

# JUDCon

JBoss Users & Developers Conference

## 2012:India

### Getting Started Developing Applications for JBoss AS7

Shane Bryzak

# Prerequisites

- JDK 6 or higher
- Eclipse and m2eclipse
- Apache Maven
- Configure JBoss Maven repository  
<https://community.jboss.org/wiki/MavenGettingStarted-Users>
- Git client for downloading quickstarts
- Set **\$JBOSS\_HOME** environment variable

# Downloading JBoss AS7

- <http://www.jboss.org/jbossas/downloads/>
- Extract to local directory
- Test it works!

```
bin/standalone.sh
```

- Browse to <http://localhost:8080>
- Check out the Admin console

# Quickstarts

- Available at:

<https://github.com/jbossas/quickstart>

- Get them with this git command:

```
git clone git@github.com:jbossas/quickstart  
quickstart
```

# Using Maven – Hello World

- Deploy the helloworld example:

```
mvn clean package jboss-as:deploy
```

- Test the app:

<http://localhost:8080/jboss-as-helloworld>

- Undeploy with Maven:

```
mvn jboss-as:undeploy
```

# Using the Command Line Interface (CLI)

- Build the war

```
mvn clean package
```

- Start the CLI

```
$JBOSS_HOME/bin/jboss-admin.sh -connect
```

- Deploy the app

```
deploy target/jboss-as-helloworld.war
```

- Undeploy the app

```
undeploy jboss-as-helloworld.war
```



# Using the web management interface

- First we add a user:

```
$JBASS_HOME/bin/add-user.sh
```

- Open the web management interface

<http://localhost:9990/console>

- Click Manage Deployments -> Add Content
- Browse to `jboss-as-helloworld.war`

# Using the file system

- Build the war

```
mvn clean package
```

- Copy the war

```
cp target/jboss-as-helloworld.war  
$JBOSS_HOME/standalone/deployments
```

- Undeploy the war

```
rm $JBOSS_HOME/standalone/deployments/  
jboss-as-helloworld.war.deployed
```



# Using Eclipse

- Add a JBoss AS Server
  - *Window -> Show View -> Other -> Server -> Servers*
  - *Right click, select New -> Server*
- Start the server
  - *Select server, click start button*
- Deploy the application
  - *Right click, Run As -> Run on Server*

# Let's dive a little deeper

- Hello World example
  - deployment in Java EE6 is simplified!
    - No web.xml required
    - beans.xml is just a marker file
    - index.html just contains a redirect
    - pom.xml is very simple
      - No parent pom
      - Imports for CDI, JSR-250 and Servlet API

# Defining a Servlet

- HelloWorldServlet
  - `@WebServlet` annotation tells the server this class is a servlet
  - We can inject other services using `@Inject`
  - HelloService is a managed bean
  - Injection is *typesafe* – no string-based bean resolution

# A more complex application

- The numberguess example adds a "complete" view layer
- JSF is a server-side rendering, component-oriented view framework
- Views are written in an HTML-like markup

# Numberguess views

- index.html just redirects us to the app
- home.xhtml is where all the action is
  - line 20 displays application messages
  - line 39 to 42 is the input field, bound to the `guess` field of the `game` bean.
  - uses a validator method to ensure the input is valid.

# Numberguess beans

- Game.java is the main application bean
  - @Named exposes the bean in EL
  - @SessionScoped means there is one game per session
  - @Inject allows us to inject the max number, and the random number
  - @PostConstruct is used to set initial values



# kitchensink quickstart

- Bean validation
  - ensures data is always valid
  - expressed in one place, applied at multiple layers
- JAX-RS
  - RESTful web services
  - Configured with simple annotations

# Arquillian

- Integration testing framework
- Managed and remote deployment
- Micro deployments allows you to deploy just what you need to execute your test