Getting Started With Java EE on WebSphere® Application Server 3h

Java Web Fundamentals 2.5h

The MVC Request Life Cycle 2.5h

Prepare the workshop:

1. Define the goals
2. Create agenda:
   1. Main points
   2. Aids
   3. Main activities with timeline
3. Prepare documents to share with participants

Run the workshop:

1. Run an Ice Breaker - some ball throwing or funny facts.
2. Plan a breaks:
   1. 10 min walkthrough
   2. 45 min session
   3. 10 min for debrief
   4. **10 min coffe**
   5. 10 min walkthrough
   6. 45 min session
   7. 10 min for debrief
   8. **lunch**
   9. 10 min walkthrough
   10. 45 min session
   11. 10 min for debrief
   12. **10 min coffe**
   13. Summary and Q&A
3. Split attendants into small groups (max.3 per group)
4. Run exercises in cycle:
   1. Walkthrough - how to do it
   2. Exercise - do it for a more complex case
   3. Debrief - discuss issues and take aways
5. After lunch run a new Ice Breaker for warm up.

Ideas:

1. Exercise where each team has a bit different task.
2. In the middle of the exercise mix up the teams.
3. In the middle of the exercise make the teams switch the code.

Initial setup:

* Maven
* IntelliJ IDEA or Eclipse
* Apache Tomcat
* MySQL Server
* Make sure your env variables are set up
* Hello World JEE application the same like the one created at the begining of “Java Web Fundamentals” (?)

Objectives:

2 **Java EE Web [40 hours]**

2.1 Servlets

2.2 JSPs y JSTL

2.3 Java EE Architecture

2.3.1 Application Design with Model – View - Controller

2.3.2 Application maintenance

2.3.3 Multilayer Applications

2.4 JDBC

2.4.2 JDBC

2.4.2.1 Integration of database applications

Tasks for workshop:

1. Implement simple MVC for display of bank account.   
   The display should contain:
   1. User data (name, surname)
   2. Account data (account number, current account balance)
   3. List of transactions (date of transaction, amount of founds with + or - sign)

There must be separate bean for user and account data.  
 Data for the display can be static (no need to prepare input or get it from db)  
 Time for exercise: 45 min

1. Refactor your application to have following 3 layers (sub projects):
   1. web - it will be the project you were using so far
   2. service
   3. model

The maven should be reconfigured to build and deploy new application to the tomcat.  
The “Hello World” static string should now come from the model project, through service project, to web project - as a simple test that new architecture works.   
Time for exercise: 45 min

1. Introduce database connection in your application. (<https://tomcat.apache.org/tomcat-7.0-doc/jndi-resources-howto.html>)
   1. Configure Data Source access in Tomcat.
   2. Configure Data Source access in your web application.
   3. Establish connection with newly configured Data Source to extract “Hello World” from the simple table:   
      create table hello (hello varchar(30));  
      insert into hello values(‘Hello World”);

Time for exercise: 45 min

1. Create new tables in the database for UserEntity, AccountEntity and TransactionEntity. Create separate classes for CRUD operations over those tables. Take into consideration relation between those tables.  
   Time for exercise: 45 min
2. Implement communication between the layers for UserEntity, AccountEntity and TransactionEntity:
   1. Prepare DTO classes and converter classes.
   2. Establish communication between the layers for those objects - service layer must provide methods for:
      1. Extracting the user with his accounts.
      2. Extracting account transactions.

Time for exercise: 45 min

1. Implement user login screen.
   1. It must allow user to input his “name.surname”.
   2. In case data input matches any existing user in the application, user should be able to see his account and transactions on this account.
   3. Otherwise application should add the user and create an account for him. Such account should have 0 balance and no transactions.

Time for exercise: 45 min

1. Implement account transaction. Such transaction must give user an ability to move his or her translate funds to the other - target account. This operation must ensure that:
   1. It will create new transaction on user account which will indicate amount of funds moved from the account.
   2. It will create new transaction on target account which will indicate the amount of funds moved to account.
   3. It will update the assets on each account.
   4. All operations will be done in one Transaction.

Time for exercise: 70 min