**Importing and using a developed software product line with VariaMos and the FragOP approach**

April, 2019

The VariaMos team

**NOTE:** *this tutorial shows how to import and use a SPL which was designed and implemented with the FragOP approach. It already includes some developed models and some developed software components. If you want to follow a tutorial which shows how to design and implement a SPL with the FragOP approach from scratch (implementing from scratch the models and the software components). Then, go to this link:* [*https://docs.google.com/document/d/1d6ERa50Kjr6CQR9zygljLuY0UJX4UBna9BswZ9CKqfg/edit*](https://docs.google.com/document/d/1d6ERa50Kjr6CQR9zygljLuY0UJX4UBna9BswZ9CKqfg/edit)

**FragOP**

Fragment-oriented programming (FragOP) is a framework used to design, implement and reuse domain components in the context of an SPL. FragOP is a mix between compositional and annotative approaches.

FragOP is based on:

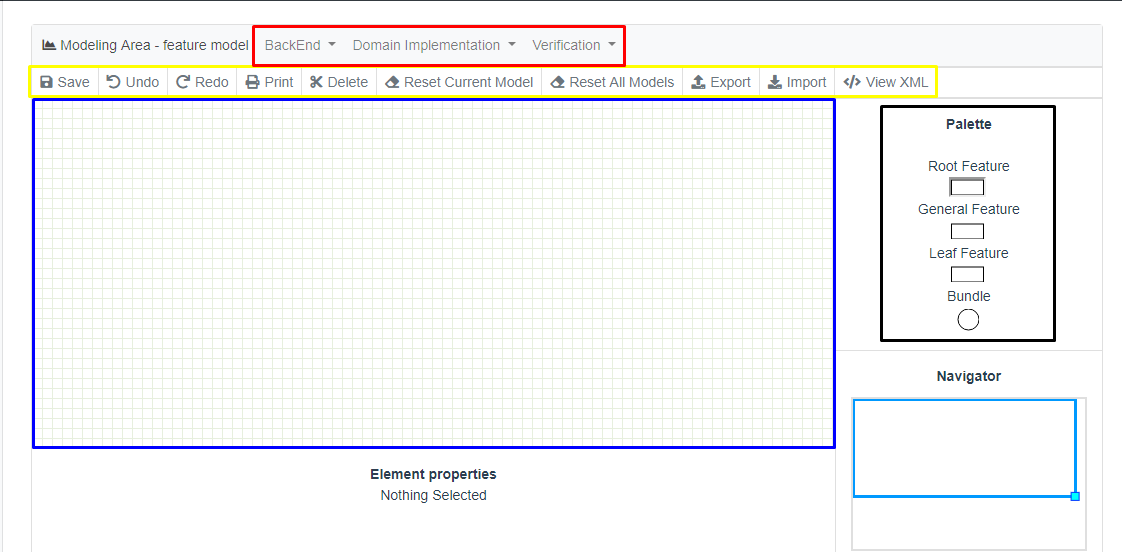
* Domain components
* Fragments
* Fragmentation points
* Customization files
* Customization points

**VariaMos modeling support**

Currently, VariaMos Web version supports three different types of models:

* Feature model
* Component model
* Binding model

Those models are displayed in the modeling area:



In the previous image, we highlighted 5 important sections.

* **Black section** is the Palette, the palette defines what are the elements allowed to create your model, in this case, for example, the Feature model palette contains 4 elements.
* **Blue section** is the Modeling area, this is where you can actually drop the elements from the palette which are going to compose each model.
* **Yellow section** is the toolbar, the toolbar has some functionalities such as import or export the model, with the toolbar we will be saving the model or erasing everything in the modeling area.
* **Red section** contains some functionalities to configure the VariaMos Services (BackEnd) project, execute a product derivation, and execute a product customization, among others.

**Stage 0 - Pre-requirements**

1. Download, install and run VariaMos web “development” version <https://github.com/SPLA/VARIAMOS-WEB/tree/development>
2. Download, install and run VariaMos services <https://github.com/SPLA/VARIAMOS-SERVICES>

**Note:** you can use these tutorials, to download, install, run and even test the previous two projects:

* VariaMos Services: <https://docs.google.com/document/d/1UtLtJ2XxPDLYxlLr7V4LCaV7m1A5w24V__LdUfMwcfc/edit>
* VariaMos Front-end (please remember you should install the “development” version): <https://docs.google.com/document/d/1gU18SYM6A7M61e54O7L6jpvQpdQAhTZahMaPlWxrejk/edit>
* Create and test a service: <https://docs.google.com/document/d/13zgzJ6HZtD8FLkFRZ4dEoDnzDtHqmLx3BPNLbQ-x2Dw/edit>

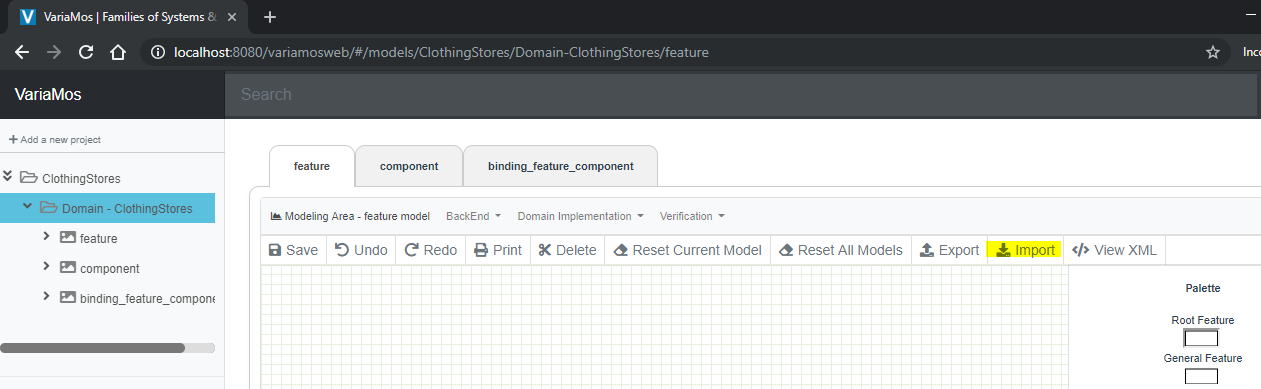
**Stage 1 - Create a new project**

Go to VariaMos home (i.e. <http://localhost:8080/variamosweb/#/>) and click add a new project. Put **“ClothingStores”** as the name.

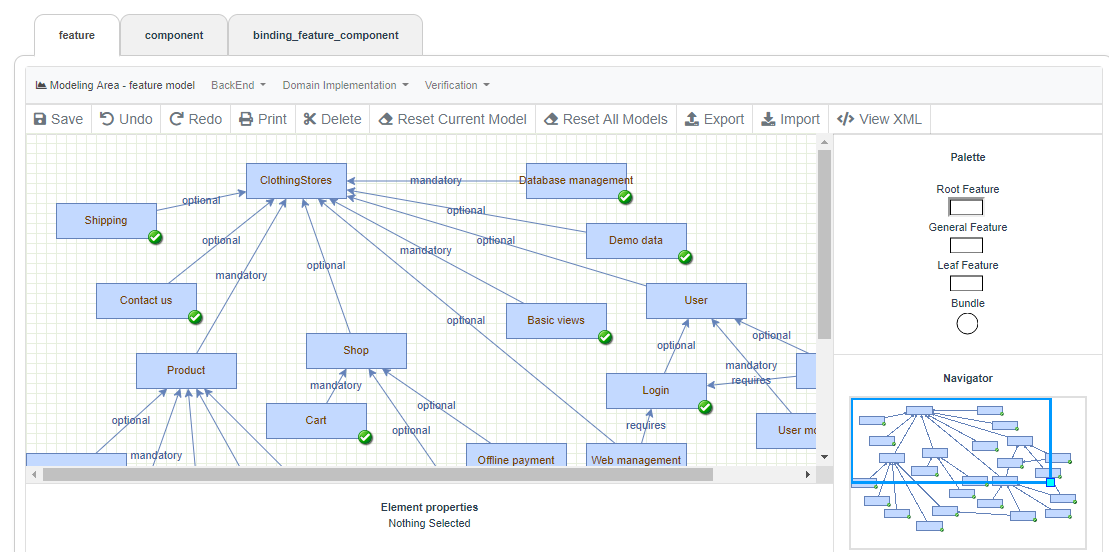


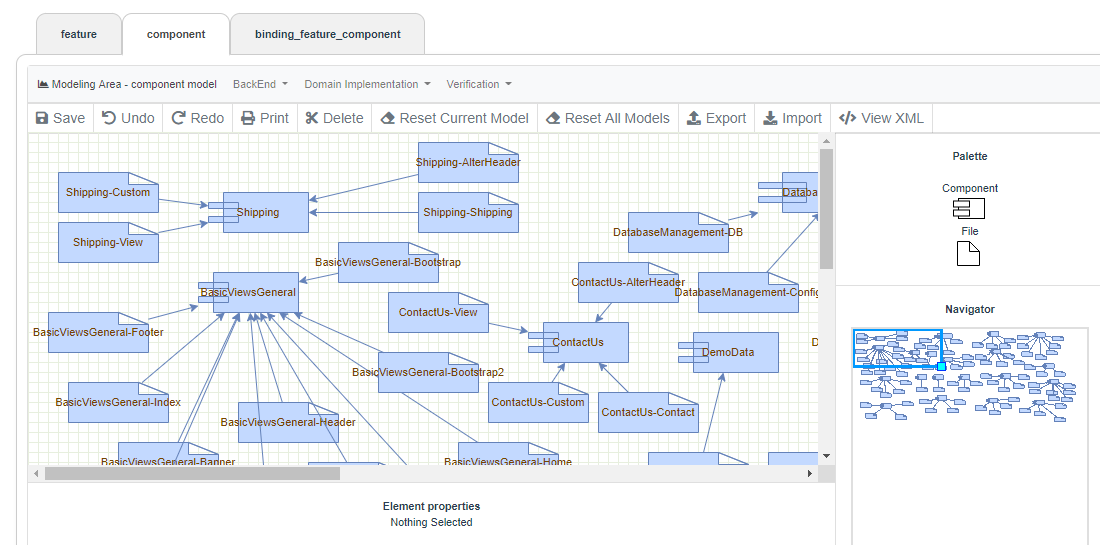
**Stage 2 - Import models**

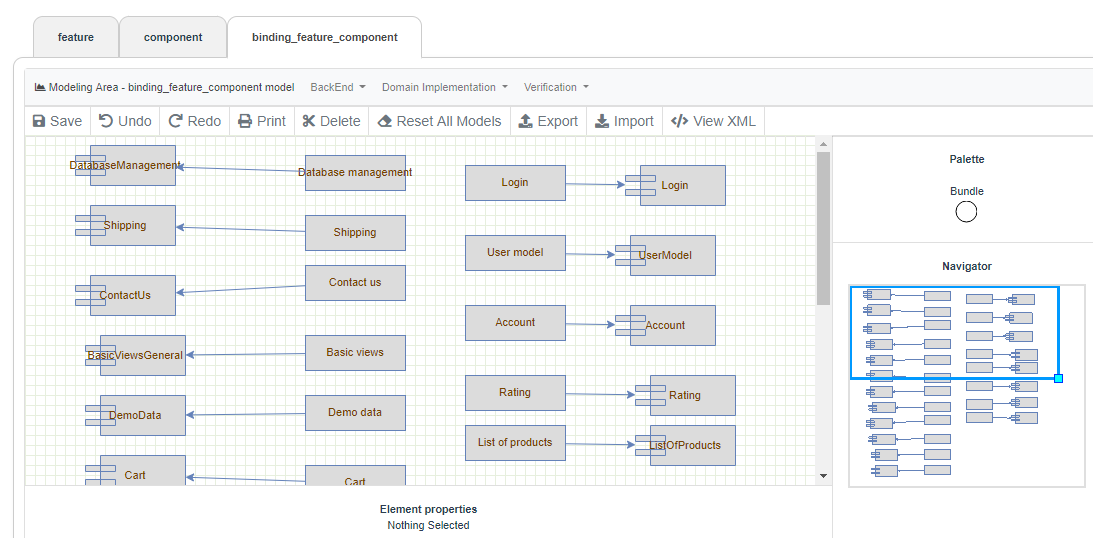
1. Navigate in the project tree area to the Domain folder. You will see three tabs with three models. Click import, and import this file: <https://mega.nz/#!HFJhFQbB!M50ObFpsSVUOgXc5o_IhwYVR0Zs08xBbFrkvSPaW640>



1. Check that you have the following three models: feature, component and binding model.

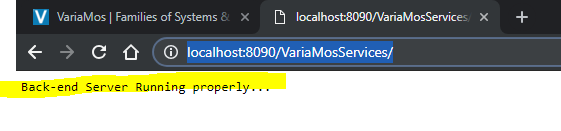






**Stage 3 - VariaMos Services project setup**

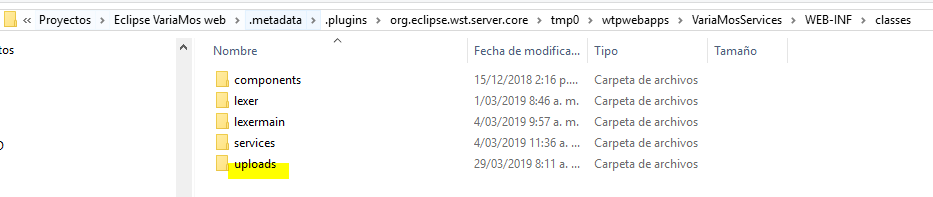
1. Run the VariaMos services project and verify if it runs properly.

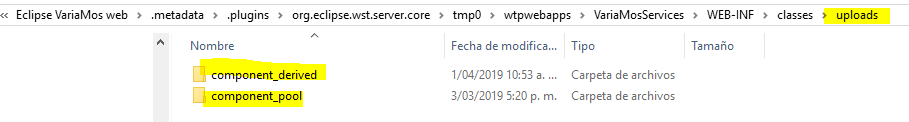


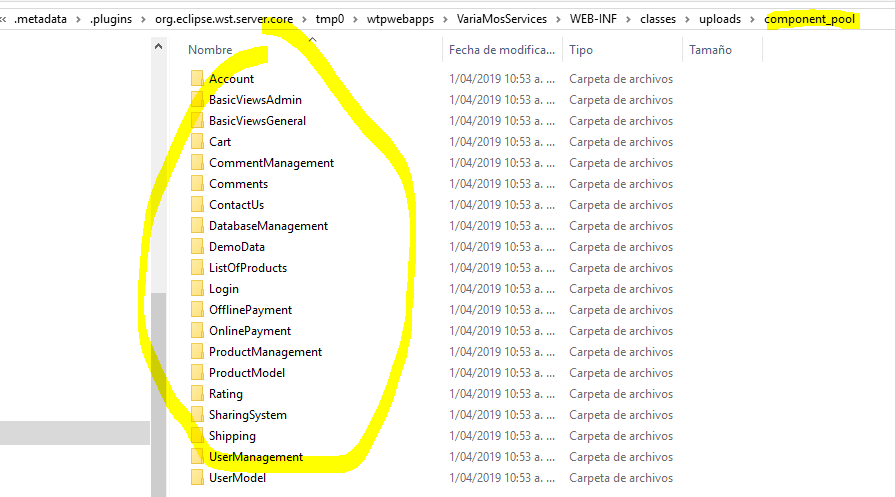
1. Go to the VariaMos Services server path. It should be something like this:

C:/PROJECT\_NAME\_FOLDER/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps/VariaMosServices/WEB-INF/classes/ (make sure you are located in the “classes” folder).

1. Create a folder there called “uploads”
2. Create the “component\_pool” and “component\_derived” folders inside “uploads”
3. Download and unzip this file <https://mega.nz/#!yNBQSIYR!gznqqvbTF2TvgroWpnt9IVMNjTXeFKrDyqXnfl5P0FY> and include the content inside the “component\_pool” folder.
4. Check the results by accessing those folders. It should be like this:

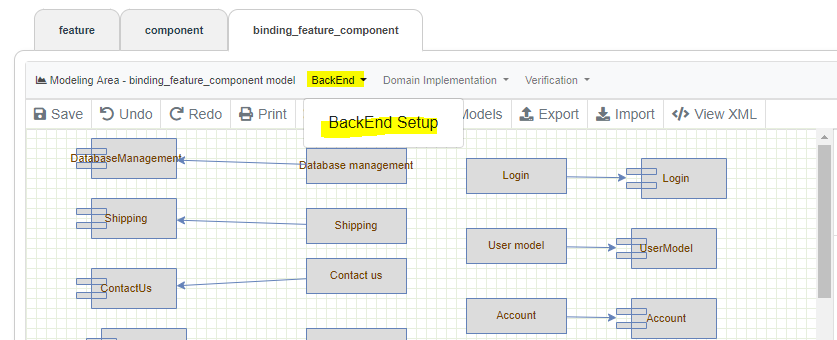




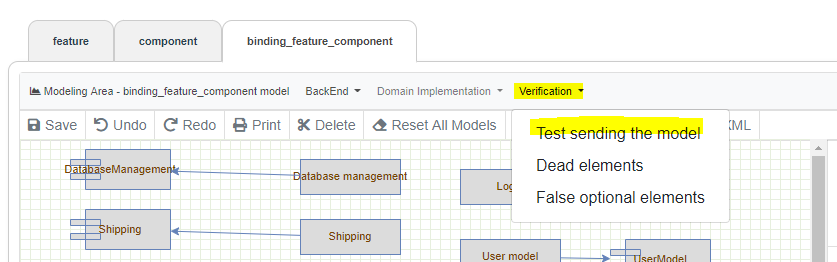


**Stage 4 - VariaMos front-end link**

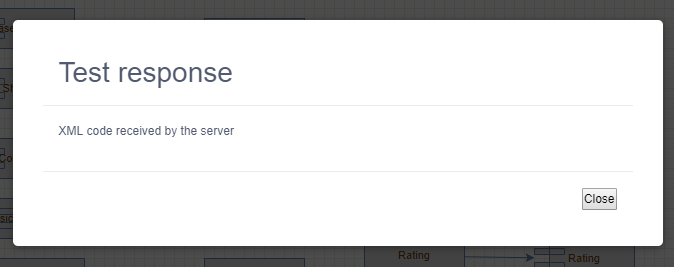
1. Go back to the VariaMos web project. And click “BackEnd -> BackEnd Setup”:



1. Put the main VariaMos services path (i.e. <http://localhost:8090/VariaMosServices/>) and save it.
2. Click Verification -> Test sending the model

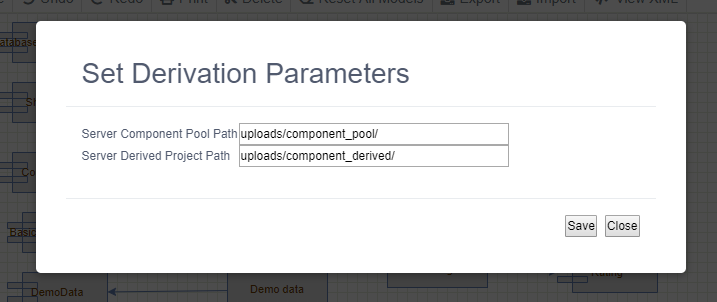


1. If everything is ok, it should display:



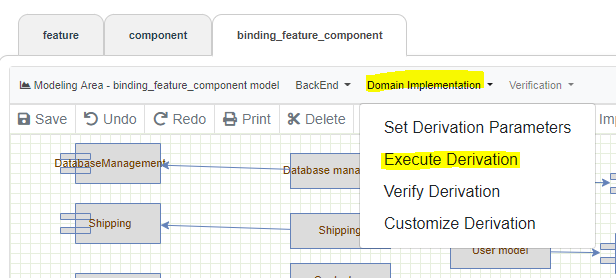
**Stage 5 - Set derivation parameters**

1. Click “Domain implementation” -> Set derivation parameters. And put the next content and save it.

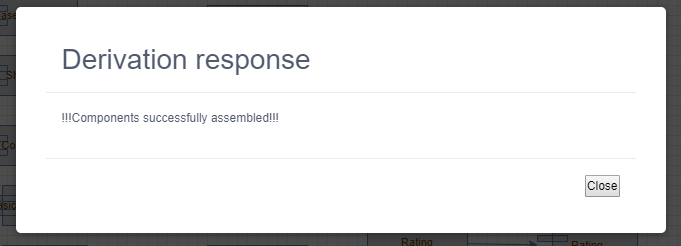


**Stage 6 - Deriving a product.**

1. Click “Domain implementation” -> Execute derivation

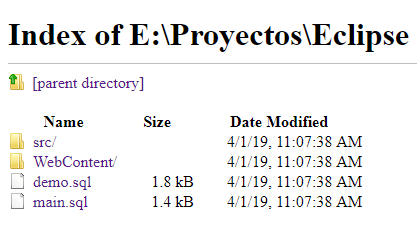


* If everything is ok, it should display:



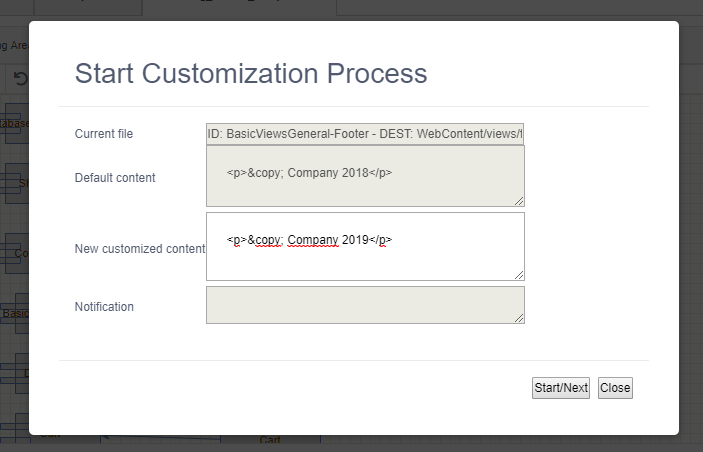
1. Now, you can go to: C:/PROJECT\_NAME\_FOLDER/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps/VariaMosServices/WEB-INF/classes/uploads/component\_derived/

and you will see the derived product.

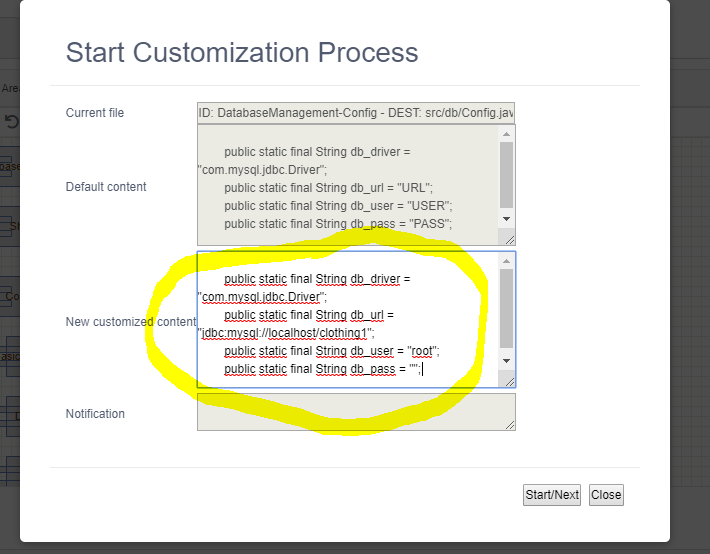


**Stage 7 - Customizing a product**

1. Click “Domain implementation” -> Customize derivation. Then, start to customize the derived product. This is optional.

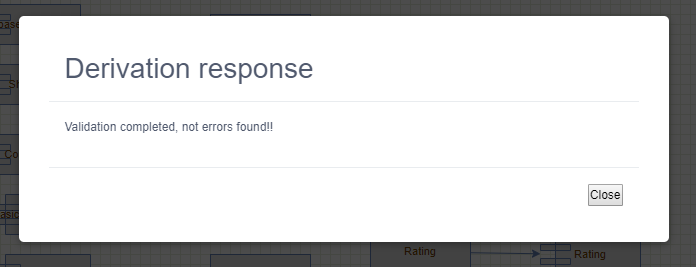


1. Use the customization process to put the real database variable values, check:



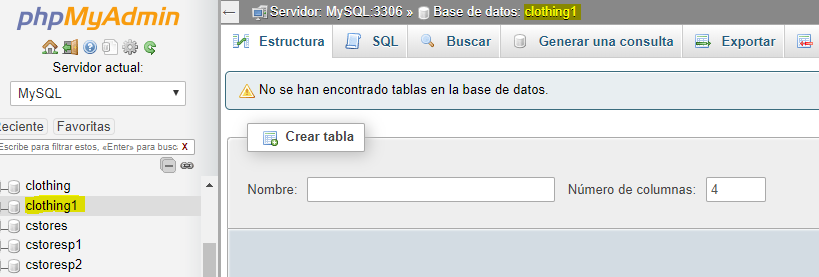
**Stage 8 - Verify derivation**

1. Click “Domain implementation” -> Verify derivation.
2. It could take a while (15-20 seconds). If everything is ok, it should display:

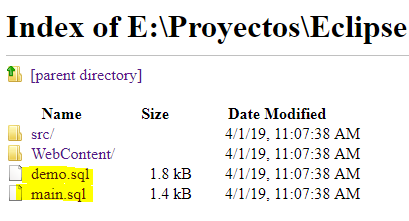


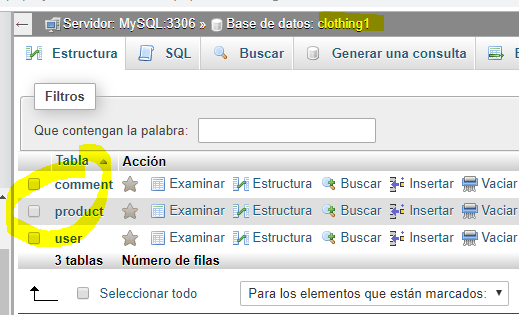
**Stage 9 - Setup the database**

1. Install MySQL and phpmyadmin if you don’t have it.
2. Create your custom database (I named it “clothing1”).



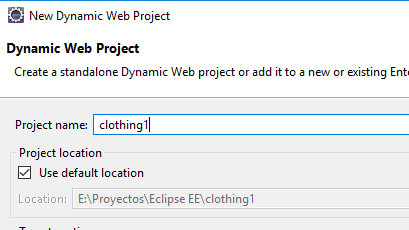
1. Import the generated database files (first import main.sql, then demo.sql):



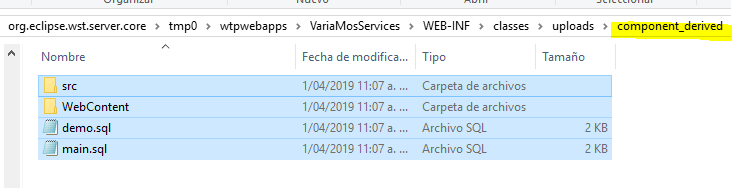


**Stage 10 - Create the new product project**

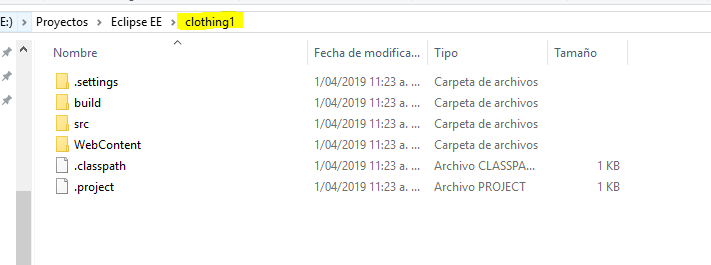
1. Go to Eclipse, and create a new “Dynamic web project” called “clothing1”.



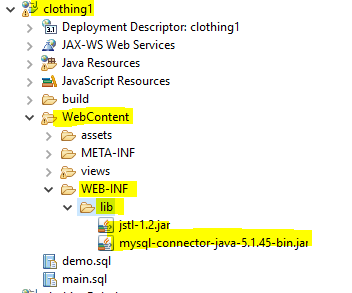
1. Go to the component\_derived folder and copy all the content



1. Go to the new project root folder and paste the previous folders, here:

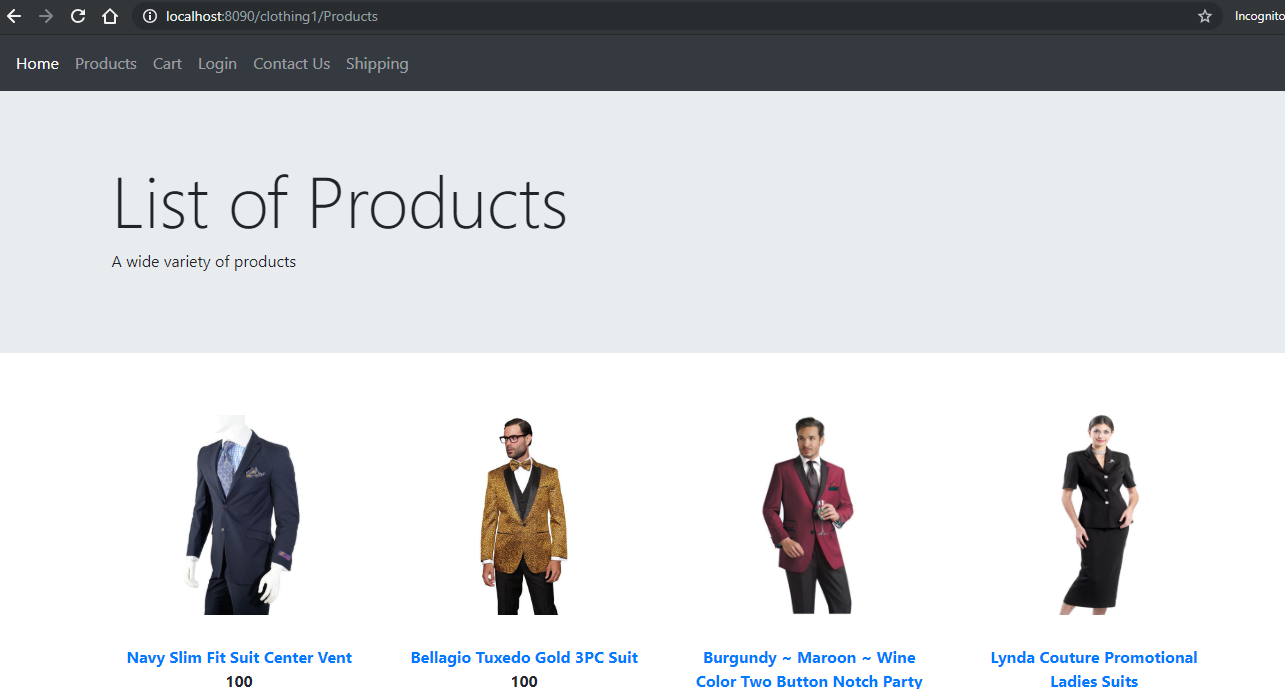


1. Download and include these two libraries in the WEB-INF/lib folder
   * <https://mvnrepository.com/artifact/javax.servlet/jstl/1.2> (download jar file)
   * <https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.45> (download jar file)



**Stage 11 - Run the application**

1. Go to Eclipse and run the new project.



**Deriving several applications at the same time**

If you want to generate different applications (based on the developed components), you just need to select the specific “concrete features” that you want the applications include. Then, you have to replicate step 6 and forward.

