

Building Real Java EE applications with JBoss AS

Tomáš Remeš
tremes@redhat.com



What is it?



- „A core framework for rapid-application development in a standards-based environment. Plugins / incremental project enhancement for Java EE, and more.“
- Documentation and download link available on docs.jboss.org/author/display/FORGE/Home
- Source code on [github.com/forge](https://github.com/fuseforge)
- 1.0.0.Final (last week)
- Lead by Lincoln Baxter III

Common usage and integration



- Add '\$FORGE_HOME/bin' to your path and run “forge” from terminal
- Using JBoss Developer Studio Forge runtime

Basic commands



- Using “Tab” makes your work easier

- Some basic commands

```
list-commands --all
```

```
new-project --named jbug_example  
--topLevelPackage com.redhat.jbug.forge
```

```
scaffold setup
```

```
build
```

- You can also use some shell commands (ls, cd, rm, ...)

Persistence



- Setup

- persistence setup**

- Choose your JPA provider and container

- Generate some entities

- entity --named** Customer

- field string --named** userName

- scaffold from-entity**

Most important plugins

- Searching and installing new plugins

```
forge find-plugin as
```

```
forge install-plugin jboss-as-7
```

- AS7 plugin

```
as7 setup
```

```
as7 deploy
```

- Arquillian plugin

```
arquillian setup
```

```
arquillian create-test --class
```

```
com.redhat.jbug.forge.model.Customer.java
```

```
test --profile JBOSS_AS_REMOTE_7.X
```

Other features



- Rest integration

rest setup

rest endpoint-from-entity

`com.redhat.jbug.forge.model.Customer.java`

- Richfaces plugin

richfaces setup

richfaces install-example-facelet

Forge scripts



```
@currentStep = 1;
@v = SHELL.prompt("Start at which step?");
@startAt = Integer.parseInt( "" .equals(v.trim()) ? 1 : v );

def step( cmd ) {

    if ( startAt <= currentStep )
    {
        @SHELL.println();
        if ( SHELL.promptBoolean("Execute " + currentStep + ": " + cmd + " ?") )
        {
            @SHELL.println();
            $cmd;
            @SHELL.println();
            wait;
            clear;
        }
    }
    currentStep ++;
};
clear;
@step("new-project --named conftrack --topLevelPackage com.conftrack");
```

Create your own simple plugin



- Using CDI & Annotation-driven patterns
- Create new project for your plugin
- Install Forge Plugin API

plugins setup

- Create new plugin

plugins new-plugin --named ExamplePlugin

- Generated Plugin implementation already contains methods to implement your new commands

Create your own simple plugin



- Give your plugin a name

```
@Alias("example")
public class ExamplePlugin implements Plugin {
    // commands

}
```

- Specify your new commands with method annotations
`@DefaultCommand` and `@Command`

`@DefaultCommand` – not named. Each plugin may have only one default command.

`@Command` – name optional.

`@Option` – method attribute annotation, which allows you to pass arguments to your command.

Create your own simple plugin



- Make your plugin available to Forge

```
forge source-plugin {PATH_TO_YOUR_PLUGIN}
```

Thank's for your attention and try forging!!

