JBoss Forge Tools in Action

Getting Ready

Environment

- Intellij Idea in Presentation mode
- Open Terminal and increase font
- Go to the CDI directory : `cd $CDI`
- Make sur `generate.fsh` is there
- Launch Forge : `./forge.sh`

Live template in Intellij Idea

frgpro

```
@Inject
private ProjectFactory factory;

private Project getProject(UIExecutionContext context) {
    return Projects.getSelectedProject(factory, context.getUIContext());
}
```

Install the needed Forge addons

```
addon-install-from-git --url https://github.com/jerr/as-forge-addon.git --
coordinate org.jboss.forge.addon:as --ref forge2
addon-install-from-git --url https://github.com/forge/addon-arquillian.git -
-coordinate org.arquillian.forge:arquillian-addon
addon-install-from-git --url https://github.com/forge/wildfly-swarm-
addon.git
```

- Check if there are any plugins that need to be deleted with `addon-list` and `addon-remove --addons`
- Install the needed addons

Java EE is Difficult

Project : Where to start to create a web app ?

```
project-new --named quickdemo
```

Show :

- the project on Intellij IDEA :
- the `pom.xml` and
- the empty Maven directory structure

Persistence : What is JPA, an Entity, Database configuration ?

```
jpa-new-entity --named QuickEntity
```

Show :

- the entity (generated id, version...)
- default package is `model`
- the Hibernate dependency in the `pom.xml`
- the `persistence.xml` configuration

Persistence : How to deal with attribute mapping ?

```
jpa-new-field --named quickstring --columnName quick
```

Show :

- the `@Column` in the entity

Constraints : how to add constraints to the entity ?

```
constraint-add --onProperty quickstring --constraint NotNull
```

Show :

- the `NotNull` on the attribute `quickstring`
- the `validation.xml` file
- in the `pom.xml` there is a new validation api dependency

Constraints : I've heard we can create constraints ?

```
constraint-new-annotation --named MyConstraint
```

Show :

- the constraint `MyConstraint`

- the default package `constraints`

Web UI : how to scaffold pages for the entity like Grails/Rails ?

```
scaffold-generate --provider Faces --targets org.quickdemo.model.QuickEntity
```

Show :

- the JSF backing bean `QuickEntityBean` (stateful EJB, conversation scoped)
- default pacakage is `view`
- the `webapp/quickentity` directory with all the web pages
- the `webapp/resources` directory with the resources (Bootstrap, images, templates..)
- under `WEB-INF` show `beans.xml`, `faces-config` and `web.xml`

REST : how to create a REST interface ?

```
rest-generate-endpoints-from-entities --targets
org.quickdemo.model.QuickEntity
```

Show :

- the `QuickEntityEndpoint` class
- the default package is `rest`
- the `RestApplication` class with the `/rest` path
- Insist on *the Entity now is annotated with @XMLRootElement*

Does is really Build ?

```
build
```

Say that Forge uses Maven to build the application

Deploy it really deploy ?

```
as-setup --server wildfly8 --version 8.2.0.Final
as-start
as-deploy
```

Execute the app

- http://localhost:8080/quickdemo

- http://localhost:8080/quickdemo/rest/quickentities
- http://localhost:8080/quickdemo/rest/quickentities/1

Create entities in the JSF app, show the REST interface, updade and delete data on JSF, and show REST again.

I've created a Java EE application in 5 minutes, and that's because I was talking

Undeploy the app and stop JBoss

```
as-undeploy
as-shutdown
cd ~~
```

Testing a Java EE app, really ?

```
arquillian-setup --testFramework junit --testFrameworkVersion 4.12 --
containerAdapter wildfly-remote --containerAdapterVersion 8.2.0.Final
```

Show :

- the `pom.xml` with the Arquillian plugin, dependency and profile
- the `arquillian.xml`

QuickEntityBeanTest

Integration Test of a JSF Backing Bean ?

```
arquillian-create-test --targets org.quickdemo.view.QuickEntityBean --
enableJPA
```

Show :

- the `QuickEntityBeanTest` with ShrinkWrap
- the `should_be_deployed` method
- execute the test with `build test --profile arquillian-wildfly-remote`
- Add the `QuickEntity` to the test so it passes.

But Forge is not just about Java EE, it's about anything

Modular Architecture with addons

Show Forge slide

CLI and IDE are just addon

- There are addons for Eclipse, Netbeans and Intellij IDEA
- In Intellij IDAE `CTRL+A` type `Forge` create a new bean `MyBean`
- Show `MyBean`
- Back to CLI and in `MyBean` type `cdi-add-observer-method --named quickObserve --eventType java.lang.String`

No Forge dependency in the pom.xml

- Show `pom.xml` with no specific Forge dependencies

Forge parses and generates code on and on

- In CLI, `cd MyBean` and type `ls`
- show the `quickObserve` method
- In Intellij IDEA with Forge, add a new Java field `quick` of type `Integer`
- In CLI, `cd MyBean` and type `ls`
- In Intellij IDEA type `private Float demo` and generated getters and setters
- In CLI, `cd MyBean` and type `ls`

Let's create an addon

Create a new addon project called quickaddon

```
project-new --named quickaddon --type addon
```

In another Forge terminal, create a new project. Show :

- the project on Intellij IDEA :
- the `pom.xml` and
- the empty Maven directory structure

Create a addon called quick

```
addon-new-ui-command --named QuickCommand --commandName quick
addon-build-and-install
```

Show : * the command `QuickCommand` * `name("quick")` is to give the command a name * build the addon with `addon-build-and-install` * test it typing `quick`

Addon to create a Class

```
@Override
public Result execute(UIExecutionContext context) throws Exception {
    JavaClassSource javaClass = Roaster.create(JavaClassSource.class);

javaClass.setName("MyClass").setPackage("org.quickdemo").addField().setName(
"myfield").setType(String.class);

getProject(context).getFacet(JavaSourceFacet.class).saveJavaSource(javaClass
);
    return Results.success("Command 'quick' successfully executed!!!");
}
```

- build the addon with `addon-build-and-install`
- test it typing `quick`
- use it in the `QuickDemo` project

Forge can be scripted

Show generate.fsh

- Open `generate.fsh`

Execute generate.fsh

- In forge `run generate.fsh`

Give it a try and contribute

- It's just a matter of Addons
- Create new addons
- Add new commands to existing addons