INTRO

opinionated guide to sustainable development for those who would like to build modern web based apps in java ee

+some more talk about tools and server setup that will make you productive and as future-proof as possible

SCOPE

We are not covering the whole java ee spec, neither will we go very deep.

- JPA
- EJB
- CDI/Interceptors
- JAXRS
- WebSockets
- Servlet

Notable omissions: JSF related technologies, JCA, JTA or new Batch and JSON specs.

PREREQUISITES

- JDK v1.8x
- Maven v3.x
- Wildfly v9.x
- any IDE you feel most familiar with

HISTORY

1999 J2EE 1.2 was released
2005 Java EE 5 comes with changed name
2009 Java EE 6 goes live - **EE stops being terrible**
2013 Java EE 7

Learn the 7 and you are good until at least 2017(scheduled release of JavaEE 8) :)

ALTERNATIVES

- Spring Framework
  Spring was created as a direct response to the ugliness of J2EE
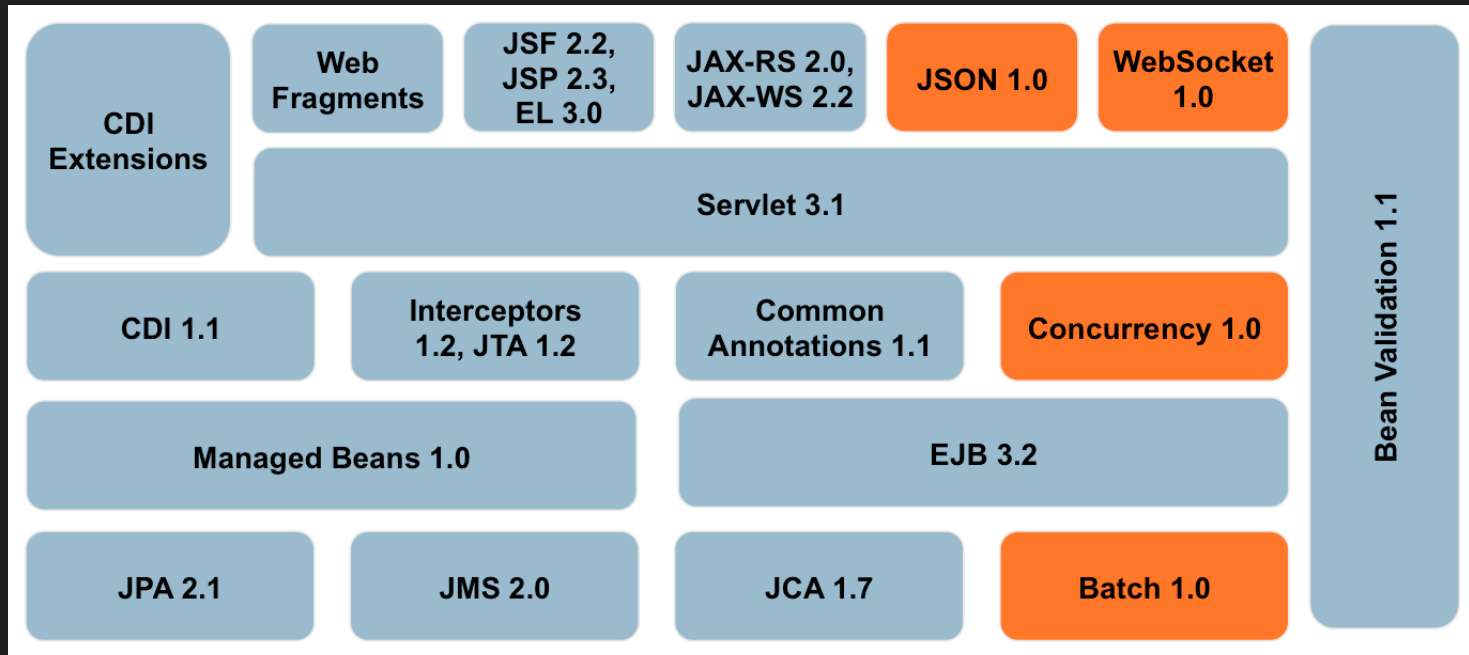- Play Framework
- Spark
- Drop Wizard
- Ninja Framework
- Non-java frameworks for JVM: grails, scalatra, RoR(torguebox)

PARTS

JavaEE 7 = 31 JSR specs created by Java Community Process approved by Executive Comittee

APPLICATION SERVERS

Java EE certified server has to provide implementations for whole Java EE spec.

5 certified java EE 7 servers [wildfly, glassfish, Websphere, Cosminexus, TMax] as of 2015
11 for java EE 6 [JBoss EAP, SAP netweaver, Fujitsu, Hitachi,...]

AS provides additional features - management, deployment, orchestration, etc.

SIMPLEST SETUP

```
mvn archetype:generate com.binarytale.archetypes:javaee7-simple-ar
```

... all you need is love(or 1 dependency)....

```xml
<dependencies>
  <dependency>
    <groupId>javax</groupId>
    <artifactId>javaee-api</artifactId>
    <version>7.0</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

SETUP PART 2

- install the required software
- configure your IDE - new workspace, set default JDK to 1.8, add server definition
- create new blank maven project - packaging = war
- add the javaee-api dependency

-

1 STEP BACK: MAVEN

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="ht
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.test</groupId>
    <artifactId>testwar</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>JavaEE 7 simple archetype</name><!--friendly name-->
    <packaging>war</packaging><!--default jar-->

    <properties></properties>
    <dependencyManagement></dependencyManagement>
    <dependencies></dependencies>
    <build><pluginManagement/></build>
    <profiles></profiles>
  </project>
```

1 STEP BACK: MAVEN BUILD

```
mvn [goals] [-Pprofiles] [-Djvmargs]
```

GOALS

[clean] validate < compile < test < package < integration-test
< verify < install < deploy

DEPENDENCY SCOPES

compile - provided - runtime - test - import

THINGS TO HAVE IN MIND

- transitive dependency
- nearest definition
- exclude

HANDS-ON

Tasks for today:

1. create a blank maven project
2. import to your IDE
3. add the dependency to javaee7
4. setup an app server within IDE
5. build the project and deploy it on app server
6. get it all running without errors
7. import the SampleProject from /hands-on/SampleProject, deploy and inspect

HOMEWORK

1. Get all the tasks from hands-on lab done and working!
2. read the important parts of Hibernate/JPA docs http://docs.jboss.org/hibernate/orm/4.3/devguide/en-US/html_single/ chapters: 4, 5, 7, 8, 13, 16/17.
   fly through http://download.oracle.com/otn-pub/jcp/persistence-2_1-fr-eval-spec/JavaPersistence.pdf chapters 2.9, 2.10, 3.2, 3.5,
3. think about JPA problems you have had
4. which parts of JPA are hard to understand?
5. google this:
   -owner of the relationship
   -hibernate n+1 problem
   -owner of the relationship
6. bookmark this:
   http://docs.jboss.org/hibernate/orm/4.3/manual/en-US/html_single/
   https://github.com/javaee-samples/javaee7-samples