

# SOA Installation Java & Maven & IntelliJ & Wildfly

Jacek Oleś<sup>1</sup>

<sup>1</sup>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

- 1 Installing Java
- 2 Installing Maven
- 3 Installing IntelliJ
- 4 Installing Wildfly
- 5 Installing SOAP UI

# 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	XBoss
admin	admin	admin	yes
user	user	user	yes

XBoss - 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 commands 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/jackdael/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:

url <https://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

- 6 Maven - New project
- 7 IntelliJ - Importing Project
- 8 IntelliJ - First SOAP application
- 9 Maven - Building & Deploying project
- 10 Wildfly - Deploying using Management user
- 11 SOAP UI - Working with SOAP application
- 12 IntelliJ - Creating new project
- 13 IntelliJ - Building Project
- 14 IntelliJ - Server configuration
- 15 IntelliJ - Deploying to Wildfly
- 16 IntelliJ - Testing 2nd Project
- 17 IntelliJ - Debugging

# 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/jackdael/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-ejb/pom.xml
[INFO] Parent element not overwritten in
/home/jackdael/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-web/pom.xml
[INFO] Parent element not overwritten in
/home/jackdael/IdeaProjects/MavenProjects/MyFirstProjectName/MyFirstProjectName-ear/pom.xml
[INFO] Project created from Archetype in dir: /home/jackdael/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

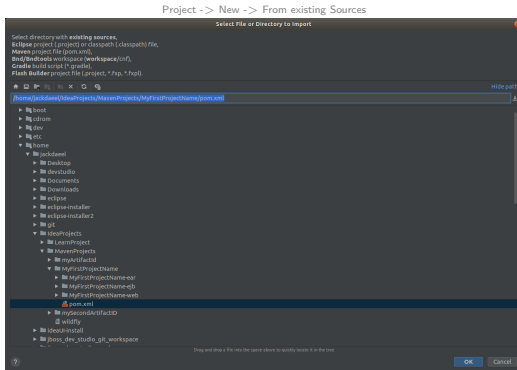
- On 1st generation of project using Maven you may encounter different generating process
- `mvn archetype:generate`

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

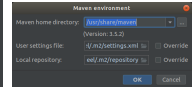
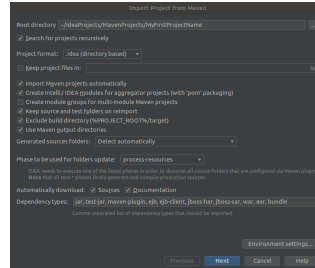
- After launching IntelliJ go to



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

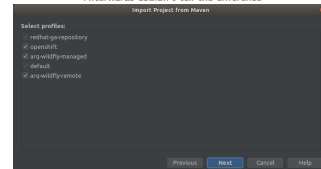
Shown how to find Maven location in section [Installing Maven](#)

- Click OK -> Next



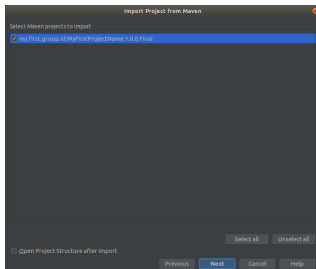
- OK -> Next -> Selected profiles: redhat-ga-repository \*checked\*

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



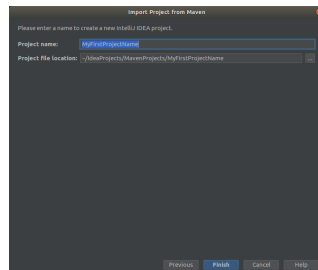
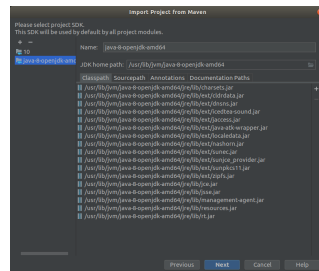
# IntelliJ - Importing Project

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



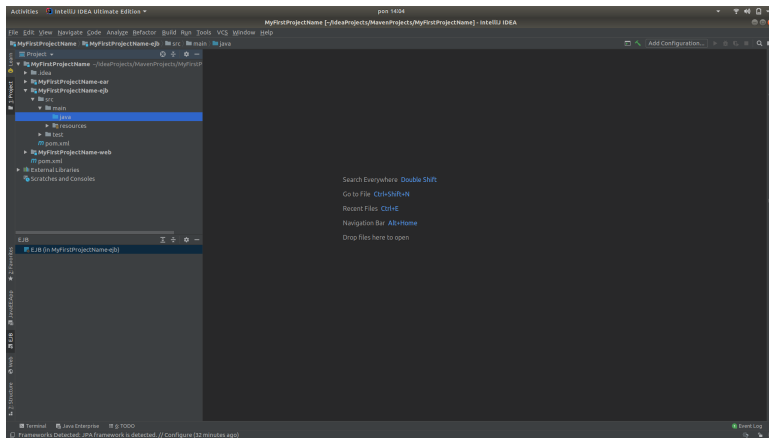
- Next -> Please Select SDK. This SDK will be used by default by all project modules.
- Click "+" -> Add New SDK -> JDK -> Browse to real Java location
- OK -> Name it however you want -> Next

Shown how to find Maven location in section [Installing Java](#)  
I recommend Java 8 ( /usr/lib/jvm/java-8-openjdk-amd64 )



# IntelliJ - Importing Project

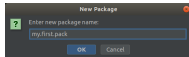
- Finish
- This is how your imported project should look like



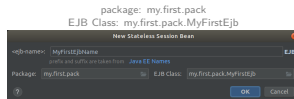


# IntelliJ - First SOAP application

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



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



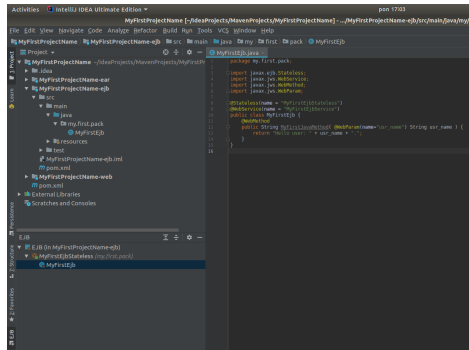
- 8 Click OK
- 9 You should see new generated class like so
- 10 And if you created class through '... Session Bean' You should see 'MyFirstEjb' in EJB view of the project

Fresh class

```
1 package my.first.pack;
2
3 import javax.ejb.Stateless;
4
5 @Stateless(name = "MyFirstEjbName")
6 public class MyFirstEjb {
7     public MyFirstEjb() {
8     }
9 }
```

Modified class

```
1 package my.first.pack;
2
3 import javax.ejb.Stateless;
4 import javax.jws.WebService;
5 import javax.jws.WebMethod;
6 import javax.jws.WebParam;
7
8 @Stateless(name = "MyFirstEjbStateless")
9 @WebService(name = "MyFirstEjbService")
10 public class MyFirstEjb {
11     @WebMethod
12     public String MyFirstJavaMethod( @WebParam(name="usr_name") String usr_name ) {
13         return "Hello user: " + usr_name + ".";
14     }
15 }
```



# 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=myfirst.pack.MyFirstEjb
serviceName=http://pack.first.my/MyFirstEjbService
portName=http://pack.first.my/MyFirstEjbServicePort
annotationWsdLocation=null
wsdlLocationOverride=null
mtomEnabled=false
(...) INFO (...) Creating Service http://pack.first.my/MyFirstEjbService from class myfirst.pack.MyFirstEjb
(...) INFO (...) Setting the server's publish address to be http://localhost:8080/MyFirstProjectName-ejb/MyFirstEjbService
(...) INFO (...) WSDL published to:
file:/home/jackdael/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

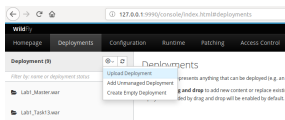
- To deploy package to wildfly you need to have running wildfly server and create management user

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

- If you have done those 2 things go to:

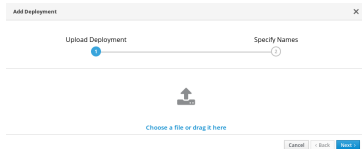
`http://127.0.0.1:9990`

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

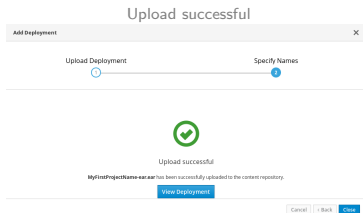
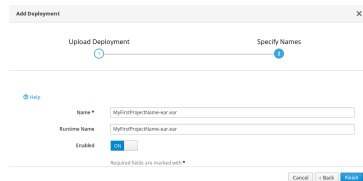


- Select .war file

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



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



# Wildfly - Deploying using Management user

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

Wildfly Management Console - Deployments

Deployment (10)

MyFirstProjectName-ear.ear

The deployment **MyFirstProjectName-ear.ear** is enabled and active. [Disable](#)

Main Attributes	
Name:	MyFirstProjectName-ear.ear
Runtime Name:	MyFirstProjectName-ear.ear
Content Roots:	MyFirstProjectName-web.ear → MyFirstProjectName-web MyFirstProjectName-ear.jar → MyFirstProjectName-ear
Enabled, Managed, Exposed:	✓ ✓ ✗
Status:	OK
Last enabled at:	3/15/19, 8:19 PM
Last disabled at:	na

Subdeployments

- MyFirstProjectName-web.ear
- MyFirstProjectName-ear.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/jackdael/wildfly-14.0.1.Final/standalone/data/wsd/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

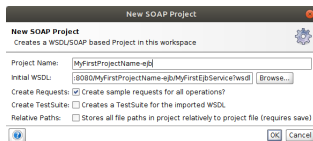
```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="MyFirstEjbService" targetNamespace="http://pack.first.my">
  <wsdl:types>
    <xs:schema elementFormDefault="unqualified" targetNamespace="http://pack.first.my" version="1.0">
      <xs:element name="MyFirstEjbMethod" type="tns:MyFirstEjbMethod"/>
      <xs:element name="MyFirstEjbMethodResponse" type="tns:MyFirstEjbMethodResponse"/>
      <xs:complexType name="MyFirstEjbMethod">
        <xs:sequence>
          <xs:element minOccurs="0" name="arg_name" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="MyFirstEjbMethodResponse">
        <xs:sequence>
          <xs:element minOccurs="0" name="return" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="MyFirstEjbMethod">
    <wsdl:part element="tns:MyFirstEjbMethod" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="MyFirstEjbMethodResponse">
    <wsdl:part element="tns:MyFirstEjbMethodResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="MyFirstEjbService">
    <wsdl:operation name="MyFirstEjbMethod">
      <wsdl:input message="tns:MyFirstEjbMethod" name="MyFirstEjbMethod"/>
      <wsdl:output message="tns:MyFirstEjbMethodResponse" name="MyFirstEjbMethodResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="MyFirstEjbServiceSoapBinding" type="tns:MyFirstEjbService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soaphttp/">
      <soap:operation soapAction="" style="document"/>
    </wsdl:binding>
  </wsdl:binding>
</definitions>
```

- The server stores deployments in:

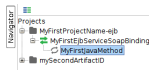
~/wildfly-14.0.1.Final/standalone/deployments/  
If you have no deployments on the server this folder should contain only README.txt  
If you have issues with deployment try undeploying every .war file and emptying this folder (leaving only README.txt)

# 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>

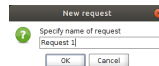


- If you check 'Create Requests: [ ] Create sample requests for all operations?' and click OK  
You will already have generated request for your SOAP application
- Go to right side of SOAP UI and inside Navigator you'll see your new project

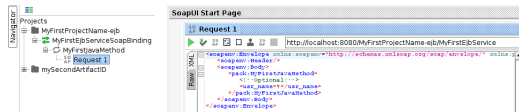


- If you don't check 'Create Requests' option you need to:
- Expand 'MyFirstProjectName-ejb' -> 'MyFirstEjbServiceSoapBinding' -> 'MyFirstJavaMethod'
- Right-click 'MyFirstJavaMethod' -> New request

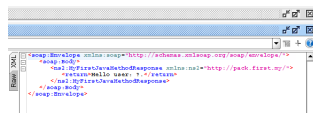
- New request -> Specify name of request: Request 1 -> OK



- 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

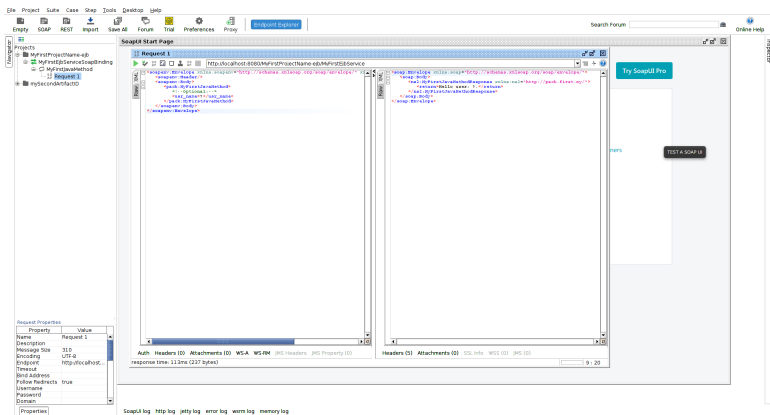


- Click 'Play' Button  
( 'Submit request to specified endpoint URL' button )
- Now you should receive answer in the right text box



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

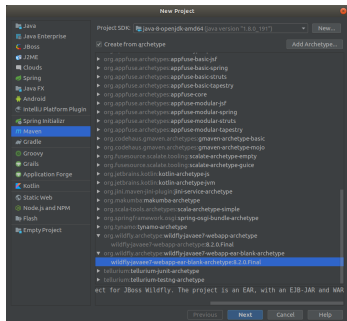
# SOAP UI - Working with SOAP application



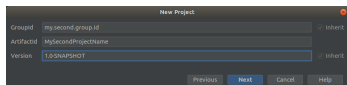
Request	Answer
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:pack="http://pack.first.my"/>	1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <soapenv:Header/>	2 <soap:Body>
3 <soapenv:Body>	3 <ns2:MyFirstJavaMethodResponse xmlns:ns2="http://pack.first.my"/>
4 <pack:MyFirstJavaMethod>	4 <return>Hello user: MyNameIs.</return>
5 <!--Optional:-->	5 </ns2:MyFirstJavaMethodResponse>
6 <usr_name>MyNameIs</usr_name>	6 </soap:Body>
7 </pack:MyFirstJavaMethod>	7 </soap:Envelope>
8 </soapenv:Body>	
9 </soapenv:Envelope>	

# IntelliJ - Creating new project

- File -> New -> Project



- 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

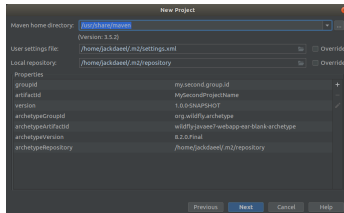


- Next -> Fill out:

GroupId: my.second.group.id

ArtifactId: MySecondProjectName

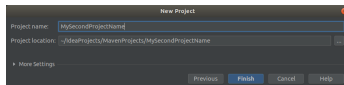
Version: 1.0.0-SNAPSHOT



- 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



- Fill out:

Project name: MySecondProjectName

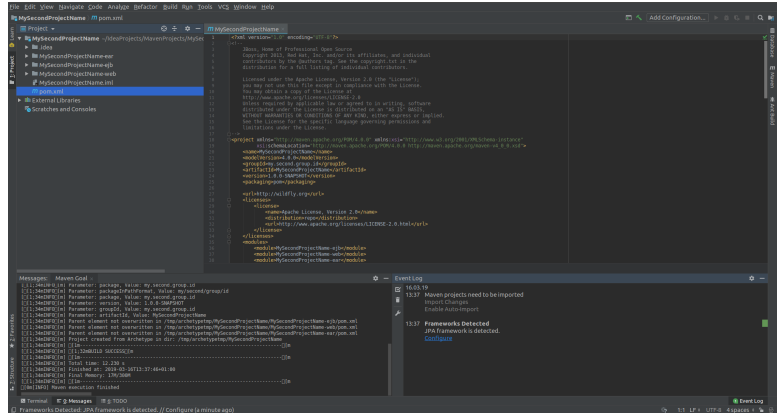
Project location: /IdeaProjects/MavenProjects/MySecondProjectName

- 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

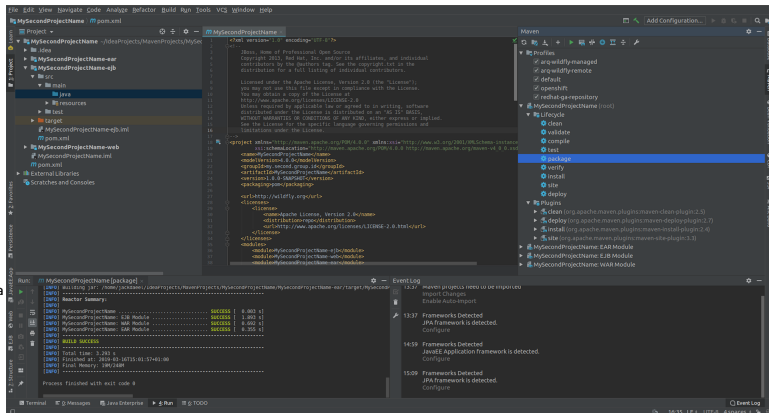




# IntelliJ - Building Project

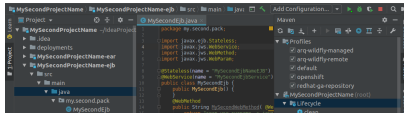
- On the right side of IDE you should see 'Database', 'Maven', 'Ant build'
- Click 'Maven' -> Expand 'Life cycle' -> Double-click 'Package'
- It should build your package like you would in terminal using command 'mvn clean package'
- In Events tab IntelliJ should detect JavaEE framework
- Configure -> 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

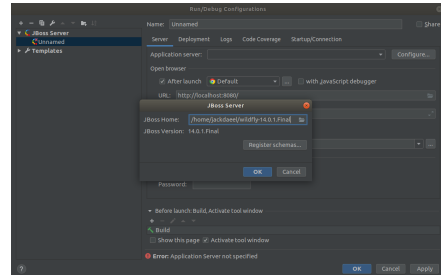
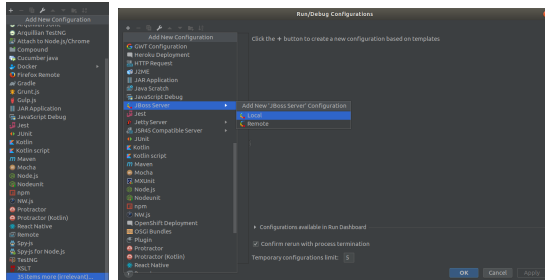


# IntelliJ - Server configuration

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



- 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'



- In window 'Run/Debug Configurations' -> 'Server' tab -> Application server: [ ... ] Configure... -> Click 'Configure'
- In window 'JBoss Server' browse to server location  
example: /home/jackdael/wildfly-14.0.1.Final
- Click 'OK' when it detects version

# IntelliJ - Server configuration

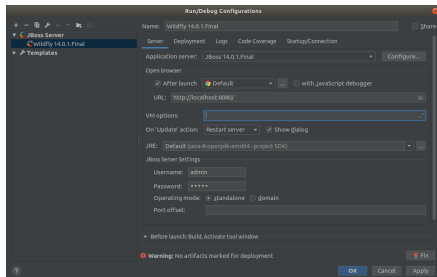
## ● Fill up 'JBoss Server Settings'

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/



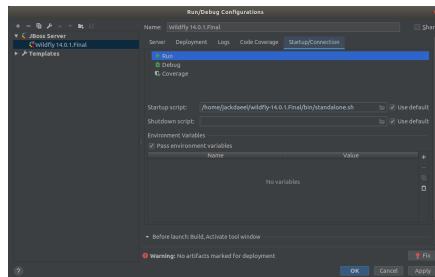
## ● In window 'Run/Debug Configurations' -> 'Startup/Connection' tab

## ● When you click on 'Run' you should see link to 'Startup script'

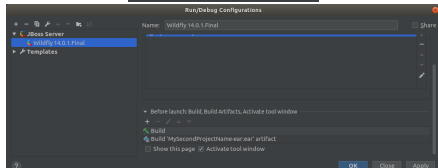
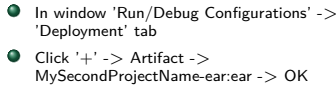
/home/jackdael/wildfly-14.0.1.Final/bin/standalone.sh \*use default\*

## ● You can change it to (by unchecking 'use default' first):

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



Startup script: /fly-14.0.1.Final/bin/standalone.sh -c standalone-full-ha.xml [Use default]



- [illegible]

# IntelliJ - Deploying to Wildfly

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

```

1 <project (...)>
2
3   (...)
4
5   <build>
6     <pluginManagement>
7       <plugins>
8         <plugin>
9           <groupId>org.wildfly.plugins</groupId>
10          <artifactId>wildfly-maven-plugin</artifactId>
11          <version>${version.wildfly.maven.plugin}</version>
12          <inherited>true</inherited>
13          <configuration>
14            <skip>true</skip>
15          </configuration>
16        </plugin>
17      </plugins>
18    </pluginManagement>
19  </build>
20
21 </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:redploy
mvn clean package wildfly:undeploy

```

- If you can't run those commands make sure server is running

- Or try running you command with -e / -X parameter

```

mvn clean package wildfly:deploy -e
mvn clean package wildfly:redploy -X

```

- For more information visit

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

# IntelliJ - Testing 2nd Project

EJB

```

1 package my.second.pack;
2
3 import javax.ejb.Stateless;
4 import javax.jws.WebService;
5 import javax.jws.WebMethod;
6 import javax.jws.WebParam;
7
8 @Stateless(name = "MySecondEjbNameEJB")
9 @WebService(name = "MySecondEjbService")
10 public class MySecondEjb {
11     public MySecondEjb() {}
12
13     @WebMethod
14     public String MySecondWebMethod( @WebParam(name = "soap_usr_surname") String java_usr_surname ) {
15         System.out.println( "MySecondWebMethod was called with parameter = " + java_usr_surname );
16         return "soap_usr_surname-> java_usr_surname-> " + java_usr_surname;
17     }
18 }

```

SOAP Request

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap
   /envelope/" xmlns:pack="http://pack.second.my/">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <pack:MySecondWebMethod>
5       <!--Optional:-->
6       <soap_usr_surname>MyNameIs</soap_usr_surname>
7     </pack:MySecondWebMethod>
8   </soapenv:Body>
9 </soapenv:Envelope>

```

SOAP Answer

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <ns2:MySecondWebMethodResponse xmlns:ns2="http://pack.second.my/">
4       <return>soap_usr_surname -> java_usr_surname -> MyNameIs</
5       return>
6     </ns2:MySecondWebMethodResponse>
7   </soap:Body>
8 </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' )

```

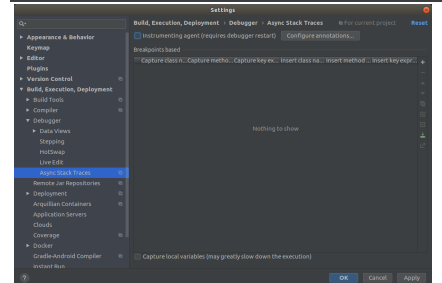
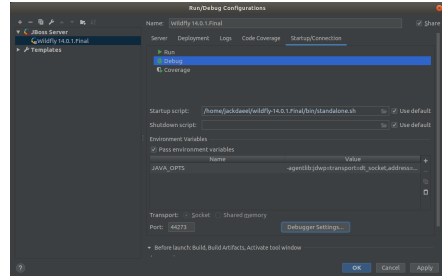
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project version="4">
3
4   (...)
5
6   <component name="RunManager">
7     <configuration name="Wildfly 14.0.1.Final" type="JBossConfiguration" factoryName="Local">
8       APPLICATION_SERVER_NAME="JBoss 14.0.1.Final" ALTERNATIVE_JRE_ENABLED="false">
9         <option name="OPEN_IN_BROWSER_URL" value="http://localhost:8080/MySecondProjectName-web/" />
10        <option name="UPDATING_POLICY" value="restart-server" />
11      </option>
12    </configuration>
13  </component>
14</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>



# 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 31a32a0ee859b9bdf89c1edbd4d34cfb85d64b1b5 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

```

1 <server>
2   (...)
3   <deployments>
4     <deployment name="mySecondArtifactID-ear.ear" runtime-name="
      mySecondArtifactID-ear.ear">
5       <content sha1="31a32a0ee859b9bdf89c1edbd4d34cfb85d64b1b5"/>
6     </deployment>
7   </deployments>
8 </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:

```

1 (...) ERROR (...): Operation ("deploy") failed - address: ([{"deployment" => "MySecondProjectName-ear.
2   ear"}]) - failure description: {
3     "WFLYCTL0412: Required services that are not installed:" => [{"jboss.naming.context.java.jboss.
      DefaultJMSConnectionFactory"},
4     "WFLYCTL0180: Services with missing/unavailable dependencies" => {
5       "jboss.naming.context.java.comp.MySecondProjectName-ear.MySecondProjectName-ejb.
      MySecondEjbNameEJB.DefaultJMSConnectionFactory is missing [jboss.naming.context.java.jboss
      .DefaultJMSConnectionFactory]",
6       "jboss.naming.context.java.module.MySecondProjectName-ear.MySecondProjectName-web.
      DefaultJMSConnectionFactory is missing [jboss.naming.context.java.jboss.
      DefaultJMSConnectionFactory]"
7     }
8   }

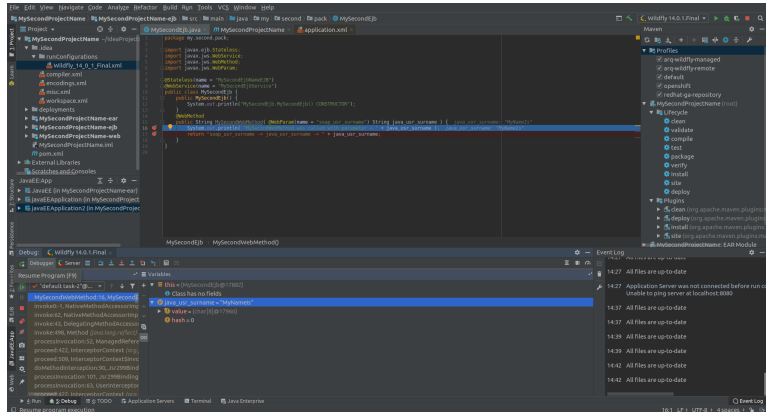
```

How to fix: [https://developer.jboss.org/thread/262233?\\_ssc=t](https://developer.jboss.org/thread/262233?_ssc=t)  
<https://developer.jboss.org/wiki/HowToUseOutOfProcessActiveMQWithWildFly>



# IntelliJ - Debugging

- This is how it is supposed to look:



## Part III

# Problems with Wildfly

# Table of Contents

18 Wildfly - The node-identifier attribute

19 Wildfly - Debug mode

20 Wildfly - Deployment's issues

# 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:

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

# Wildfly - Debug mode

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

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

- You can also create your own loggers (remember to also create 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/wsd/


## Part IV

# Appendix

# Bibliography I

 Wikibooks  
L<sup>A</sup>T<sub>E</sub>X/Source Code Listings  
[https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings)

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

 Leslie Lamport  
L<sup>A</sup>T<sub>E</sub>X: 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=IwAR0sFMqG1PGsG1uJfPn2oQ9NVn-wpi6kj71rUUdUK7voTNpsU664VrINdV4&\\_sscc=t](https://developer.jboss.org/wiki/BasicTutorialToSetUpWildFly?fbclid=IwAR0sFMqG1PGsG1uJfPn2oQ9NVn-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>