

Lab 8

Create a JSF project with Maven

1. Create a project as instructed for the previous labs but do not include the jsp-api dependency.
2. When deploying to a full-platform Java EE server like GlassFish, you still need to include the following dependency for compilation, but you set the scope as provided so your WAR file won't include these libraries but they will be used for compilation:

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.1.0</version>
  <!-- <scope>provided</scope> -->
</dependency>
```

To use CDI, include

```
<dependency>
  <groupId>javax.enterprise</groupId>
  <artifactId>cdi-api</artifactId>
  <version>1.2</version>
</dependency>
```

Note: for Lab 8, you can add the project exercise05_3 as a dependency

Problem A: Bank Application

The Java project exercise05_3 has a simple application that allows to:

- Create bank accounts
- Retrieve the accounts
- Make deposits
- Withdraw money
- Make deposits in Euros
- Withdraw money in Euros
- Transfer funds from between accounts

There is a **cs545.bank. App** class that serves as example on how to use the services.

Create a web application using JSF for this application. It must include the following functions:

1. Create an account
2. List all accounts
3. View account details
4. Deposit money
5. Withdraw money

The home page of the application must have those five options.

If you want to do more, add the option to select if the deposit/withdraw money is in Euros. You can also add the transfer functionality, letting the operator to choose the accounts from a list – transfers must not be using the same account (from/to).

Note: Inject the AccountService instead of using a singleton class.

I recommend adding a hard-coded account objects during the initialization process to mock data that you could use in the views of the application.