TEAM 3 — SCRUM 4 JAVA WEB APP AND SOAP WEB SERVICES

TEAM3_SCRUM4 PIANO PROJECT

Scrum Team Members

- Jason Nelson (Scrum Master)
- Mario Montes
- Jacob Jorgensen

Project Goal

 Build a Java-based Web App and a SOAP Web Service to create Piano objects.

Technologies Used

- Java
- JSP / HTML
- XML / SOAP WEB SERVICES
- IBM WEBSPHERE SERVER

TEAM3_PIANO.JAVA - PIANO DATA CLASS

Stores piano details: model, manufacturer, and price.

It's just a simple container for data.

Key Features:

- Three private properties: model, manufacturer, price.
- Two constructors:
 - Empty constructor (sets default values).
 - Full constructor (sets all three fields).
- Getters and Setters for each property.

```
🞵 Team3_Piano.java 🟻 🚺 Team3_Piano_API.java
     package Team3 Scrum4;
     import java.text.NumberFormat;
     public class Team3 Piano {
      private String model;
      private String manufacturer;
      private double price;
 10
      public Team3 Piano()
 12
13
       setModel("");
14
       setManufacturer("");
15
       setPrice(0.0);
16
17
      public Team3_Piano(String model, String manufacturer, double price)
 19
 20
       this.model = model;
 21
       this.manufacturer = manufacturer;
 22
       this.price = price;
 23
```

TEAM3_PIANO_API.JAVA — PIANO WEB SERVICE CLASS

Creates a Piano object when users send input.

Key Features:

- Marked with @WebService: means it can be called by other programs (SOAP).
- Has a createPiano() method
 - Takes model, manufacturer, and price as inputs.
 - Builds and returns a Team3_Piano object.

Each input is labeled with **@WebParam(name="...")** to match what users see.

Keeps everything simple and focused on just creating Piano objects.

```
    ∏ Team3 Piano.java

     package Team3 Scrum4;
   mport javax.jws.WebService;
     @WebService
     public class Team3_Piano_API {
     @WebMethod
      public Team3 Piano createPiano(
12
              @WebParam(name="model")String model,
13
              @WebParam(name="manufacturer")String manufacturer,
              @WebParam(name="price")double price)
14
15
16
       return new Team3 Piano (model, manufacturer, price);
17
18
```

WEB CONTENT: HTML PAGE

- Sets up a web form with inputs for piano model,
 piano manufacturer, and piano price.
- Passes those values to the team3_piano.jsp file to generate a piano object.

WEB CONTENT: JAVA PAGE

- Receives the model, manufacturer, and price from the get request using request.getparameter()
- Creates a new api object
- Creates a new piano object by calling the api's createPiano() method and passing all 3 values into it.
- Then displays the html page using the piano's get methods to populate the list items.

```
Piano Created

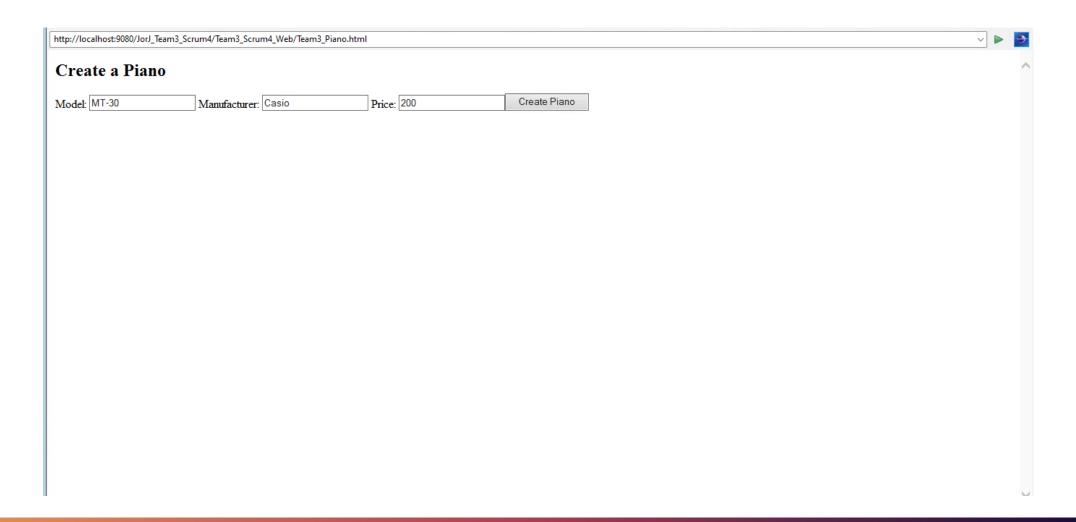
    Model:

    Manufacturer:

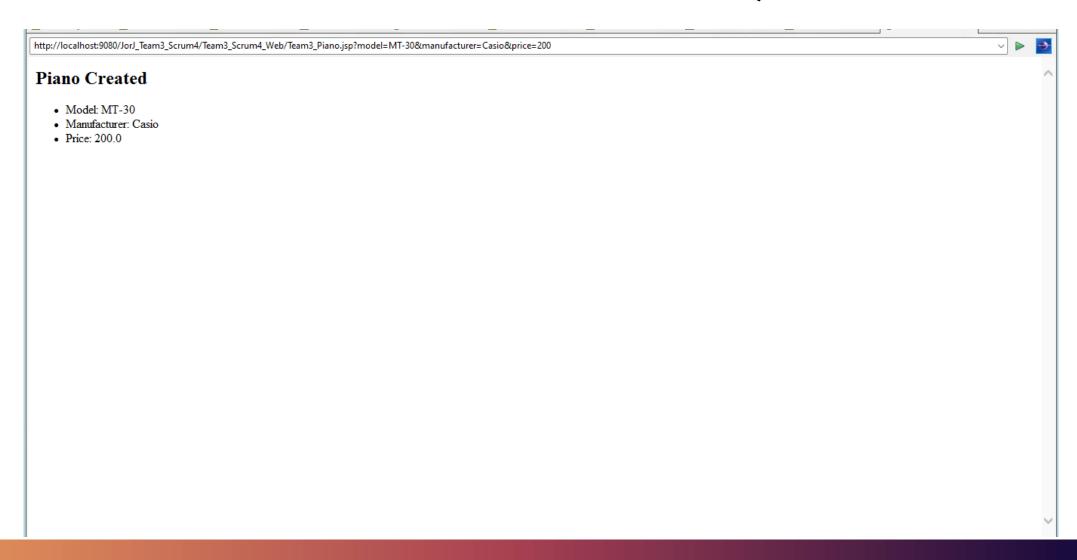
    Price:

 1 <!DOCTYPE HTML><%@page language="java"
        contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
    <%@page import="Team3 Scrum4.*"%>
     String model = request.getParameter("model");
     String manufacturer = request.getParameter("manufacturer");
     double price = Double.parseDouble(request.getParameter("price"));
     Team3 Piano API api = new Team3 Piano API();
     Team3 Piano piano = api.createPiano(model, manufacturer, price);
10
     %>
11@ <html>
13 <title>Team3 Piano - Create Piano</title>
14 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
15 </head>
16@ <body>
        <h2>Piano Created</h2>
           Model: <%= piano.getModel() %>
           Manufacturer: <%= piano.getManufacturer() %>
           Price: <%= piano.getPrice() %>
       23 </body>
24 </html>
```

HTML BEFORE GET REQUEST



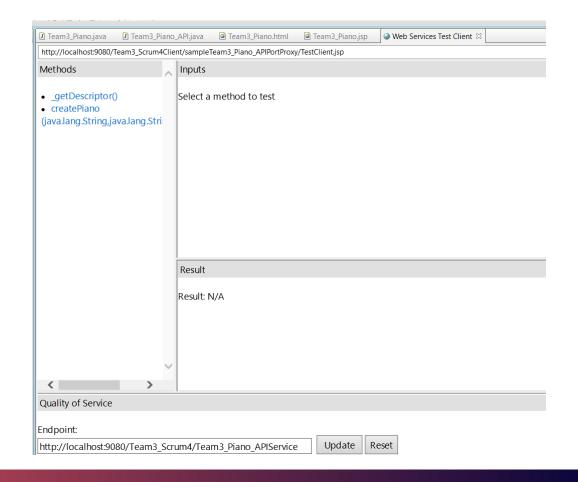
JAVA PAGE AFTER GET REQUEST



AFTER SUCCESSFULLY CREATING SOAP WEB SERVICE TEST CLIENT

 Service operations loaded successfully: createPiano is ready.

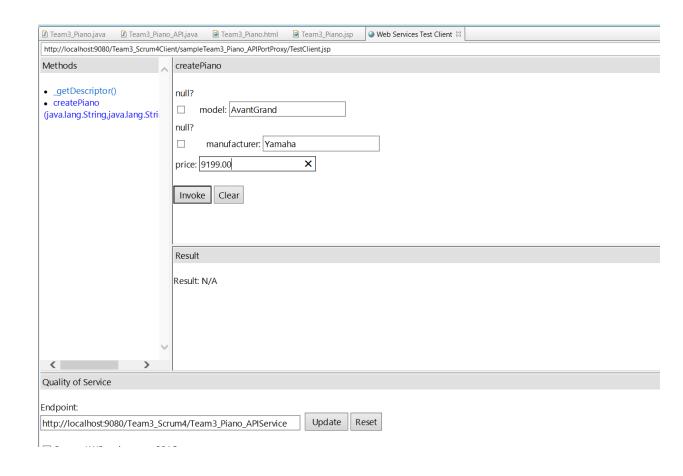
Endpoint URL shows where the service is running.



BEFORE INVOKING CREATEPIANO() METHOD OF API

 Filled out model, manufacturer, and price inputs.

 Getting ready to send request to the createPiano() method.



AFTER INVOKING CREATEPIANO() METHOD OF API

 SOAP request sent successfully to the service.

Service returned a Piano object with correct values.

 Model, Manufacturer, and Price displayed properly in the result.

