

# Unit 3 Lesson Plan 2 - Modifying a SOAP DB Emulator Web Service

Taylor Mommer

**Product DB Class Tester:**

Workspace - [http://localhost:9080/MomT\\_CIS266/ProductClassTesters/ProductDBClassTester.jsp](http://localhost:9080/MomT_CIS266/ProductClassTesters/ProductDBClassTester.jsp) - Rational® Application Developer for WebSphere® Software

File Edit View Window Help

ProductDB Class Tester

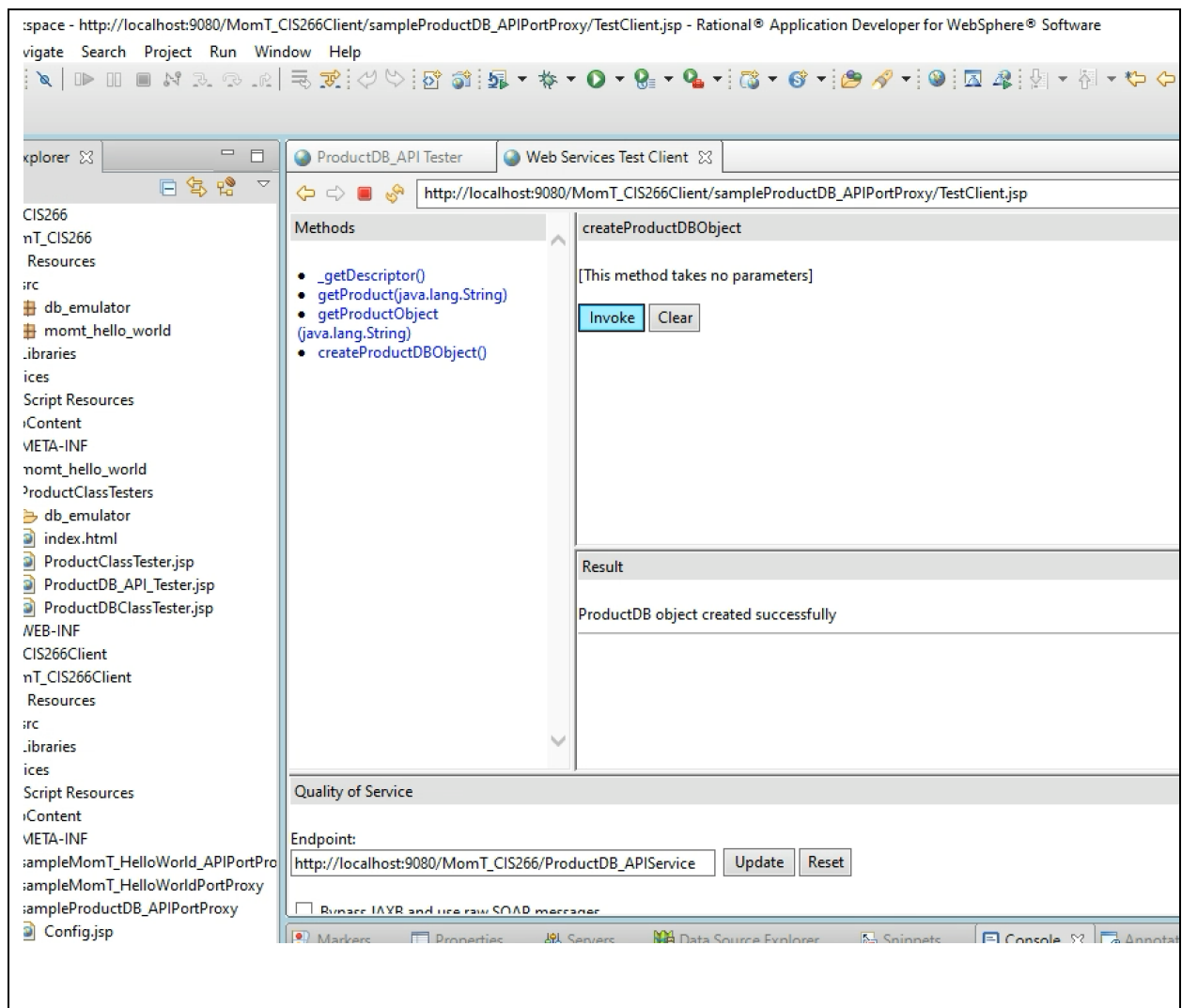
[http://localhost:9080/MomT\\_CIS266/ProductClassTesters/ProductDBClassTester.jsp](http://localhost:9080/MomT_CIS266/ProductClassTesters/ProductDBClassTester.jsp)

**ProductDB Class Tester:**

Output from toString() after calling the static method getProduct(String) which returns a Product object  
Product Code: momt, Product Description: Taylor Mommer's DB Record, Product Price: \$59.99

File Explorer

- CIS266
  - momt\_hello\_world
  - Resources
  - irc
    - db\_emulator
  - libraries
  - ices
  - Script Resources
  - Content
  - META-INF
  - momt\_hello\_world
  - ProductClassTesters
    - db\_emulator
    - index.html
    - ProductClassTester.jsp
    - ProductDB\_API\_Test.jsp
    - ProductDBClassTester.jsp
  - WEB-INF
  - CIS266Client
  - momt\_hello\_worldClient
  - Resources
  - irc
  - libraries
  - ices
  - Script Resources
  - Content
  - META-INF
  - sampleMomT\_HelloWorld\_APIPortPro
  - sampleMomT\_HelloWorldPortProxy
  - sampleProductDB\_APIPortProxy
  - Config.jsp



The screenshot shows a Web Services Test Client window with the following details:

- URL:** `http://localhost:9080/MomT_CIS266Client/sampleProductDB_APIPortProxy/TestClient.jsp`
- Methods:** A list of methods including `_getDescriptor()`, `getProduct(java.lang.String)`, `getProductObject(java.lang.String)`, and `createProductDBObject()`.
- Selected Method:** `getProduct`
- Arguments:** `arg0: momt`
- Buttons:** `Invoke` and `Clear`
- Result:** `Product Code: momt, Product Description: Taylor Mommer's DB Record, Product Price: $59.99`
- Quality of Service:**
  - Endpoint:** `http://localhost:9080/MomT_CIS266/ProductDB_APIService` (with `Update` and `Reset` buttons)
  - ☐ `Remove JAX-RS and use raw SOAP message`
- Bottom Panel:** Includes tabs for `Markers`, `Properties`, `Servers`, `Data Source Explorer`, `Snippets`, `Console`, and `Annotations`. The `Console` tab shows a log entry: `WebSphere Application Server traditional V9.0 at localhost (WebSphere Application Server traditional V9.0) [4/2/23 22:39:43:042 CDT] 00000082 ServletWrapper I com.ibm.ws.webcontainer.servlet.ServletWrapper init SRVE0242I: [MomT`

ProductDB.java Code:

```
package db_emulator;

public class ProductDB
{
    public static Product getProduct(String productCode)
    {
        // In a more realistic application, this code would
        // get the data for the product from a file or database
        // For now, this code just uses if/else statements
        // to return the correct product data

        // create the Product object
        Product product = new Product();

        // fill the Product object with data based on productCode received
        product.setCode(productCode);

        if (productCode.equalsIgnoreCase("java"))
        {
            product.setDescription("Murach's Beginning Java 2");
            product.setPrice(49.50);
        }
    }
}
```

```
}
else if (productCode.equalsIgnoreCase("jsps"))
{
    product.setDescription("Murach's Java Servlets and JSP");
    product.setPrice(49.50);
}
else if (productCode.equalsIgnoreCase("mcb2"))
{
    product.setDescription("Murach's Mainframe COBOL");
    product.setPrice(59.50);
}
else if (productCode.equalsIgnoreCase("txtp"))
{
    product.setDescription("TextPad");
    product.setPrice(20.00);
}
else if (productCode.equalsIgnoreCase("momt"))
{
    product.setDescription("Taylor Mommer's DB Record");
    product.setPrice(59.99);
}
else
{
    product.setDescription("Unknown Product Code");
}
return product;
}
}
```