

Workshop 3: A Component Based Approach, Java Server Faces

DAT076/DIT126 Chalmers/Göteborgs Universitet
Joachim von Hacht

Objectives

The goal for this workshop is to expose the product catalogue of the shop model as a web application using the component based approach.

You need the following tools and skills;

- GlassFish application server.
- Java Server Faces (JSF) and Facelets (possibly must add as framework in NetBeans).
- Context and Dependency Injection (CDI).
- Bean validation.

Please, have a look at the code samples, everything you need should be there. Also links in slides for more detailed information.

Functionality

To see the intended functionality watch the video (same as for previous workshops)

The JSF request cycle

Download the JSF request cycle demo from course page (located in sample code the zip bundle). Run and inspect output from GlassFish.

- What happens?
- When?

Note: The URL's, to trigger the JSF-system, see `<servlet-mapping>` in `web.xml` (compare how to trigger JAX-RS).

Design and Technical Requirements

See project structure last page

- We will use Facelets. Overall page design is given by `template.xhtml`. Partial are swapped in and out of template.

- We have one JSF page (one partial) for each CRUD operation.
- Each page has a CDI managed bean holding page data (i.e. a backing bean).
- The control layer is a single CDI bean, ProductCtrl. Any modification of the model is handled by the bean.
- The SingletonShop is a bean wrapping the shop model.
- faces-config.xml handles the main menu navigation. There could be (is) other navigation.
- web.xml contains mappings to trigger JSF (JSF Servlet). Also used if implementing authorization.
- beans.xml is needed by CDI.
- All associations between beans should be realized by CDI injection.

Implementing missing parts

1. Download the skeleton code from course page. It should be possible to run. There will possibly be some navigation error messages, no problem leave for now.
2. Start with productList functionality i.e. productList.xhtml. The backing bean is implemented, just use it. Goal is to list the ProductCatalogue.
3. Implement edit and delete functionality
 - a. Create backing beans for edit and delete pages
 - b. Add links from listpage (master) to edit and delete (details) views (use view parameters)
 - c. Implement methods in ProductCtrl.
4. Add validation. Try both JSF validation and Bean validation (prefer). There are some validation messages in src/main/resources. Use!
5. Check that PRG-pattern is implemented.

Have Fun ... (Optional)

Authorization

1. Try to add “the official” JEE style of authorization, see code samples. Use the login page, the User class and the AuthBean class to handle login/logout.

Using a JSF component suite

1. Clone the project and try a higher level component suite (PrimeFaces recommended). This may end in a single page ProductCatalogue (using advanced AJAX table components).

NetBeans Project structure

