Web Component Development with Servlet & JSP Technologies (EE 6)

Module-10: More Options for the Model

Team Emertxe

Objectives

Upon completion of this module, you should be able to:

- Understand the nature of the model as a macro-pattern
- Implement persistent storage for web applications using JDBC or Java Persistence API

Relevance

Discussion – The following questions are relevant to understanding what technologies are available for developing web applications and the limitations of those technologies:

- What goes into a model component? Is it actually just a simple piece of the application, or more than that?
- How can you provide persistent storage for data in your web applications?

The Model as a Macro-Pattern

In the MVC architecture pattern, the model component embodies a great deal of functionality. Here are some of the responsibilities that fall to the model as it has been discussed so far:

- Present an interface to the controller
- Implement the business logic
- Present a JavaBeans compliant view of the data to the view component, in a form that is convenient for the view to use.

The View Helper Pattern

According to OO principles, any object is designed best if designed without considering the way in which it will be used. This is because if it were to be designed specifically to support a certain usage style, and that usage style changes, then the object must be changed too. A foundational premise of OO is that good design minimizes the consequences of change.

Database and Resource Access

Virtually all practical web applications will require access to a database or other external support system. Good OO requires that the elements of the model that handle this should be separated from the elements of the model that represent the domain objects and the elements that perform business operations on those objects.

The desire to separate these concerns leads to the Data Access Object pattern, also known as DAO.

Data Access Object (DAO) Pattern

The DAO pattern eases maintenance of applications that use databases by separating the business logic from the data access (data storage) logic. The data access implementation (perhaps JDBC technology calls) is encapsulated in DAO classes.

The DAO pattern permits the business logic and the data access logic to change independently, increasing the flexibility of your application. For example, if the database schema changes, you only need to change the DAO methods, and not the business services or the domain objects.

DAO Pattern Advantages

The DAO pattern has the following advantages:

- Domain objects and persistence logic are now separate.

  The domain objects do not need to know how their persistence is handled.

- The data access objects promote reuse and flexibility in changing the system.

  New domain objects and business services can be constructed that reuse the data access logic in the DAO classes.

- Developers writing other clients, whether servlets or regular client code, can reuse the same data access code.

JDBC API

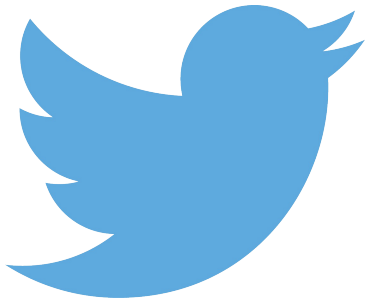
We have already discussed JDBC just after Core Java.

Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You