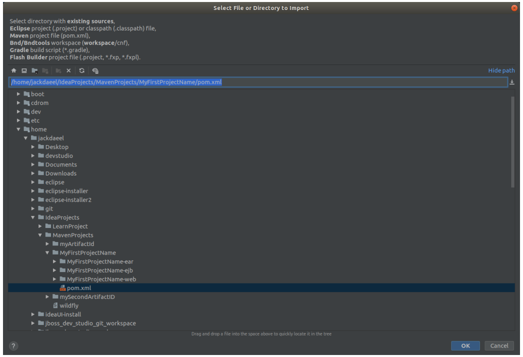
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:

- 8: 1.4
- Continue with generation (similar to the one in previous instruction) and finish it
- This project isn't what we want to create
- Try generating project again.
- Now it should generate the way it was described on the previous slide

Intellij - Importing Project

- After launching IntelliJ go to

Project -> New -> From existing Sources



Select File or Directory to Import

Select directory with **existing sources**,
Eclipse project (.project) or classpath (.classpath) file,
Maven project file (pom.xml),
Bnd/Bndtools workspace (**workspace**/cnf),
Gradle build script (*.gradle),
Flash Builder project file (.project, *.fxp, *.fxpl).

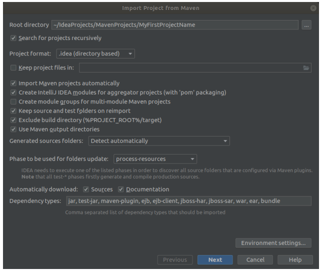
```
/home/jackdaeel/IdeaProjects/MavenProjects/MyFirstProjectName/pom.xml
boot
cdrom
dev
etc
home
  jackdaeel
    Desktop
    devstudio
    Documents
    Downloads
    eclipse
    eclipse-installer
    eclipse-installer2
    git
    IdeaProjects
      LearnProject
      MavenProjects
        myArtifactId
        MyFirstProjectName
          MyFirstProjectName-ear
          MyFirstProjectName-ejb
          MyFirstProjectName-web
          pom.xml
        mySecondArtifactID
        wildfly
    ideaUI-install
    jboss_dev_studio_git_workspace
```

Drag and drop a file into the space above to quickly locate it in the tree

- Click OK after selecting pom.xml inside your project directory
- Click Environment settings... -> Maven Environment
- Set 'Maven home directory:' real Maven location

Shown how to find Maven location in section **Installing Maven**

- Click OK -> Next



Import Project from Maven

Root directory: ~/IdeaProjects/MavenProjects/MyFirstProjectName
- [✓] Search for projects recursively

Project format: .idea (directory based)
- [ ] Keep project files in:
- [✓] Import Maven projects automatically
- [✓] Create IntelliJ IDEA modules for aggregator projects (with 'pom' packaging)
- [ ] Create module groups for multi-module Maven projects
- [✓] Keep source and test folders on reimport
- [✓] Exclude build directory (%PROJECT_ROOT%/target)
- [✓] Use Maven output directories

Generated sources folders: Detect automatically

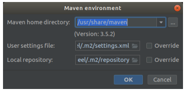
Phase to be used for folders update: process-resources

IDEA needs to execute one of the listed phases in order to discover all source folders that are configured via Maven plugins. **Note** that all test-* phases firstly generate and compile production sources.

Automatically download: [✓] Sources [✓] Documentation

Dependency types: jar, test-jar, maven-plugin, ejb, ejb-client, jboss-har, jboss-sar, war, ear, bundle

Comma separated list of dependency types that should be imported

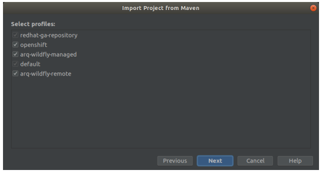


Maven environment

Maven home directory: /usr/share/maven
(Version: 3.5.2)
User settings file: el/.m2/settings.xml [ ] Override
Local repository: eel/.m2/repository [ ] Override

- OK -> Next -> Selected profiles: redhat-ga-repository *checked*

On next import there were more options to check so I checked them
Afterwards couldn't tell the difference

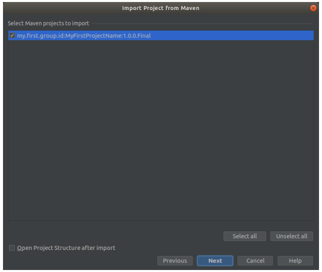


Import Project from Maven

Select profiles:
- [✓] redhat-ga-repository
- [✓] openshift
- [✓] arq-wildfly-managed
- [✓] default
- [✓] arq-wildfly-remote

Intellij - Importing Project

- Next -> Select Maven projects to import:
- my.first.group.id:MyFirstProjectName:1.0.0.Final *checked*


Import Project from Maven

Select Maven projects to import

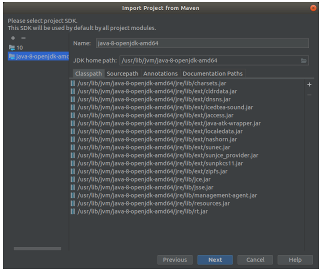
☑ my.first.group.id:MyFirstProjectName:1.0.0.Final

☐ Open Project Structure after import

- Next -> Please Select SDK. This SDK will be used by default by all project modules.
- Click "+" -> Add New SDK -> JDK -> Browse to real Java location
- OK -> Name it however you want -> Next

Shown how to find Maven location in section **Installing Java**

I recommend Java 8 (/usr/lib/jvm/java-8-openjdk-amd64)


Import Project from Maven

Please select project SDK.
This SDK will be used by default by all project modules.

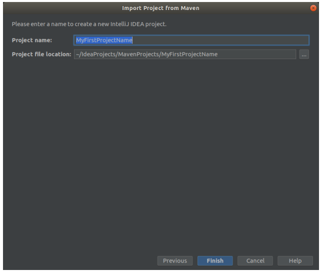
10
java-8-openjdk-amd

Name: java-8-openjdk-amd64

JDK home path: /usr/lib/jvm/java-8-openjdk-amd64

Classpath | Sourcepath | Annotations | Documentation Paths

```
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/charsets.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/cldrdata.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/dnsns.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/icedtea-sound.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/jaccess.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/java-atk-wrapper.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/localedata.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/nashorn.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/sunec.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/sunjce_provider.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/sunpkcs11.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext/zipfs.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/jce.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/jsse.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/management-agent.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/resources.jar
/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/rt.jar
```


Import Project from Maven

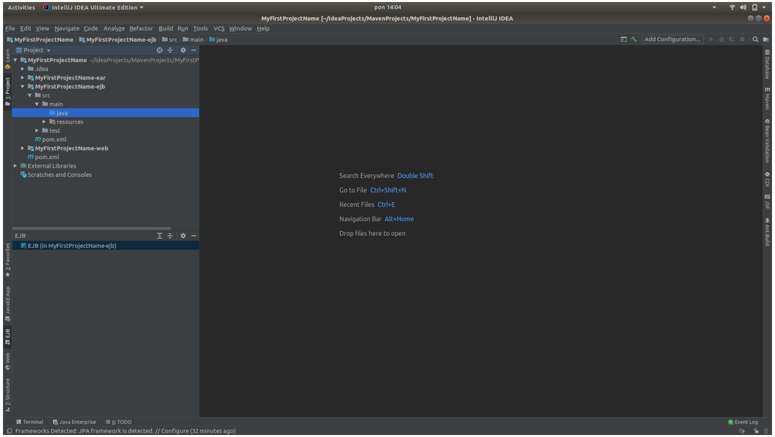
Please enter a name to create a new IntelliJ IDEA project.

Project name: MyFirstProjectName

Project file location: ~/IdeaProjects/MavenProjects/MyFirstProjectName

Intellij - Importing Project

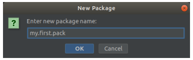
- Finish
- This is how your imported project should look like



Frameworks Detected: JPA framework is detected. // Configure (32 minutes ago)

Intellij - First SOAP application

- In project view (on left side of IDE) expand:
- Project_name -> Project_name-ejb -> src -> main -> java -> Right-click -> New -> Package -> New Package
- Enter new package name: my.first.pack -> OK


New Package

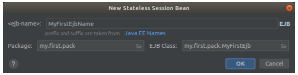
Enter new package name:

my.first.pack

OK Cancel

- Right-click on my.first.pack -> New
- Choose 'Stateless Session Bean' or 'Java Class'
- Fill out '<ejb-name>': MyFirstEjbName
- The rest should populate itself – like so:

package: my.first.pack
EJB Class: my.first.pack.MyFirstEjb


New Stateless Session Bean

<ejb-name>: MyFirstEjbName EJB

prefix and suffix are taken from Java EE Names

Package: my.first.pack EJB Class: my.first.pack.MyFirstEjb

OK Cancel

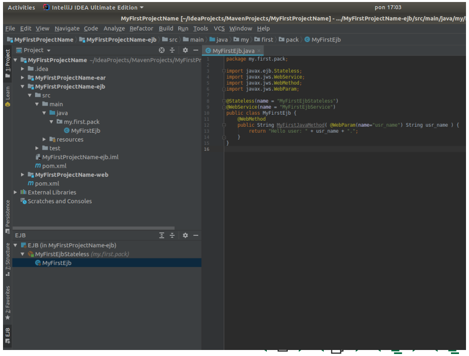
- Click OK
- You should see new generated class like so
- And if you created class through '... Session Bean' You should see 'MyFirstEjb' in EJB view of the project

Fresh class

```java
package my.first.pack;

import javax.ejb.Stateless;

@Stateless(name = "MyFirstEjbName")
public class MyFirstEjb {
    public MyFirstEjb() {
    }
}
```

Modified class

```java
package my.first.pack;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@Stateless(name = "MyFirstEjbStateless")
@WebService(name = "MyFirstEjbService")
public class MyFirstEjb {
    @WebMethod
    public String MyFirstJavaMethod( @WebParam(name="usr_name") String usr_name ) {
        return "Hello user: " + usr_name + ".";
    }
}
```


Activities IntelliJ IDEA Ultimate Edition pon 17:03

MyFirstProjectName [~/IdeaProjects/MavenProjects/MyFirstProjectName] - .../MyFirstProjectName-ejb/src/main/java/my/

MyFirstProjectName 〉 MyFirstProjectName-ejb 〉 src 〉 main 〉 java 〉 my 〉 first 〉 pack 〉 MyFirstEjb

```
MyFirstProjectName ~/IdeaProjects/MavenProjects/MyFirstPr
  .idea
  MyFirstProjectName-ear
  MyFirstProjectName-ejb
    src
      main
        java
          my.first.pack
            MyFirstEjb
        resources
      test
    MyFirstProjectName-ejb.iml
    pom.xml
  MyFirstProjectName-web
  pom.xml
External Libraries
Scratches and Consoles
```

EJB

```
EJB (in MyFirstProjectName-ejb)
  MyFirstEjbStateless (my.first.pack)
    MyFirstEjb
```

MyFirstEjb.java

```java
package my.first.pack;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@Stateless(name = "MyFirstEjbStateless")
@WebService(name = "MyFirstEjbService")
public class MyFirstEjb {
    @WebMethod
    public String MyFirstJavaMethod( @WebParam(name="usr_name") String usr_name ) {
        return "Hello user: " + usr_name + ".";
    }
}
```

Maven - Building & Deploying project

- To build project open console
- Change location to folder containing project
  cd ∼/IdeaProjects/MavenProjects/MyFirstProjectName
- mvn clean package
- Now you should see message:

```
[INFO] MyFirstProjectName ........................................ SUCCESS [... s]
[INFO] MyFirstProjectName: EJB Module ..................... SUCCESS [... s]
[INFO] MyFirstProjectName: WAR Module ................... SUCCESS [... s]
[INFO] MyFirstProjectName: EAR Module .................... SUCCESS [... s]
[INFO] - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[INFO] BUILD SUCCESS
```

- To deploy package to wildfly you need to
  have running wildfly server
  and set up JBOSS_HOME environment variable

It is explained how to do those 2 things in section **Installing Wildfly**

- If you have it prepared just type in the same folder
- mvn clean package wildfly:deploy
- In console from which you deployed package
  you should see:

```
[INFO] MyFirstProjectName ........................................ SUCCESS [... s]
[INFO] MyFirstProjectName: EJB Module ..................... SUCCESS [... s]
[INFO] MyFirstProjectName: WAR Module ................... SUCCESS [... s]
[INFO] MyFirstProjectName: EAR Module .................... SUCCESS [... s]
[INFO] - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[INFO] BUILD SUCCESS
```

- In console from which you are running Wildfly server
  you should see:

```
(...) INFO (...) Adding service endpoint metadata: id=MyFirstEjbStateless
address=http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService
implementor=my.first.pack.MyFirstEjb
serviceName=http://pack.first.my/MyFirstEjbService
portName=http://pack.first.my/MyFirstEjbServicePort
annotationWsdlLocation=null
wsdlLocationOverride=null
mtomEnabled=false
(...) INFO (...) Creating Service http://pack.first.my/MyFirstEjbService from class my.first.pack.MyFirstEjb
(...) INFO (...) Setting the server's publish address to be http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService
(...) INFO (...) WSDL published to:
file:/home/jackdaeel/wildfly-14.0.1.Final/standalone/data/wsdl/MyFirstProjectName-ear.ear/MyFirstProjectName-ejb.jar/MyFirstEjbService.wsdl
(...) INFO (...) Starting service jboss.ws.endpoint."MyFirstProjectName-ear.ear"."MyFirstProjectName-ejb.jar".MyFirstEjbStateless
(...) INFO (...) Starting Persistence Unit (phase 2 of 2) Service 'MyFirstProjectName-ear.ear/MyFirstProjectName-ejb.jar#primary'
(...)
(...) INFO (...) Registered web context: '/MyFirstProjectName-ejb' for server 'default-server'
(...) INFO (...) Initializing Mojarra 2.3.5.SP2 for context '/MyFirstProjectName-web'
```

- Now go to:

```
http://localhost:
8080/MyFirstProjectName-ejb/MyFirstEjbService?wsdl
```

Wildfly - Deploying using Management user

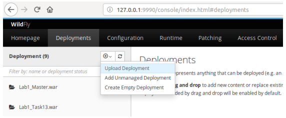
- To deploy package to wildfly you need to
  have running wildfly server
  and create management user

It is explained how to do those 2 things in section **Installing Wildfly**

- If you have done those 2 things go to:

`http://127.0.0.1:9990`

- Log in using management user credentials
- go to Deployments -> (+) -> Add/Upload deployment



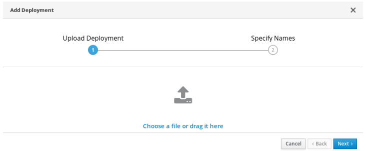
presents anything that can be deployed (e.g. an

g and drop to add new content or replace existi

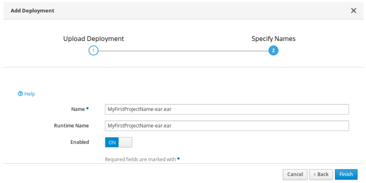
ded by drag and drop will be enabled by default.

- Select .war file

.war file - ~/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-ear/target/MyFirstProjectName-ear.war



- Next -> set 'Enabled' to ON -> Finish



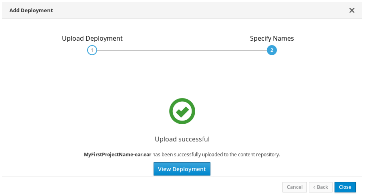
Name *: MyFirstProjectName-ear.ear

Runtime Name: MyFirstProjectName-ear.ear

Enabled: ON

Required fields are marked with *

Upload successful

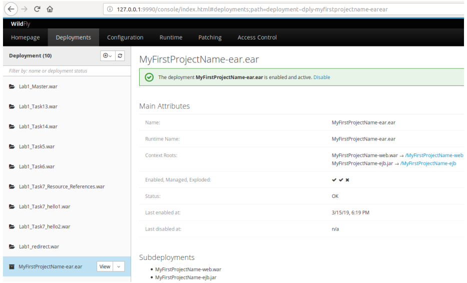


Upload successful

MyFirstProjectName-ear.ear has been successfully uploaded to the content repository.

Wildfly - Deploying using Management user

- Close and you should see all of your deployments including new one



127.0.0.1:9990/console/index.html#deployments;path=deployment~dply-myfirstprojectname-earear

Lab1_Master.war
Lab1_Task13.war
Lab1_Task14.war
Lab1_Task5.war
Lab1_Task6.war
Lab1_Task7_Resource_References.war
Lab1_Task7_hello1.war
Lab1_Task7_hello2.war
Lab1_redirect.war
MyFirstProjectName-ear.ear

MyFirstProjectName-ear.ear

The deployment **MyFirstProjectName-ear.ear** is enabled and active. Disable

Main Attributes

Name:	MyFirstProjectName-ear.ear
Runtime Name:	MyFirstProjectName-ear.ear
Context Roots:	MyFirstProjectName-web.war → /MyFirstProjectName-web MyFirstProjectName-ejb.jar → /MyFirstProjectName-ejb
Enabled, Managed, Exploded:	✓ ✓ ✗
Status:	OK
Last enabled at:	3/15/19, 6:19 PM
Last disabled at:	n/a

Subdeployments

- MyFirstProjectName-web.war
- MyFirstProjectName-ejb.jar

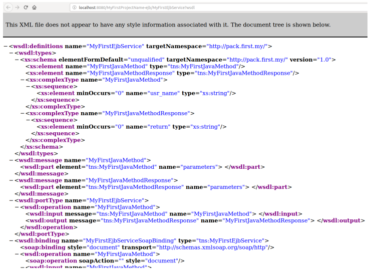
- Go to console running your Wildfly server and you should see:

```
(...) INFO (...) Adding service endpoint metadata: id=MyFirstEjbStateless
address=http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService
implementor=my.first.pack.MyFirstEjb
serviceName=http://pack.first.my/MyFirstEjbService
portName=http://pack.first.my/MyFirstEjbServicePort
annotationWsdlLocation=null
wsdlLocationOverride=null
mtomEnabled=false
(...) INFO (...) Creating Service http://pack.first.my/MyFirstEjbService from class my.first.pack.MyFirstEjb
(...) INFO (...) Setting the server's publish address to be http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService
(...) INFO (...) WSDL published to:
file:/home/jackdaeel/wildfly-14.0.1.Final/standalone/data/wsdl/MyFirstProjectName-ear.ear/MyFirstProjectName-ejb.jar/MyFirstEjbService.wsdl
(...) INFO (...) Starting service jboss.ws.endpoint."MyFirstProjectName-ear.ear"."MyFirstProjectName-ejb.jar".MyFirstEjbStateless
(...) INFO (...) Starting Persistence Unit (phase 2 of 2) Service 'MyFirstProjectName-ear.ear/MyFirstProjectName-ejb.jar#primary'
(...)
(...) INFO (...) Registered web context: '/MyFirstProjectName-ejb' for server 'default-server'
(...) INFO (...) Initializing Mojarra 2.3.5.SP2 for context '/MyFirstProjectName-web'
```

- Now got to:

`http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService?wsdl`

This is what you should see



localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService?wsdl

This XML file does not appear to have any style information associated with it. The document tree is shown below.

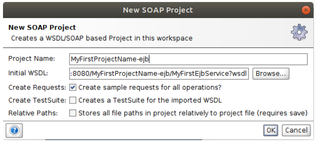
```
– <wsdl:definitions name="MyFirstEjbService" targetNamespace="http://pack.first.my/">
  – <wsdl:types>
    – <xs:schema elementFormDefault="unqualified" targetNamespace="http://pack.first.my/" version="1.0">
        <xs:element name="MyFirstJavaMethod" type="tns:MyFirstJavaMethod"/>
        <xs:element name="MyFirstJavaMethodResponse" type="tns:MyFirstJavaMethodResponse"/>
      – <xs:complexType name="MyFirstJavaMethod">
        – <xs:sequence>
            <xs:element minOccurs="0" name="usr_name" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      – <xs:complexType name="MyFirstJavaMethodResponse">
        – <xs:sequence>
            <xs:element minOccurs="0" name="return" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:schema>
    </wsdl:types>
  – <wsdl:message name="MyFirstJavaMethod">
      <wsdl:part element="tns:MyFirstJavaMethod" name="parameters"> </wsdl:part>
    </wsdl:message>
  – <wsdl:message name="MyFirstJavaMethodResponse">
      <wsdl:part element="tns:MyFirstJavaMethodResponse" name="parameters"> </wsdl:part>
    </wsdl:message>
  – <wsdl:portType name="MyFirstEjbService">
    – <wsdl:operation name="MyFirstJavaMethod">
        <wsdl:input message="tns:MyFirstJavaMethod" name="MyFirstJavaMethod"> </wsdl:input>
        <wsdl:output message="tns:MyFirstJavaMethodResponse" name="MyFirstJavaMethodResponse"> </wsdl:output>
      </wsdl:operation>
    </wsdl:portType>
  – <wsdl:binding name="MyFirstEjbServiceSoapBinding" type="tns:MyFirstEjbService">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    – <wsdl:operation name="MyFirstJavaMethod">
        <soap:operation soapAction="" style="document"/>
      – <wsdl:input name="MyFirstJavaMethod">
```

- The server stores deployments in:

~/wildfly-14.0.1.Final/standalone/deployments/
If you have no deployments on the server this folder should contain only README.txt
If you have issues with deployment try undeploying every .war file and emptying this folder (leaving only README.txt)

SOAP UI - Working with SOAP application

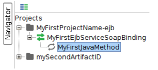
- Open SOAP UI -> File -> New SOAP Project
- Project Name: MyFirstProjectName-ejb
  Initial WSDL: `http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService?wsdl`



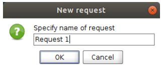
New SOAP Project
Creates a WSDL/SOAP based Project in this workspace

Project Name:	MyFirstProjectName-ejb
Initial WSDL:	:8080/MyFirstProjectName-ejb/MyFirstEjbService?wsdl
Create Requests:	☑ Create sample requests for all operations?
Create TestSuite:	☐ Creates a TestSuite for the imported WSDL
Relative Paths:	☐ Stores all file paths in project relatively to project file (requires save)

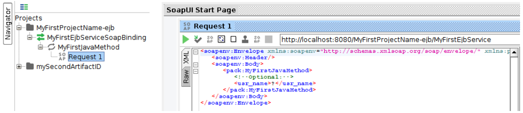
- If you check 'Create Requests: [ ] Create sample requests for all operations?' and click OK
  You will already have generated request for your SOAP application
- Go to right side of SOAP UI and inside Navigator you'll see your new project



- If you don't check 'Create Requests' option you need to:
- Expand 'MyFirstProjectName-ejb' -> 'MyFirstEjbServiceSoapBinding' -> 'MyFirstJavaMethod'
- Right-click 'MyFirstJavaMethod' -> New request
- New request -> Specify name of request: Request 1 -> OK



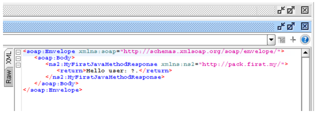
- Expand 'MyFirstJavaMethod' -> Double-click 'Request 1'
- A new window should open inside SOAP UI - Request 1
- In section XML left text box will contain request



http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:p
   <soapenv:Header/>
   <soapenv:Body>
      <pack:MyFirstJavaMethod>
         <!--Optional:-->
         <usr_name>?</usr_name>
      </pack:MyFirstJavaMethod>
   </soapenv:Body>
</soapenv:Envelope>
```

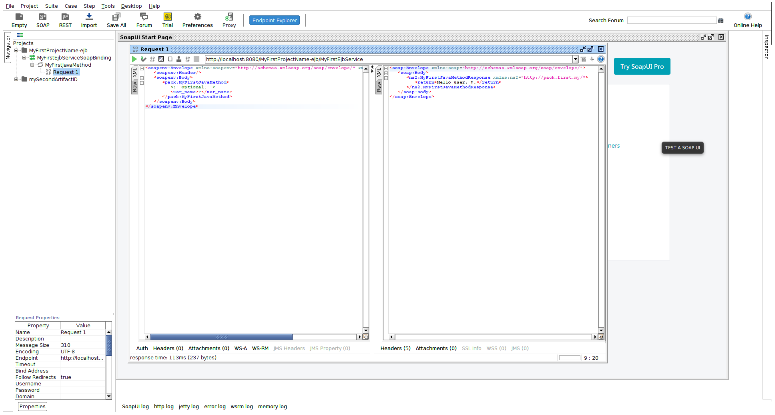
- Click 'Play' Button
  ('Submit request to specified endpoint URL' button)
- Now you should receive answer in the right text box



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:MyFirstJavaMethodResponse xmlns:ns2="http://pack.first.my/">
         <return>Hello user: ?.</return>
      </ns2:MyFirstJavaMethodResponse>
   </soap:Body>
</soap:Envelope>
```

- Now just click 'Save all' Button
- Save the project in new 'SOAP Projects' directory

SOAP UI - Working with SOAP application


File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Endpoint Explorer
Search Forum
Online Help
Navigator
Projects
MyFirstProjectName-ejb
MyFirstEjbServiceSoapBinding
MyFirstJavaMethod
Request 1
mySecondArtifactID
SoapUI Start Page
Request 1
Try SoapUI Pro
TEST A SOAP UI
Inspector
Auth Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)
Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)
SoapUI log http log jetty log error log wsrm log memory log

http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	310
Encoding	UTF-8
Endpoint	http://localhost...
Timeout	
Bind Address	
Follow Redirects	true
Username	
Password	
Domain	

Properties

response time: 113ms (237 bytes)

9 : 20

Request

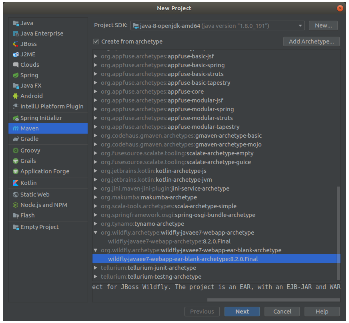
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap
        /envelope/" xmlns:pack="http://pack.first.my/">
   <soapenv:Header/>
   <soapenv:Body>
      <pack:MyFirstJavaMethod>
         <!--Optional:-->
         <usr_name>MyNameIs</usr_name>
      </pack:MyFirstJavaMethod>
   </soapenv:Body>
</soapenv:Envelope>
```

Answer

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
        envelope/">
   <soap:Body>
      <ns2:MyFirstJavaMethodResponse xmlns:ns2="http://pack.
            first.my/">
         <return>Hello user: MyNameIs.</return>
      </ns2:MyFirstJavaMethodResponse>
   </soap:Body>
</soap:Envelope>
```

Intellij - Creating new project

- File -> New -> Project



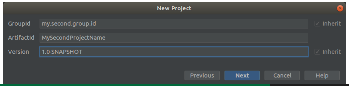
Project SDK: java-8-openjdk-amd64 (java version "1.8.0_191")

☑ Create from archetype

```
▸ org.appfuse.archetypes:appfuse-basic-jsf
▸ org.appfuse.archetypes:appfuse-basic-spring
▸ org.appfuse.archetypes:appfuse-basic-struts
▸ org.appfuse.archetypes:appfuse-basic-tapestry
▸ org.appfuse.archetypes:appfuse-core
▸ org.appfuse.archetypes:appfuse-modular-jsf
▸ org.appfuse.archetypes:appfuse-modular-spring
▸ org.appfuse.archetypes:appfuse-modular-struts
▸ org.appfuse.archetypes:appfuse-modular-tapestry
▸ org.codehaus.gmaven.archetypes:gmaven-archetype-basic
▸ org.codehaus.gmaven.archetypes:gmaven-archetype-mojo
▸ org.fusesource.scalate.tooling:scalate-archetype-empty
▸ org.fusesource.scalate.tooling:scalate-archetype-guice
▸ org.jetbrains.kotlin:kotlin-archetype-js
▸ org.jetbrains.kotlin:kotlin-archetype-jvm
▸ org.jini.maven-jini-plugin:jini-service-archetype
▸ org.makumba:makumba-archetype
▸ org.scala-tools.archetypes:scala-archetype-simple
▸ org.springframework.osgi:spring-osgi-bundle-archetype
▸ org.tynamo:tynamo-archetype
▾ org.wildfly.archetype:wildfly-javaee7-webapp-archetype
    wildfly-javaee7-webapp-archetype:8.2.0.Final
▾ org.wildfly.archetype:wildfly-javaee7-webapp-ear-blank-archetype
    wildfly-javaee7-webapp-ear-blank-archetype:8.2.0.Final
▸ tellurium:tellurium-junit-archetype
▸ tellurium:tellurium-testng-archetype
```

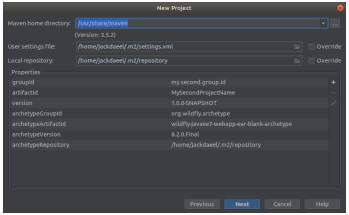
```
ect for JBoss Wildfly. The project is an EAR, with an EJB-JAR and WAR
```

- In 'New project' window
- On the right side choose: Maven
- On top in 'Project SDK' choose java-8-openjdk-amd64
- Check 'Create from archetype' option and Expand:
  org.wildfly.archetype:wildfly-javaee7-webapp-ear-blank-archetype
- Choose: wildfly-javaee7-webapp-ear-blank-archetype:8.2.0.Final



GroupId	my.second.group.id	☑ Inherit
ArtifactId	MySecondProjectName	
Version	1.0-SNAPSHOT	☑ Inherit

- Next -> Fill out:
  GroupId: my.second.grou p.id
  ArtifactId: MySecondProjectName
  Version: 1.0.0-SNAPSHOT

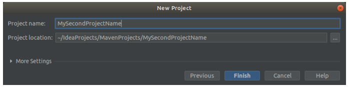


Maven home directory: /usr/share/maven
(Version: 3.5.2)
User settings file: /home/jackdaeel/.m2/settings.xml ☐ Override
Local repository: /home/jackdaeel/.m2/repository ☐ Override

Properties

groupId	my.second.group.id
artifactId	MySecondProjectName
version	1.0.0-SNAPSHOT
archetypeGroupId	org.wildfly.archetype
archetypeArtifactId	wildfly-javaee7-webapp-ear-blank-archetype
archetypeVersion	8.2.0.Final
archetypeRepository	/home/jackdaeel/.m2/repository

- Next -> Set 'Maven home directory:' to /usr/share/maven (Maven home directory)
  Shown how to find Maven location in section **Installing Maven**
- Leave 'User settings file:' and 'Local repository' untouched -> Next

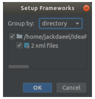


Project name: MySecondProjectName
Project location: ~/IdeaProjects/MavenProjects/MySecondProjectName

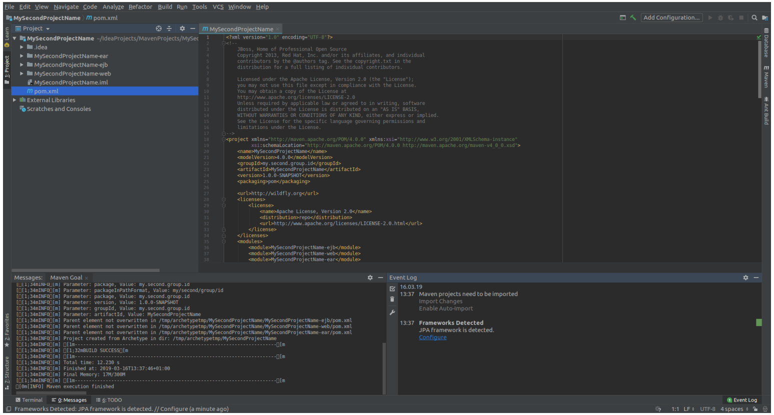
- Fill out:
  Project name: MySecondProjectName
  Project location: /IdeaProjects/MavenProjects/MySecondProjectName
- Finish

Intellij - Configuring new project

- IntelliJ may say 'Maven projects need to be imported'
- Enable auto-import
- It should show up in Right-bottom corner in 'Events' after you click on it
- You should also notice a message:
- Frameworks Detected
  JPA Framework is detected
- Configure -> Group by: Directory -> OK



Group by: directory



```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
    JBoss, Home of Professional Open Source
    Copyright 2013, Red Hat, Inc. and/or its affiliates, and individual
    contributors by the @authors tag. See the copyright.txt in the
    distribution for a full listing of individual contributors.

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<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/maven-v4_0_0.xsd">
    <name>MySecondProjectName</name>
    <modelVersion>4.0.0</modelVersion>
    <groupId>my.second.group.id</groupId>
    <artifactId>MySecondProjectName</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>pom</packaging>

    <url>http://wildfly.org</url>
    <licenses>
        <license>
            <name>Apache License, Version 2.0</name>
            <distribution>repo</distribution>
            <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
        </license>
    </licenses>
    <modules>
        <module>MySecondProjectName-ejb</module>
        <module>MySecondProjectName-web</module>
        <module>MySecondProjectName-ear</module>
```

```
[[1;34mINFO[m] Parameter: package, Value: my.second.group.id
[[1;34mINFO[m] Parameter: packageInPathFormat, Value: my/second/group/id
[[1;34mINFO[m] Parameter: package, Value: my.second.group.id
[[1;34mINFO[m] Parameter: version, Value: 1.0.0-SNAPSHOT
[[1;34mINFO[m] Parameter: groupId, Value: my.second.group.id
[[1;34mINFO[m] Parameter: artifactId, Value: MySecondProjectName
[[1;34mINFO[m] Parent element not overwritten in /tmp/archetypetmp/MySecondProjectName/MySecondProjectName-ejb/pom.xml
[[1;34mINFO[m] Parent element not overwritten in /tmp/archetypetmp/MySecondProjectName/MySecondProjectName-web/pom.xml
[[1;34mINFO[m] Parent element not overwritten in /tmp/archetypetmp/MySecondProjectName/MySecondProjectName-ear/pom.xml
[[1;34mINFO[m] Project created from Archetype in dir: /tmp/archetypetmp/MySecondProjectName
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] [1;32mBUILD SUCCESS[m
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[[1;34mINFO[m] Total time: 12.230 s
[[1;34mINFO[m] Finished at: 2019-03-16T13:37:46+01:00
[[1;34mINFO[m] Final Memory: 17M/300M
[[1;34mINFO[m] [1m------------------------------------------------------------------------[m
[0m[INFO] Maven execution finished
```

```
16.03.19
13:37  Maven projects need to be imported
       Import Changes
       Enable Auto-Import

13:37  Frameworks Detected
       JPA framework is detected.
       Configure
```

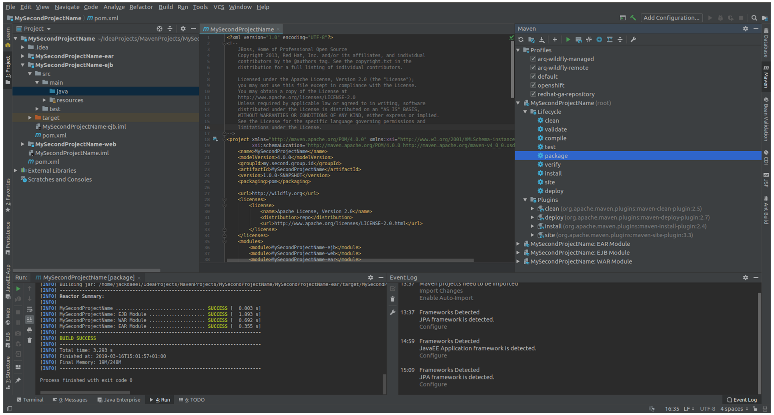
Frameworks Detected: JPA framework is detected. // Configure (a minute ago)

Intellij - Building Project

- On the right side of IDE you should see 'Database', 'Maven', 'Ant build'
- Click 'Maven' -> Expand 'Life cycle' -> Double-click 'Package'
- It should build your package like you would in terminal using command 'mvn clean package'
- In Events tab IntelliJ should detect JavaEE framework
- Configure -> OK
- In 'Maven' tab -> Expand 'Profiles' uncheck 'redhat-ga-repository'
- More profiles should appear: 'arq-wildfly-managed', 'arq-wildfly-remote', 'default', 'openshift' and 'redhat-ga-repository'
- Check them all and configure if IntelliJ is asking to
- Now you can go to 'Project' tab and expand Expand: MySecondProjectName -> MySecondProjectName-ejb -> src -> main -> java
- Right-click 'java'
- Now you should see options in New -> like 'Package' instead of 'Directory' and '... Session Bean'

- If you have any troubles with this process try: Right-click on MySecondProjectName -> Maven -> Generate Sources and Updates Folders and again on MySecondProjectName right-click -> Maven -> Reimport
- Now try creating new Package and EJB
  Shown how to do that in section **IntelliJ - First SOAP application**

  Package name: my.second.pack
  EJB Class: my.second.pack.MySecondEjb



```
MySecondProjectName ~/IdeaProjects/MavenProjects/MySec
  .idea
  MySecondProjectName-ear
  MySecondProjectName-ejb
    src
      main
        java
        resources
      test
    target
    MySecondProjectName-ejb.iml
    pom.xml
  MySecondProjectName-web
  MySecondProjectName.iml
  pom.xml
External Libraries
Scratches and Consoles
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
    JBoss, Home of Professional Open Source
    Copyright 2013, Red Hat, Inc. and/or its affiliates, and individual
    contributors by the @authors tag. See the copyright.txt in the
    distribution for a full listing of individual contributors.

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<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/maven-v4_0_0.xsd
    <name>MySecondProjectName</name>
    <modelVersion>4.0.0</modelVersion>
    <groupId>my.second.group.id</groupId>
    <artifactId>MySecondProjectName</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>pom</packaging>

    <url>http://wildfly.org</url>
    <licenses>
        <license>
            <name>Apache License, Version 2.0</name>
            <distribution>repo</distribution>
            <url>http://www.apache.org/licenses/LICENSE-2.0.html</url>
        </license>
    </licenses>
    <modules>
        <module>MySecondProjectName-ejb</module>
        <module>MySecondProjectName-web</module>
        <module>MySecondProjectName-ear</module>
```

```
Profiles
  ☑ arq-wildfly-managed
  ☑ arq-wildfly-remote
  ☑ default
  ☑ openshift
  ☑ redhat-ga-repository
MySecondProjectName (root)
  Lifecycle
    clean
    validate
    compile
    test
    package
    verify
    install
    site
    deploy
  Plugins
    clean (org.apache.maven.plugins:maven-clean-plugin:2.5)
    deploy (org.apache.maven.plugins:maven-deploy-plugin:2.7)
    install (org.apache.maven.plugins:maven-install-plugin:2.4)
    site (org.apache.maven.plugins:maven-site-plugin:3.3)
MySecondProjectName: EAR Module
MySecondProjectName: EJB Module
MySecondProjectName: WAR Module
```

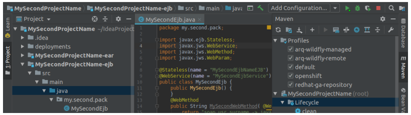
Run: MySecondProjectName [package]

```
[INFO] Building jar: /home/jackdaeel/IdeaProjects/MavenProjects/MySecondProjectName/MySecondProjectName-ear/target/MySecondP
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] MySecondProjectName ................................ SUCCESS [  0.003 s]
[INFO] MySecondProjectName: EJB Module .................... SUCCESS [  1.893 s]
[INFO] MySecondProjectName: WAR Module .................... SUCCESS [  0.692 s]
[INFO] MySecondProjectName: EAR Module .................... SUCCESS [  0.355 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3.293 s
[INFO] Finished at: 2019-03-16T15:01:57+01:00
[INFO] Final Memory: 19M/248M
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

```
13:37 Maven projects need to be imported
      Import Changes
      Enable Auto-Import

13:37 Frameworks Detected
      JPA framework is detected.
      Configure

14:59 Frameworks Detected
      JavaEE Application framework is detected.
      Configure

15:09 Frameworks Detected
      JPA framework is detected.
      Configure
```

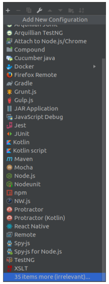
Intellij - Server configuration

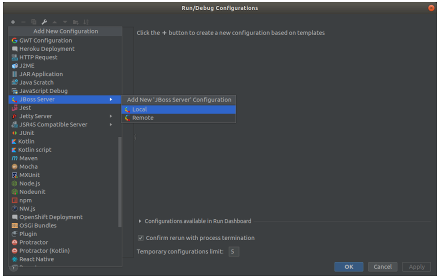
- After creating new EJB your project should look like this:



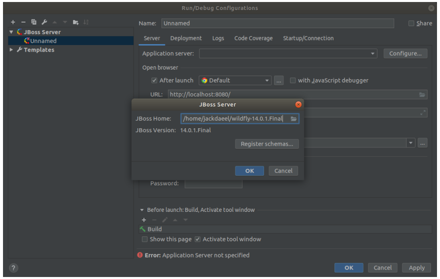
```
package my.second.pack;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@Stateless(name = "MySecondEjbNameEJB")
@WebService(name = "MySecondEjbService")
public class MySecondEjb {
    public MySecondEjb() {
    }
    @WebMethod
    public String MySecondWebMethod( @We
```

- Click on text field 'Add Configuration...'
- Click '+' to add new configuration
- Try to find 'JBoss Server' -> Local
- You may first go to the bottom of the list and click '35 items more (irrelevant)...' - those are relevant - they contain our 'JBoss Server'





Click the + button to create a new configuration based on templates



Name: Unnamed

Application server:

URL: http://localhost:8080/

JBoss Home: /home/jackdaeel/wildfly-14.0.1.Final

JBoss Version: 14.0.1.Final

Before launch: Build, Activate tool window

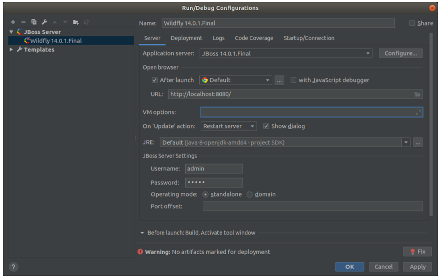
Error: Application Server not specified

- In window 'Run/Debug Configurations' -> 'Server' tab -> Application server: [...] Configure... -> Click 'Configure'
- In window 'JBoss Server' browse to server location

  example: /home/jackdaeel/wildfly-14.0.1.Final
- Click 'OK' when it detetcts version

Intellij - Server configuration

- Fill up 'JBoss Server Settings'
  Username: admin Password: admin Operating mode: *standalone* Port offset: *blank*
- And rename Configuration
- Be sure to check URL changed after adding artifact to:
  urlhttp://localhost:8080/MySecondProjectName-web/


Run/Debug Configurations

JBoss Server
Wildfly 14.0.1.Final
Templates

Name: Wildfly 14.0.1.Final ☐ Share

Server | Deployment | Logs | Code Coverage | Startup/Connection

Application server: JBoss 14.0.1.Final [Configure...]

Open browser
☑ After launch Default ☐ with JavaScript debugger
URL: http://localhost:8080/

VM options:
On 'Update' action: Restart server ☑ Show dialog
JRE: Default (java-8-openjdk-amd64 - project SDK)

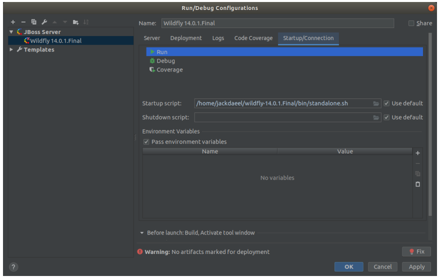
JBoss Server Settings
Username: admin
Password: •••••
Operating mode: [✓ standalone] domain
Port offset:

Before launch: Build, Activate tool window

Warning: No artifacts marked for deployment [Fix]

[OK] [Cancel] [Apply]

- In window 'Run/Debug Configurations' -> 'Startup/Connection' tab
- When you click on 'Run' you should see link to 'Startup script'
  /home/jackdaeel/wildfly-14.0.1.Final/bin/standalone.sh *use default*
- You can change it to (by unchecking 'use default' first):
  /home/jackdaeel/wildfly-14.0.1.Final/bin/standalone.sh -c standalone-full-ha.xml


Run/Debug Configurations

JBoss Server
Wildfly 14.0.1.Final
Templates

Name: Wildfly 14.0.1.Final ☐ Share

Server | Deployment | Logs | Code Coverage | Startup/Connection

Run
Debug
Coverage

Startup script: /home/jackdaeel/wildfly-14.0.1.Final/bin/standalone.sh ☑ Use default
Shutdown script: ☑ Use default

Environment Variables
☑ Pass environment variables

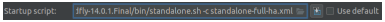
Name	Value

No variables

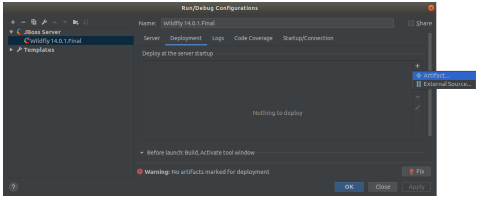
Before launch: Build, Activate tool window

Warning: No artifacts marked for deployment [Fix]

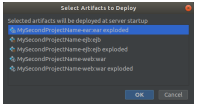
[OK] [Cancel] [Apply]

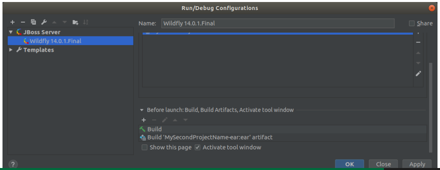

Startup script: ɟfly-14.0.1.Final/bin/standalone.sh -c standalone-full-ha.xml ☐ Use default

Intellij - Server configuration

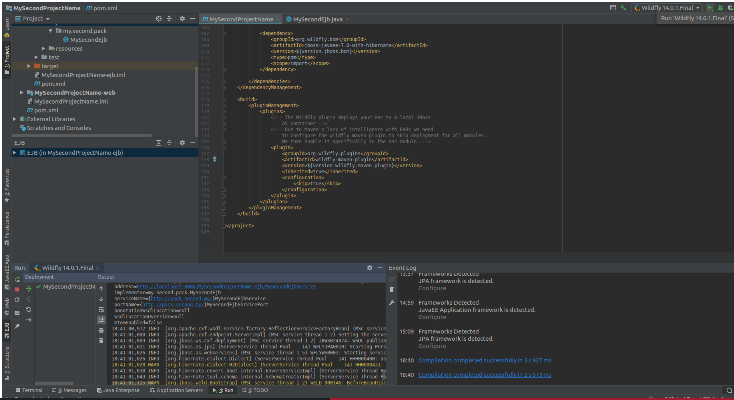


- In window 'Run/Debug Configurations' -> 'Deployment' tab
- Click '+' -> Artifact -> MySecondProjectName-ear:ear -> OK


Select Artifacts to Deploy
Selected artifacts will be deployed at server startup
MySecondProjectName-ear:ear exploded
MySecondProjectName-ejb:ejb
MySecondProjectName-ejb:ejb exploded
MySecondProjectName-web:war
MySecondProjectName-web:war exploded
OK
Cancel


Run/Debug Configurations
JBoss Server
Wildfly 14.0.1.Final
Templates
Name: Wildfly 14.0.1.Final
Share
Before launch: Build, Build Artifacts, Activate tool window
Build
Build 'MySecondProjectName-ear:ear' artifact
Show this page
Activate tool window
OK
Close
Apply

- You should now notice below in section 'Before launch: Build, Build Artifacts, Activate tool window'
  under *Green Hammer* 'Build' position should appear
  'Build "MySecondProjectName-ear:ear" artifact'
- Apply -> OK
- Now in the place of 'Add Configuration...' you should have your configuration (Wildfly 14.0.1.Final)
- Click Play (Run 'Wildfly 14.0.1.Final') button next to it
- This is what you should see:



```
            <dependency>
                <groupId>org.wildfly.bom</groupId>
                <artifactId>jboss-javaee-7.0-with-hibernate</artifactId>
                <version>${version.jboss.bom}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>

        </dependencies>
    </dependencyManagement>

    <build>
        <pluginManagement>
            <plugins>
                <!-- The WildFly plugin deploys your ear to a local JBoss
                    AS container -->
                <!-- Due to Maven's lack of intelligence with EARs we need
                    to configure the wildfly maven plugin to skip deployment for all modules.
                    We then enable it specifically in the ear module. -->
                <plugin>
                    <groupId>org.wildfly.plugins</groupId>
                    <artifactId>wildfly-maven-plugin</artifactId>
                    <version>${version.wildfly.maven.plugin}</version>
                    <inherited>true</inherited>
                    <configuration>
                        <skip>true</skip>
                    </configuration>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>

</project>
```

```
  address=http://localhost:8080/MySecondProjectName-ejb/MySecondEjbService
  implementor=my.second.pack.MySecondEjb
  serviceName={http://pack.second.my/}MySecondEjbService
  portName={http://pack.second.my/}MySecondEjbServicePort
  annotationWsdlLocation=null
  wsdlLocationOverride=null
  mtomEnabled=false
18:41:00,972 INFO  [org.apache.cxf.wsdl.service.factory.ReflectionServiceFactoryBean] (MSC service
18:41:01,000 INFO  [org.apache.cxf.endpoint.ServerImpl] (MSC service thread 1-2) Setting the serve
18:41:01,009 INFO  [org.jboss.ws.cxf.deployment] (MSC service thread 1-2) JBWS024074: WSDL publish
18:41:01,021 INFO  [org.jboss.as.jpa] (ServerService Thread Pool -- 14) WFLYJPA0010: Starting Pers
18:41:01,026 INFO  [org.jboss.as.webservices] (MSC service thread 1-5) WFLYWS0003: Starting servic
18:41:01,026 INFO  [org.hibernate.dialect.Dialect] (ServerService Thread Pool -- 14) HHH000400: Us
18:41:01,028 WARN  [org.hibernate.dialect.H2Dialect] (ServerService Thread Pool -- 14) HHH000431:
18:41:01,038 INFO  [org.hibernate.envers.boot.internal.EnversServiceImpl] (ServerService Thread Po
18:41:01,049 INFO  [org.hibernate.tool.schema.internal.SchemaCreatorImpl] (ServerService Thread Po
18:41:01,115 WARN  [org.jboss.weld.Bootstrap] (MSC service thread 1-2) WELD-000146: BeforeBeanDisc
```

```
13:37 Frameworks Detected
      JPA framework is detected.
      Configure
14:59 Frameworks Detected
      JavaEE Application framework is detected.
      Configure
15:09 Frameworks Detected
      JPA framework is detected.
      Configure
18:40 Compilation completed successfully in 3 s 927 ms
18:40 Compilation completed successfully in 2 s 573 ms
```

Intellij - Deploying to Wildfly

- If you want to deploy using maven commends you can do that after you put inside your main 'pom.xml' file (MySecondProjectName/pom.xml):

```
<project (...)>

   (...)

    <build>
        <pluginManagement>
            <plugins>
                <plugin>
                    <groupId>org.wildfly.plugins</groupId>
                    <artifactId>wildfly-maven-plugin</artifactId>
                    <version>${version.wildfly.maven.plugin}</version>
                    <inherited>true</inherited>
                    <configuration>
                        <skip>true</skip>
                    </configuration>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>

</project>
```

- Click 'Terminal' tab at bottom left

  'Run' tab may be open after deploying through configuration

- Now just type in the Terminal (local):

```
mvn clean package wildfly:deploy
mvn clean package wildfly:redeploy

mvn clean package wildfly:undeploy
```

- If you can't run those commends make sure server is running
- Or try running you command with -e / -X parameter

```
mvn clean package wildfly:deploy -e
mvn clean package wildfly:redeploy -X
```

- For more information visit

  https://www.jetbrains.com/help/idea/creating-run-debug-configuration-for-application-server.html

Intellij - Testing 2nd Project

EJB

```java
package my.second.pack;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@Stateless(name = "MySecondEjbNameEJB")
@WebService(name = "MySecondEjbService")
public class MySecondEjb {
    public MySecondEjb() {}
    @WebMethod
    public String MySecondWebMethod( @WebParam(name = "soap_usr_surname") String java_usr_surname ) {
      System.out.println( "MySecondWebMethod was called with parameter = " + java_usr_surname );
        return "soap_usr_surname-> java_usr_surname-> " + java_usr_surname;
    }
}
```

SOAP Request

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap
            /envelope/" xmlns:pack="http://pack.second.my/">
   <soapenv:Header/>
   <soapenv:Body>
      <pack:MySecondWebMethod>
         <!--Optional:-->
         <soap_usr_surname>MyNameIs</soap_usr_surname>
      </pack:MySecondWebMethod>
   </soapenv:Body>
</soapenv:Envelope>
```

SOAP Answer

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:MySecondWebMethodResponse xmlns:ns2="http://pack.second.my/">
         <return>soap_usr_surname -> java_usr_surname -> MyNameIs</
            return>
      </ns2:MySecondWebMethodResponse>
   </soap:Body>
</soap:Envelope>
```

- After using SOAP UI to send request and receive answer in terminal running Wildfly server you should see:

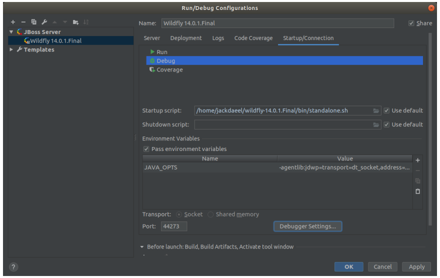
```
(...) INFO [stdout] (default task-1) MySecondWebMethod was called with parameter = MyNameIs
```

Intellij - Debugging - Problems

- Click on Configuration list-box and pick 'Edit Configurations...'
- Check 'Share' checkbox to store configuration information in:
  MySecondProjectName/.idea/runConfigurations/Wildfly_14_0_1_Final.xml (checked 'Share')
  MySecondProjectName/.idea/workspace.xml (unchecked 'Share')

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">

  (...)

   <component name="RunManager">
   <configuration name="Wildfly 14.0.1.Final" type="JBossConfiguration" factoryName="Local"
           APPLICATION_SERVER_NAME="JBoss 14.0.1.Final" ALTERNATIVE_JRE_ENABLED="false">
     <option name="OPEN_IN_BROWSER_URL" value="http://localhost:8080/MySecondProjectName-web/" />
     <option name="UPDATING_POLICY" value="restart-server" />

  (...)

</project>
```

- There a bug while running 'Debug' with latest versions of IntelliJ
  'Application server was not connected before run configuration stop, reason: unable to ping server at localhost:8080'
- Fix:
  Click on Configuration list-box and pick 'Edit Configurations...'
  'Startup/Connection' tab -> choose 'Debug' instead of 'Run' -> 'Debugger Settings...'
  'Debugger Settings...' -> expand 'Debugger' -> 'Async Stack Traces' -> uncheck 'Instrumenting agent (requires debugger restart)'
- Link to thread
  `https://youtrack.jetbrains.com/issue/IDEA-180537`


Run/Debug Configurations

Name: Wildfly 14.0.1.Final ✓ Share

JBoss Server
Wildfly 14.0.1.Final
Templates

Server | Deployment | Logs | Code Coverage | Startup/Connection

Run
Debug
Coverage

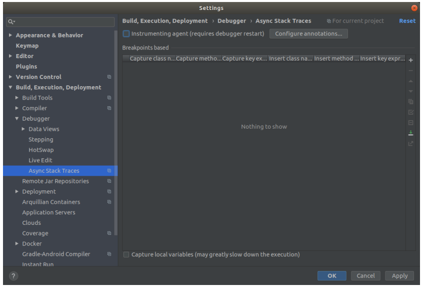
Startup script: /home/jackdaeel/wildfly-14.0.1.Final/bin/standalone.sh ✓ Use default
Shutdown script: ✓ Use default

Environment Variables
✓ Pass environment variables

Name	Value
JAVA_OPTS	-agentlib:jdwp=transport=dt_socket,address=...

Transport: [✓ Socket] Shared memory
Port: 44273
Debugger Settings...

Before launch: Build, Build Artifacts, Activate tool window


Settings

Build, Execution, Deployment › Debugger › Async Stack Traces
For current project
Reset

☐ Instrumenting agent (requires debugger restart)
Configure annotations...

Breakpoints based

Capture class n...	Capture metho...	Capture key ex...	Insert class na...	Insert method ...	Insert key expr...

Nothing to show

☐ Capture local variables (may greatly slow down the execution)

Intellij - Debugging - Problems

- Previous deployments causing errors:
  (...) ERROR [org.jboss.as.controller.management-operation] (Controller Boot Thread) WFLYCTL0013: Operation ("add") failed - address: ([("deployment" =¿ "mySecondArtifactID-ear.ear")]) - failure description: "WFLYSRV0137: No deployment content with hash 31a32a0ee859b9bdf89c1edb4d34cfb85d64b1b5 is available in the deployment content repository for deployment 'mySecondArtifactID-ear.ear'. This is a fatal boot error. To correct the problem, either restart with the –admin-only switch set and use the CLI to install the missing content or remove it from the configuration, or remove the deployment from the xml configuration file and restart."
  (...)
  Just delete fragment <deployment ...> ... </deployment> from
  .../wildfly-14.0.1.Final/standalone/configuration/standalone(...).xml

```xml
<server>
   (...)
   <deployments>
        <deployment name="mySecondArtifactID-ear.ear" runtime-name="
            mySecondArtifactID-ear.ear">
            <content sha1="31a32a0ee859b9bdf89c1edb4d34cfb85d64b1b5"/>
        </deployment>
    </deployments>
</server>
```

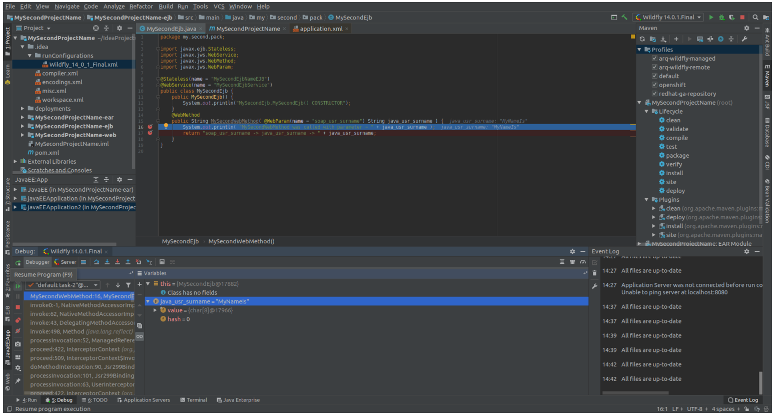
- Use standalone.xml to debug (standalone-full-ha.xml doesn't work for me)
  Config list-box -> Edit configurations... -> Startup/Connections -> Debug -> Startup script: ... *use default* (checked)
- If you use standalone-full-ha.xml you will encounter error:

```
(...) ERROR (...): Operation ("deploy") failed - address: ([("deployment" => "MySecondProjectName-ear.
            ear")]) - failure description: {
    "WFLYCTL0412: Required services that are not installed:" => ["jboss.naming.context.java.jboss.
            DefaultJMSConnectionFactory"],
    "WFLYCTL0180: Services with missing/unavailable dependencies" => [
        "jboss.naming.context.java.comp.MySecondProjectName-ear.MySecondProjectName-ejb.
            MySecondEjbNameEJB.DefaultJMSConnectionFactory is missing [jboss.naming.context.java.jboss
            .DefaultJMSConnectionFactory]",
        "jboss.naming.context.java.module.MySecondProjectName-ear.MySecondProjectName-web.
            DefaultJMSConnectionFactory is missing [jboss.naming.context.java.jboss.
            DefaultJMSConnectionFactory]"
    ]
}
```

How to fix: https://developer.jboss.org/thread/262233?_sscc=t
https://developer.jboss.org/wiki/HowToUseOutOfProcessActiveMQWithWildFly

Intellij - Debugging

- This is how it is supposed to look:



```java
package my.second.pack;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@Stateless(name = "MySecondEjbNameEJB")
@WebService(name = "MySecondEjbService")
public class MySecondEjb {
    public MySecondEjb() {
        System.out.println("MySecondEjb.MySecondEjb() CONSTRUCTOR");
    }
    @WebMethod
    public String MySecondWebMethod( @WebParam(name = "soap_usr_surname") String java_usr_surname ) {  java_usr_surname: "MyNameIs"
        System.out.println( "MySecondWebMethod was called with parameter = " + java_usr_surname );  java_usr_surname: "MyNameIs"
        return "soap_usr_surname -> java_usr_surname -> " + java_usr_surname;
    }
}
```

```
MySecondWebMethod:16, MySecondE
invoke0:-1, NativeMethodAccessorImp
invoke:62, NativeMethodAccessorImp
invoke:43, DelegatingMethodAccesso
invoke:498, Method (java.lang.reflect)
processInvocation:52, ManagedRefere
proceed:422, InterceptorContext (org.
proceed:509, InterceptorContext$Invo
doMethodInterception:90, Jsr299Bind
processInvocation:101, Jsr299Binding
processInvocation:63, UserInterceptor
proceed:422, InterceptorContext (org.
```

```
this = {MySecondEjb@17882}
    Class has no fields
java_usr_surname = "MyNameIs"
    value = {char[8]@17966}
    hash = 0
```

```
14:27  All files are up-to-date
14:27  Application Server was not connected before run co
       Unable to ping server at localhost:8080
14:37  All files are up-to-date
14:37  All files are up-to-date
14:39  All files are up-to-date
14:39  All files are up-to-date
14:42  All files are up-to-date
14:42  All files are up-to-date
```

Resume program execution

Part III

Problems with Wildfly

Table of Contents

Wildfly - The node-identifier attribute

- Issue:

```
(...) WARN [org.jboss.as.txn] (ServerService Thread Pool – 68) WFLYTX0013: The node-identifier attribute on the
/ subsystem=transactions is set to the default value. This is a danger for environments running multiple servers.
Please make sure the attribute value is unique.
```

- Fix:

Wilfly Console (browser) -> Configuration -> Subsystems -> Transaction -> View -> Edit -> Node Indentifier -> $jboss.tx.node.id:1 ===> $jboss.tx.node.id:1234

Wildfly - Debug mode

- To set messages sent to Wildfly terminal to level Debug:

Wilfly Console (browser) -> Configuration -> Subsystems -> Subsystem -> Logging -> Configuration -> Handler ->
"name: CONSOLE, level: INFO->DEBUG, target: System.out"

- You can also creater you own loggers (remember to also creater logger handlers)

Wildfly - Deployment's issues

- You may encounter issues with deployments through any means (IDE, Maven or even Management console deployment)
- Sometimes the project won't build again after small changes
- To fix this issue:

You can force just the project build (without deployment)
Or even manually delete generated files (do a backup of your project)

- Sometimes the package on the server won't update itself
- To fix the second issue:

~/wildfly-14.0.1.Final/standalone/deployments/
If you have no deployments on the server this folder should contain only README.txt
If you have issues with deployment try undeploying every .war file and emptying this folder (leaving only README.txt)

- Ejb(s) can be found in:

~/wildfly-14.0.1.Final/standalone/data/wsdl

Part IV

Appendix

Bibliography I

Wikibooks
LaTeX/Source Code Listings
https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings

Till Tantau, Joseph Wright, Vedran Miletić
The beamer class
http://mirror.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf

Leslie Lamport
LATEX: a document preparation system : user's guide and reference manual
Addison-Wesley Pub. Co., 1994

Bibliography II

Shishir Subramanyam
Java Installation Tutorial
https://developer.jboss.org/wiki/BasicTutorialToSetUpWildFly?fbclid=IwAR0sFMqGlPGsG1uJfPn2oQ9NVn-wpi6kj71rUUdUK7voTNpsU664VrINdV4&_sscc=t

Red Hat Developer Studio team
JBoss (Red Hat) Development Studio Installation Tutorial
https://developers.redhat.com/products/devstudio/download/
After download request you are redirected to tutorial (login required).

Pavithra Gunasekara
Latex Installation Tutorial
https://dzone.com/articles/installing-latex-ubuntu
DZone - Open Source Zone

Bibliography III

Stanisław Polak - Polaksta
beamer-AGH
Github - Latex - AGH Template Presentation
https://github.com/polaksta/beamer-AGH