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

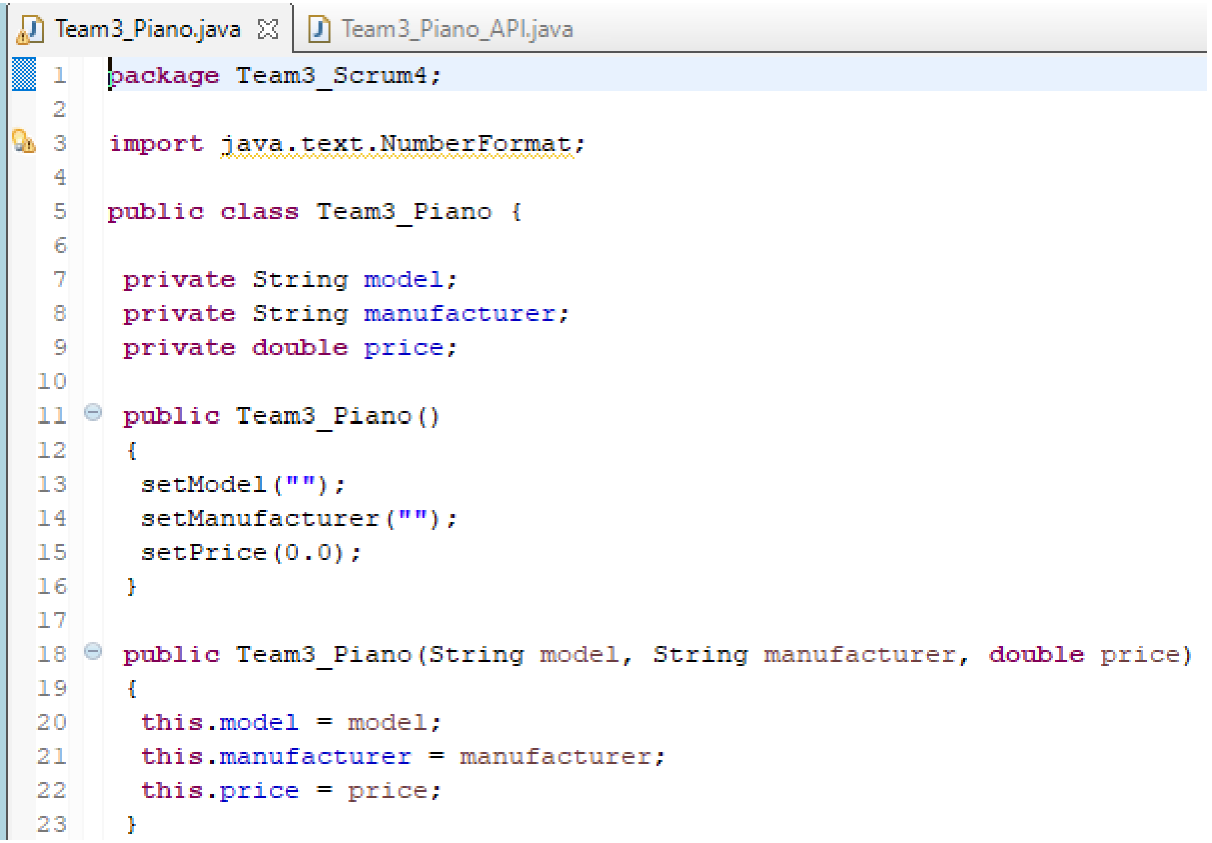
TEAM3_PIANO.JAVA – PIANO DATA CLASS

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.



```java
package Team3_Scrum4;

import java.text.NumberFormat;

public class Team3_Piano {

 private String model;
 private String manufacturer;
 private double price;

 public Team3_Piano()
 {
  setModel("");
  setManufacturer("");
  setPrice(0.0);
 }

 public Team3_Piano(String model, String manufacturer, double price)
 {
  this.model = model;
  this.manufacturer = manufacturer;
  this.price = price;
 }
```

TEAM3_PIANO_API.JAVA – PIANO WEB SERVICE CLASS

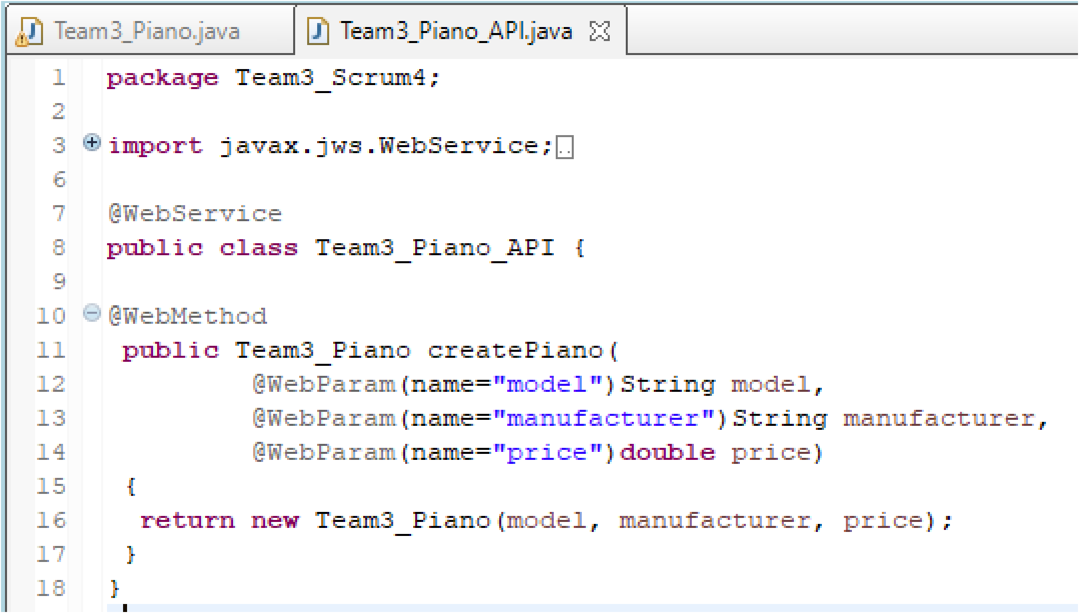
Creates a Piano object when users send input.

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.

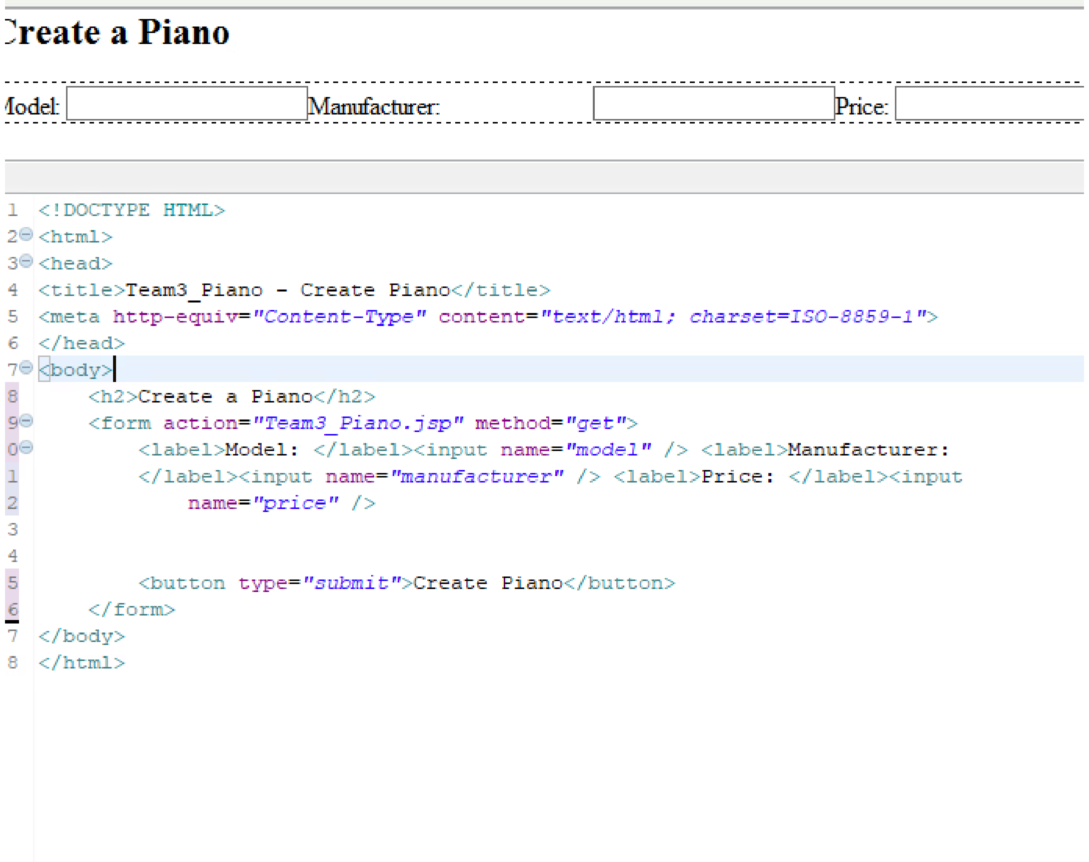
Keeps everything simple and focused on just creating Piano objects.



```java
package Team3_Scrum4;

import javax.jws.WebService;

@WebService
public class Team3_Piano_API {

@WebMethod
 public Team3_Piano createPiano(
          @WebParam(name="model")String model,
          @WebParam(name="manufacturer")String manufacturer,
          @WebParam(name="price")double price)
 {
  return new Team3_Piano(model, manufacturer, price);
 }
}
```

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.



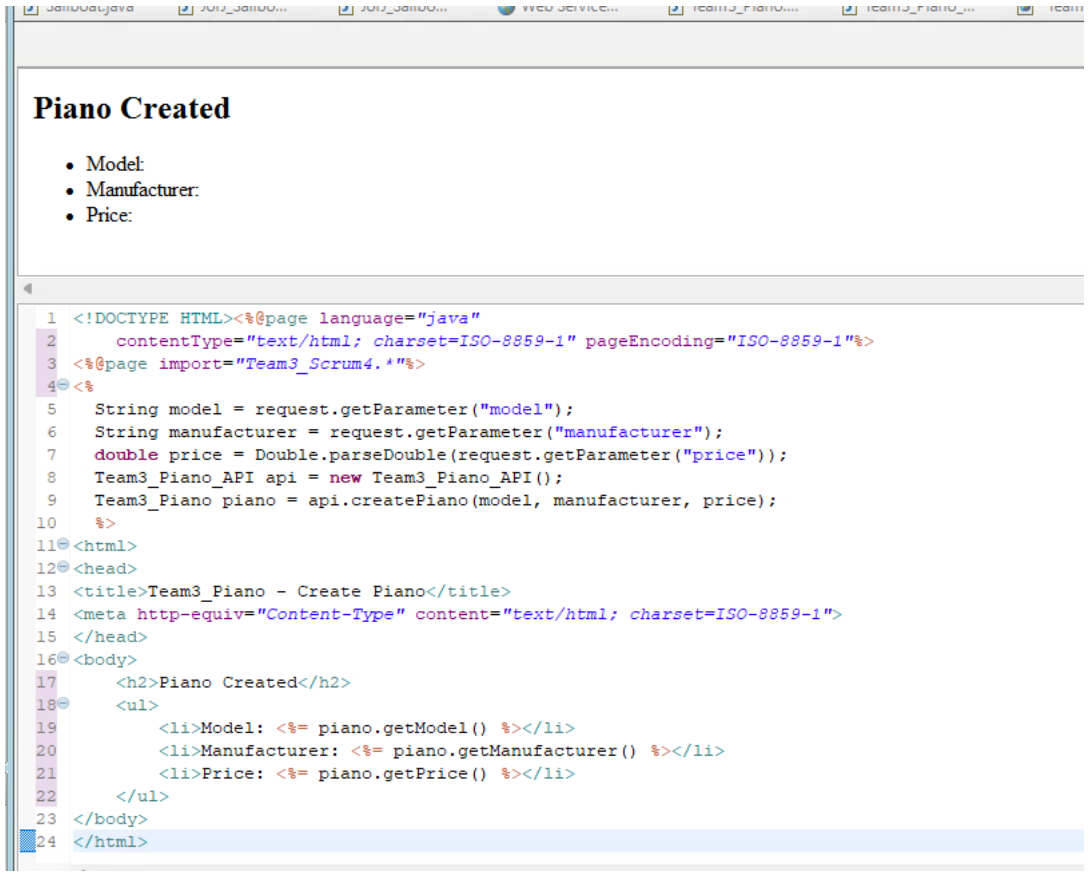
Create a Piano

Model: [ ] Manufacturer: [ ] Price: [ ]

```
<!DOCTYPE HTML>
<html>
<head>
<title>Team3_Piano - Create Piano</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
    <h2>Create a Piano</h2>
    <form action="Team3_Piano.jsp" method="get">
        <label>Model: </label><input name="model" /> <label>Manufacturer:
        </label><input name="manufacturer" /> <label>Price: </label><input
            name="price" />


        <button type="submit">Create Piano</button>
    </form>
</body>
</html>
```

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.

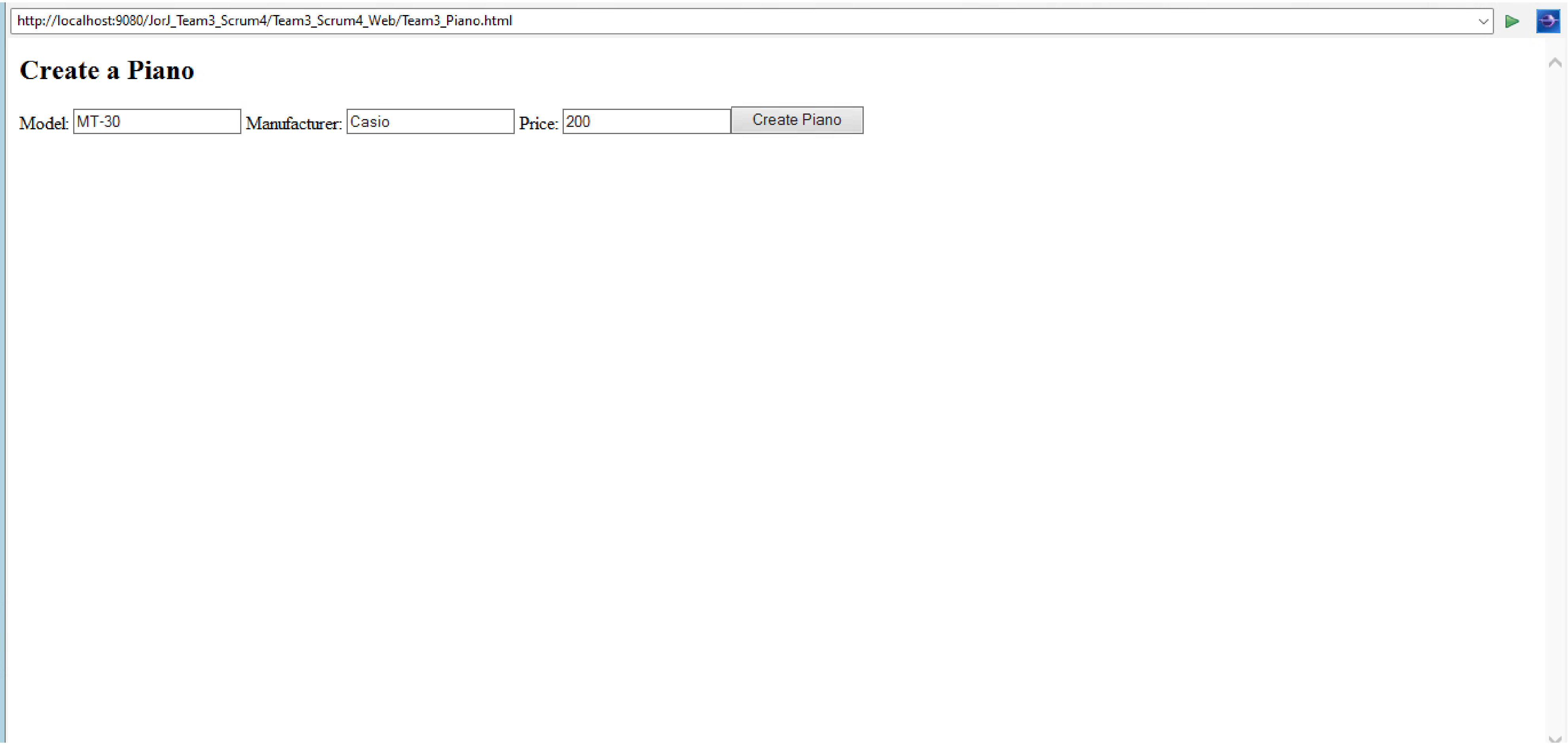


Piano Created

- Model:
- Manufacturer:
- Price:

```
<!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);
  %>
<html>
<head>
<title>Team3_Piano - Create Piano</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
    <h2>Piano Created</h2>
    <ul>
        <li>Model: <%= piano.getModel() %></li>
        <li>Manufacturer: <%= piano.getManufacturer() %></li>
        <li>Price: <%= piano.getPrice() %></li>
    </ul>
</body>
</html>
```

HTML BEFORE GET REQUEST

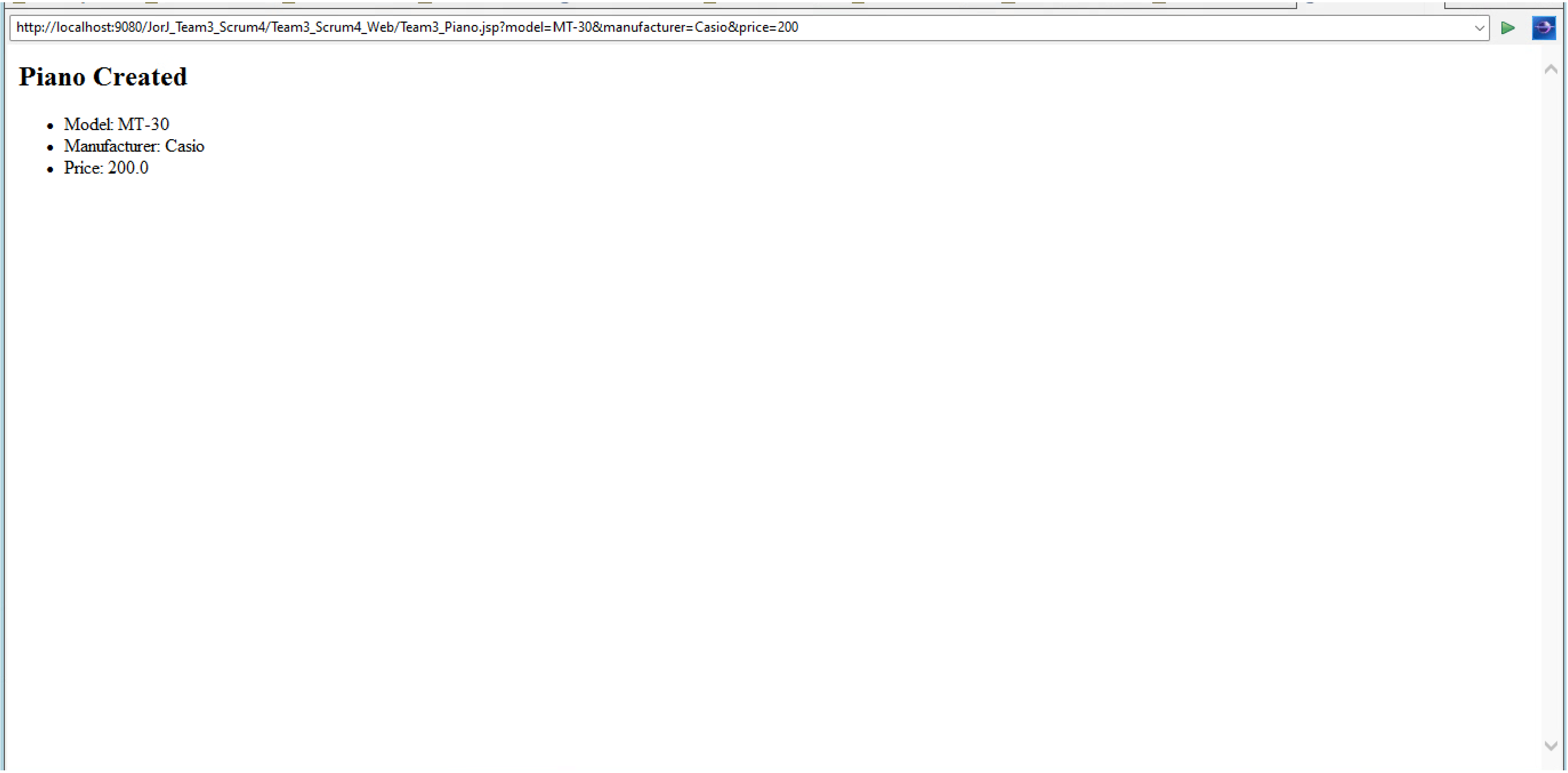


http://localhost:9080/JorJ_Team3_Scrum4/Team3_Scrum4_Web/Team3_Piano.html

Create a Piano

Model: MT-30 Manufacturer: Casio Price: 200

JAVA PAGE AFTER GET REQUEST



http://localhost:9080/JorJ_Team3_Scrum4/Team3_Scrum4_Web/Team3_Piano.jsp?model=MT-30&manufacturer=Casio&price=200

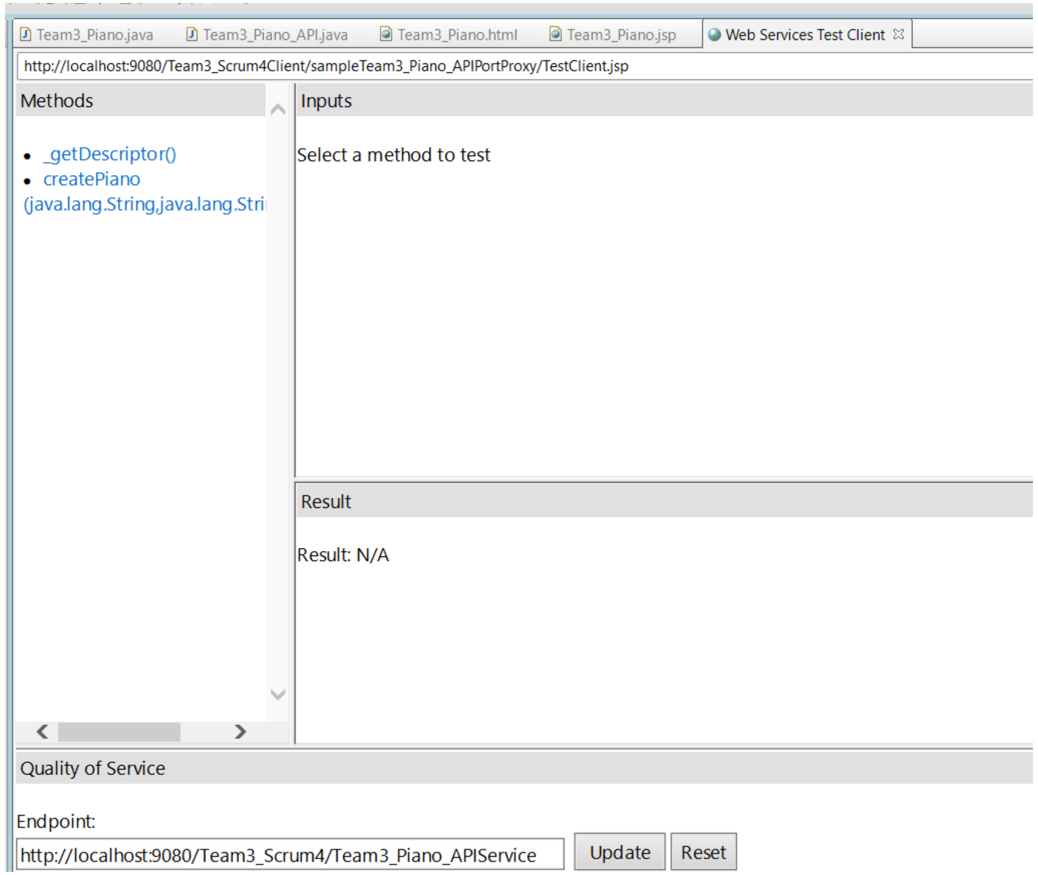
Piano Created

- Model: MT-30
- Manufacturer: Casio
- Price: 200.0

AFTER SUCCESSFULLY CREATING SOAP WEB SERVICE TEST CLIENT

- Service operations loaded successfully: **createPiano** is ready.

- Endpoint URL shows where the service is running.



http://localhost:9080/Team3_Scrum4Client/sampleTeam3_Piano_APIPortProxy/TestClient.jsp

- _getDescriptor()
- createPiano (java.lang.String,java.lang.Stri

Select a method to test

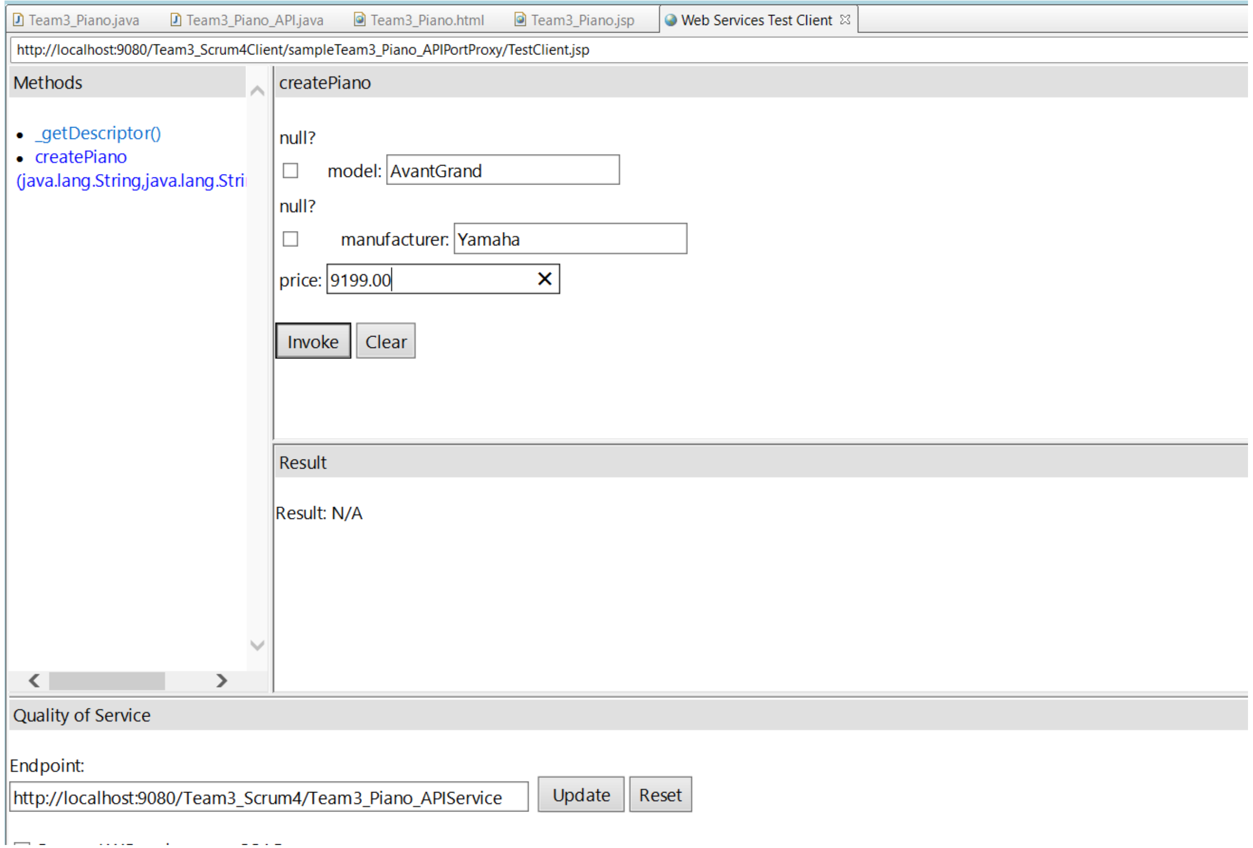
Result: N/A

Endpoint: http://localhost:9080/Team3_Scrum4/Team3_Piano_APIService

BEFORE INVOKING CREATEPIANO() METHOD OF API

- Filled out model, manufacturer, and price inputs.

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



http://localhost:9080/Team3_Scrum4Client/sampleTeam3_Piano_APIPortProxy/TestClient.jsp

- _getDescriptor()
- createPiano (java.lang.String,java.lang.Stri

createPiano

null?

☐ model: AvantGrand

null?

☐ manufacturer: Yamaha

price: 9199.00

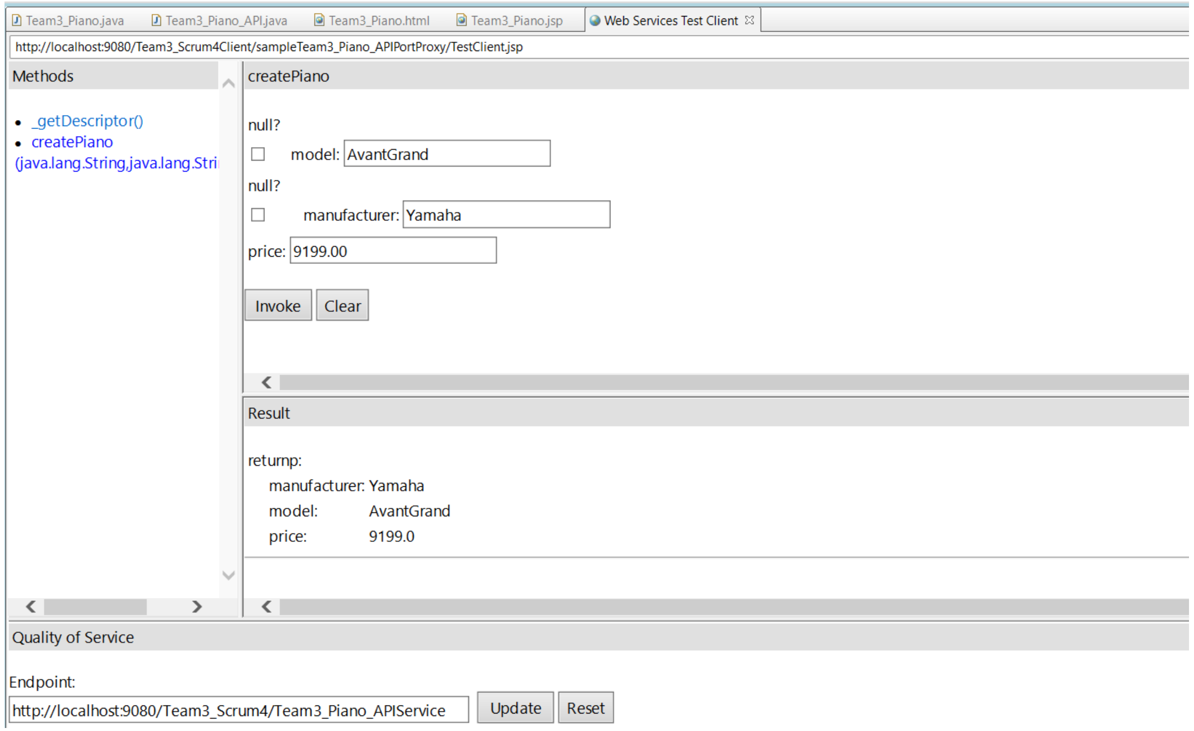
Result: N/A

Endpoint:

http://localhost:9080/Team3_Scrum4/Team3_Piano_APIService

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.



http://localhost:9080/Team3_Scrum4Client/sampleTeam3_Piano_APIPortProxy/TestClient.jsp

createPiano

null?

□ model: AvantGrand

null?

□ manufacturer: Yamaha

price: 9199.00

Result

```
returnp:
    manufacturer: Yamaha
    model:        AvantGrand
    price:        9199.0
```

Quality of Service

Endpoint:

http://localhost:9080/Team3_Scrum4/Team3_Piano_APIService

THANK YOU