

Project1_Unit6

By Yi (Elliot) Cao

Andrew: yc2

Overall understanding of unit6:

As for this unit6, I strictly follow the requirements and give adequate test. The Client is maintained as unit 5, which still can operate as unit5. The Server side can create table, add model, delete model and update model. The processed database tables can be printed in the console to verify the operations.

How to Run My Code:

- 1. Run the DbDriver.java class
- 2. Choose the operation you want to implement:

```
DbDriver [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年3月5日 上午12:22:04)
*** Welcome to the User Interface! ***
*** This is a Car Configuration client. ***
*** To use this program, follow the instructions below. ***

* * * * *
1: create table
2: add model
3: delete model
4: update model
5: exit
```

As for the server side:

Adaptor :

BuildAuto: a class implements all functions of proxyAutomobile, CreateAuto, UpdateAuto, mainly used for hiding all these function from users.

CreateAuto: an interface, used to build auto object and print auto object

FixAuto: an interface, used to fix the exceptions

proxyAutomobile: encapsulate all “CRUD” operations for automobile

UpdateAuto: an interface, used to update the OptionSet and Option

EditThreads: an interface, used to bridge the EditOptions and BuildAuto class

Exception:

AutoException: implements FixAuto used to fix exceptions:

ExceptionNum: to enumerate all exceptions

Helpers: include different fix methods for different exceptions

log: used to record the timestamp of exception and the err message of exception

Model:

Automobile: encapsulate all necessary operations and attributes for car

OptionSet: encapsulate all optionset and options’ operation and attributes

AutoList: encapsulate automobile operations and attributes

Util:

FileIO: used to build auto object and serialization and deserialization

Scale:

EditOptions: implement multithreads operations

OptinNum: to enumerate all edit options

Server:

AutoServer: the interface includes all responses for client request operations

BuildCarModelOptions: implements the AutoServer

DefaultSocketServer: access the client side requests

Server: start the server side;

Database:

JDBCAdapter: a class used to connect to the db, open the db and close the db.

printTable: print the table contents from the database to the console

CreateTable: create tables if not exist

Addmodel: add model to the database

DeleteModel: delete the model

UpdateModel: update the model name , base price, optionset name, option name and option price

Updaters: an interface include the operation to update the model name, base price, optionset name, option name or option price.

As for the client side:

Adaptor :

BuildAuto: a class implements all functions of proxyAutomobile, CreateAuto, UpdateAuto, mainly used for hiding all these function from users.

CreateAuto: an interface, used to build auto object and print auto object

FixAuto: an interface, used to fix the exceptions

proxyAutomobile: encapsulate all “CRUD” operations for automobile

UpdateAuto: an interface, used to update the OptionSet and Option

EditThreads: an interface, used to bridge the EditOptions and BuildAuto class

Exception:

AutoException: implements FixAuto used to fix exceptions:

ExceptionNum: to enumerate all exceptions

Helpers: include different fix methods for different exceptions

log: used to record the timestamp of exception and the err message of exception

Model:

Automobile: encapsulate all necessary operations and attributes for car

OptionSet: encapsulate all optionset and options’ operation and attributes

AutoList: encapsulate automobile operations and attributes

Util:

FileIO: used to build auto object and serialization and deserialization

Scale:

EditOptions: implement multithreads operations

OptinNum: to enumerate all edit options

Client:

CarModelOptionsIO: bridge the communication to the server side

DefaultSocketClient: access the server side requests

Client: start the client side;

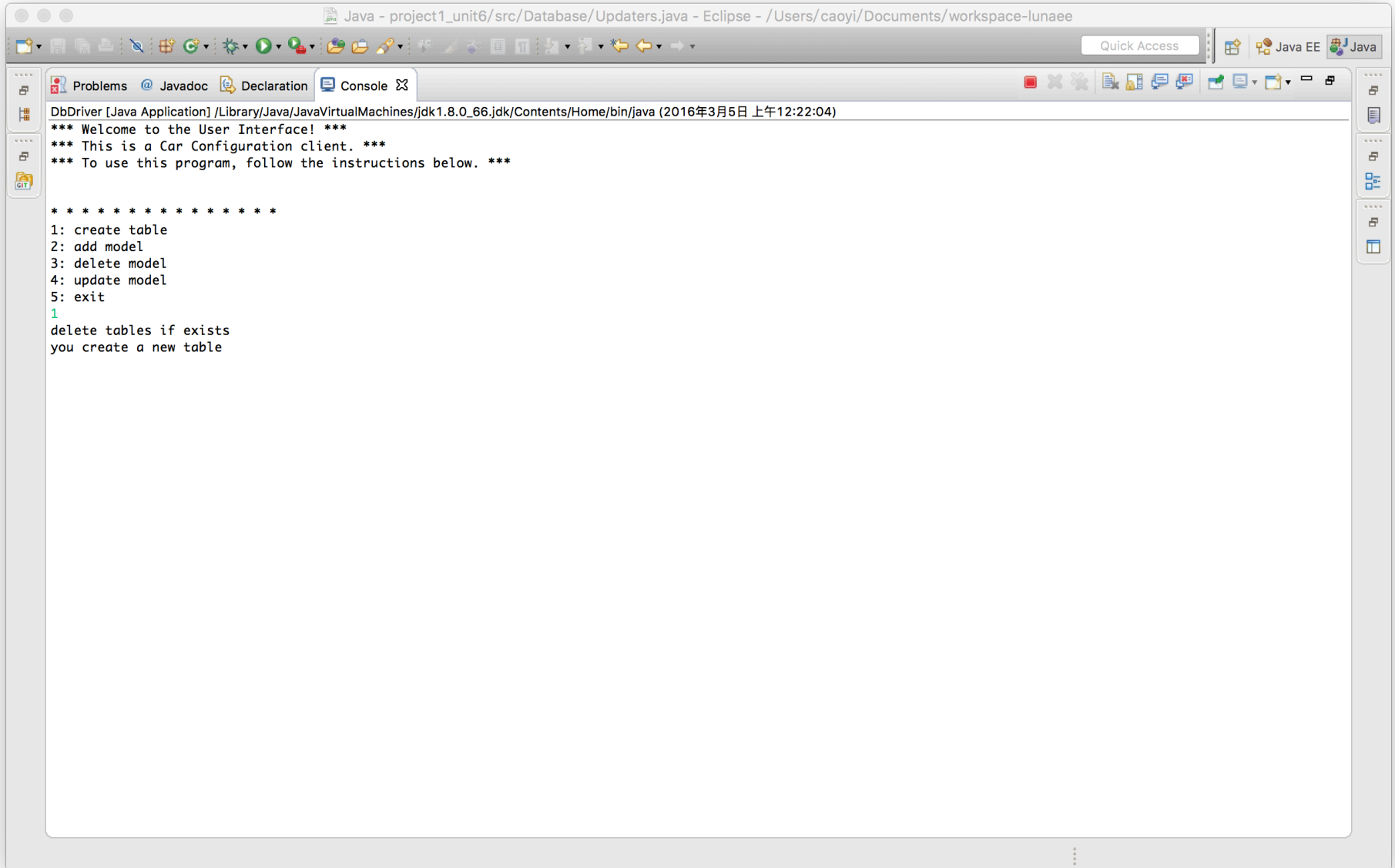
Web:

MyServlet: the abstract class includes doGet and doPost functions

getModel: list all the available models

displayModel: display the options

Create the Table:



```
Java - project1_unit6/src/Database/Updaters.java - Eclipse - /Users/caoyi/Documents/workspace-lunaee

DbDriver [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年3月5日 上午12:22:04)
*** Welcome to the User Interface! ***
*** This is a Car Configuration client. ***
*** To use this program, follow the instructions below. ***

* * * * *
1: create table
2: add model
3: delete model
4: update model
5: exit
1
delete tables if exists
you create a new table
```

Add model: BMW.txt

```
Java - project1_unit6/src/Database/Updaters.java - Eclipse - /Users/caoyi/Documents/workspace-lunaee

DbDriver [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年3月5日 上午12:22:04)
1: create table
2: add model
3: delete model
4: update model
5: exit
1
delete tables if exists
you create a new table
2
please input the filename you want to insert into table, you can add txt file or properties file( input -1 to finish the add model process)
BMW.txt
*****here is the table Automobile*****
id      model      base_price
1       BMW        18455.000000
*****here is the table OptionSet*****
id      name      auto_id
1       Color      1
2       Transmission  1
3       BrakesTractionControl  1
4       SideImpactAirbags  1
5       PowerMoonroof  1
*****here is the table Options*****
id      name      price      opset_id
1       Fort Knox Gold Clearcoat Metallic  0.0      1
2       Liquid Grey Clearcoat Metallic  0.0      1
3       Infra-Red Clearcoat  0.0      1
4       Grabber Green Clearcoat Metallic  0.0      1
5       Sangria Red Clearcoat Metallic  0.0      1
6       French Blue Clearcoat Metallic  0.0      1
7       Twilight Blue Clearcoat Metallic  0.0      1
8       CD Silver Clearcoat Metallic  0.0      1
9       Pitch Black Clearcoat  0.0      1
10      Cloud 9 White Clearcoat  0.0      1
11      Automatic  0.0      2
12      Standard  -815.0    2
13      Standard  0.0      3
14      ABS        400.0    3
15      ABS with Advance Trac  1625.0    3
16      None       0.0      4
17      Selected  350.0    4
18      None       0.0      5
19      Selected  595.0    5
please input the filename you want to insert into table, you can add txt file or properties file( input -1 to finish the add model process)
```

Add model: Focus_Wagon_ZTW.properties

Package Explorer

- ExceptionNum.java
- Helpers.java
- log.java
- Model
 - AutoList.java
 - Automobile.java
 - OptionSet.java
- scale
 - EditOptions.java
 - OptionNum.java
- server
 - AutoServer.java
 - BuildCarModelOptions.java
 - DefaultSocketServer.java
 - Server.java
- Util
 - FileIO.java
- JRE System Library [JavaSE-1.8]
- Audi.txt
- BMW.properties
- BMW.txt
- CreateTable.txt
- dbInit.txt
- Focus_Wagon_ZTW.properties
- Focus_Wagon_ZTW.txt
- log.txt
- PurchaseOrderService
- purchaseService
- Servers
- test
- WebServiceProject
- XML_Parser
- XQuery_Test
- XQuey_demo
- Yi_Cao_Lab4
- Yi_Cao_Lab7
- Yi_Cao_Lab7_

Problems

Javadoc

Declaration

Console

DbDriver [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年3月5日 上午12:33:23)
please input the filename you want to insert into table, you can add txt file or properties file(input -1 to finish the add model)
Focus_Wagon_ZTW.properties
*****here is the table Automobile*****

id	model	base_price
1	BMW	18455.000000
2	Focus Wagon ZTW	18445.000000

*****here is the table OptionSet*****

id	name	auto_id
1	Color	1
2	Transmission	1
3	BrakesTractionControl	1
4	SideImpactAirbags	1
5	PowerMoonroof	1
6	Color	2
7	Transmission	2
8	BrakesTractionControl	2
9	SideImpactAirbags	2
10	PowerMoonroof	2

*****here is the table Options*****

id	name	price	opset_id
1	Fort Knox Gold Clearcoat Metallic	0.0	1
2	Liquid Grey Clearcoat Metallic	0.0	1
3	Infra-Red Clearcoat	0.0	1
4	Grabber Green Clearcoat Metallic	0.0	1
5	Sangria Red Clearcoat Metallic	0.0	1
6	French Blue Clearcoat Metallic	0.0	1
7	Twilight Blue Clearcoat Metallic	0.0	1
8	CD Silver Clearcoat Metallic	0.0	1
9	Pitch Black Clearcoat	0.0	1
10	Cloud 9 White Clearcoat	0.0	1
11	Automatic	0.0	2
12	Standard	-815.0	2
13	Standard	0.0	3
14	ABS	400.0	3
15	ABS with Advance Trac	1625.0	3
16	None	0.0	4
17	Selected	350.0	4
18	None	0.0	5
19	Selected	595.0	5
20	Fort Knox Gold Clearcoat Metallic	0.0	6
21	Liquid Grey Clearcoat Metallic	0.0	6
22	Infra-Red Clearcoat	0.0	6
23	Grabber Green Clearcoat Metallic	0.0	6

Delete model: Focus_Wagon_ZTW.properties

```
3
*****delete the model in the database*****
here is the available models in the db
*****here is the table Automobile*****
id      model                base_price
1       BMW                  18455.000000
2       Focus Wagon ZTW      18445.000000
please input the model id you want to delete
2
you have successfully delete the model:Focus Wagon ZTW
++++++here is the new Automobile table after deleteFocus Wagon ZTW:+++++
*****here is the table Automobile*****
id      model                base_price
1       BMW                  18455.000000
```


Update model: BMW

```
+++++++update the model name from BMW to Ford+++++++
*****here is the table Automobile*****
id      model      base_price
1      Ford      18455.000000
+++++++update the baseprice+++++++
*****here is the table Automobile*****
id      model      base_price
1      Ford      1111.000000
+++++++update the OptionSet name from Color to ColorSet+++++++
*****here is the table OptionSet*****
id      name      auto_id
1      ColorSet      1
2      Transmission      1
3      BrakesTractionControl      1
4      SideImpactAirbags      1
5      PowerMoonroof      1
+++++++update the Option name from Automatic to Designed+++++++
*****here is the table Options*****
id      name      price      opset_id
1      Fort Knox Gold Clearcoat Metallic      0.0      1
2      Liquid Grey Clearcoat Metallic      0.0      1
3      Infra-Red Clearcoat      0.0      1
4      Grabber Green Clearcoat Metallic      0.0      1
5      Sangria Red Clearcoat Metallic      0.0      1
6      French Blue Clearcoat Metallic      0.0      1
7      Twilight Blue Clearcoat Metallic      0.0      1
8      CD Silver Clearcoat Metallic      0.0      1
9      Pitch Black Clearcoat      0.0      1
10     Cloud 9 White Clearcoat      0.0      1
11     Designed      0.0      2
12     Standard      -815.0      2
13     Standard      0.0      3
14     ABS      400.0      3
15     ABS with Advance Trac      1625.0      3
16     None      0.0      4
17     Selected      350.0      4
18     None      0.0      5
19     Selected      595.0      5
+++++++update the Option price +++++++
```

Update model: BMW and exist

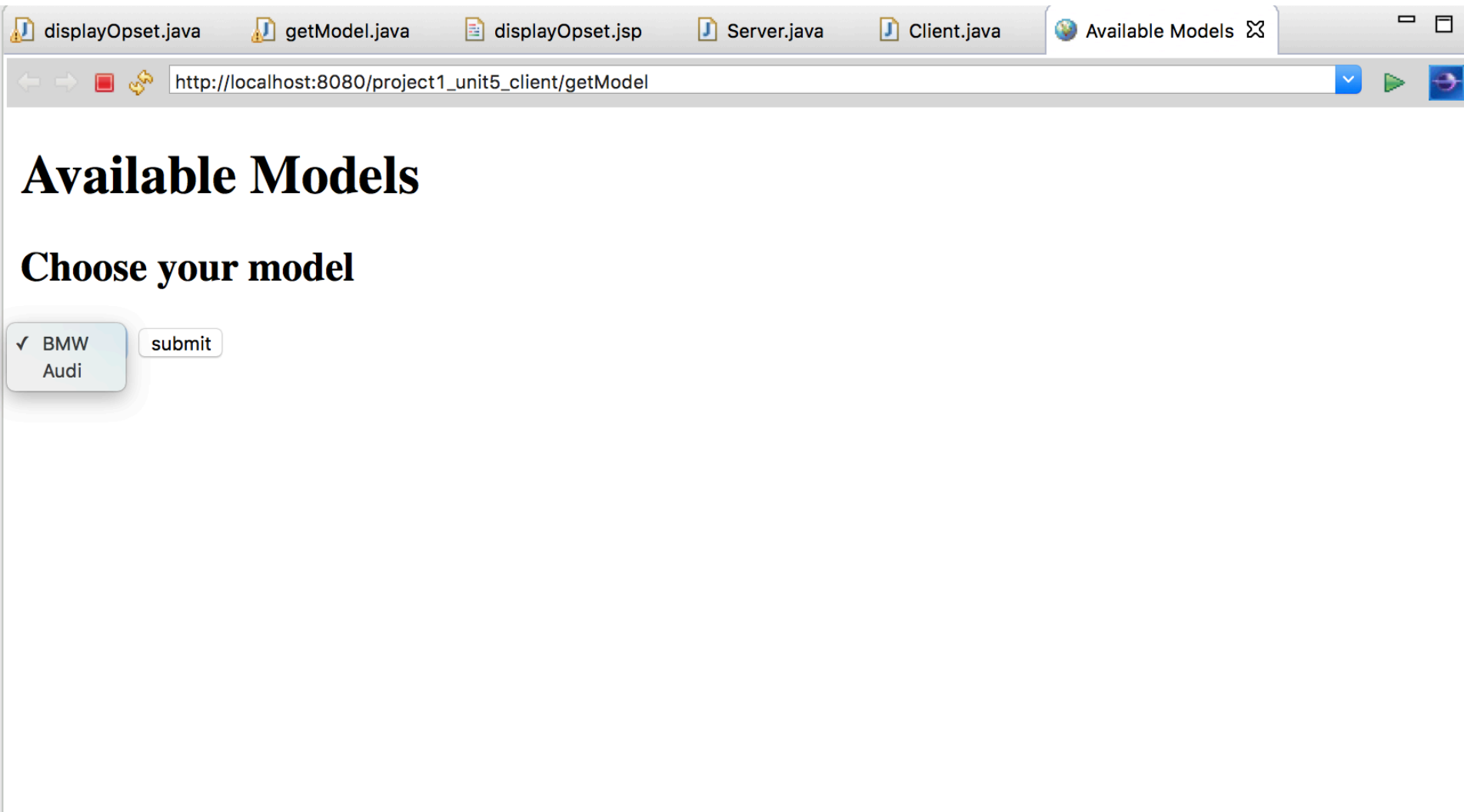
+++++++update the Option price ++++++
*****here is the table Options*****

id	name	price	opset_id
1	Fort Knox Gold Clearcoat Metallic	0.0	1
2	Liquid Grey Clearcoat Metallic	0.0	1
3	Infra-Red Clearcoat	0.0	1
4	Grabber Green Clearcoat Metallic	0.0	1
5	Sangria Red Clearcoat Metallic	0.0	1
6	French Blue Clearcoat Metallic	0.0	1
7	Twilight Blue Clearcoat Metallic	0.0	1
8	CD Silver Clearcoat Metallic	0.0	1
9	Pitch Black Clearcoat	0.0	1
10	Cloud 9 White Clearcoat	0.0	1
11	Designed	1000.0	2
12	Standard	-815.0	2
13	Standard	0.0	3
14	ABS	400.0	3
15	ABS with Advance Trac	1625.0	3
16	None	0.0	4
17	Selected	350.0	4
18	None	0.0	5
19	Selected	595.0	5

5
you exist the db successfully!

The Client side is maintained as unit 5 and tests are listed as the followings

List available models:



The screenshot shows a web browser window with several tabs open: `displayOpset.java`, `getModel.java`, `displayOpset.jsp`, `Server.java`, `Client.java`, and `Available Models`. The address bar shows the URL `http://localhost:8080/project1_unit5_client/getModel`. The main content area displays the heading **Available Models** and the instruction **Choose your model**. Below this, there is a light blue button with a checkmark icon and the text `BMW` and `Audi` stacked vertically. To the right of this button is a `submit` button.

displayOpset.java getModel.java displayOpset.jsp Server.java Client.java Available Models

http://localhost:8080/project1_unit5_client/getModel

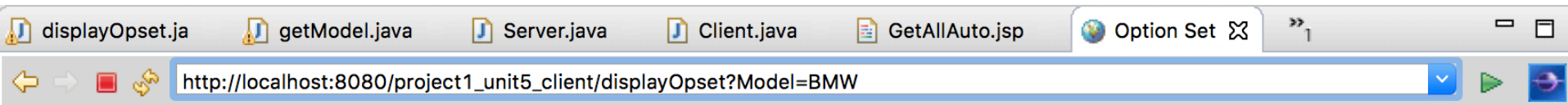
Available Models

Choose your model

✓ BMW
Audi

submit

List the option set:



Basic Car Choice

Make/Model:	<input type="text" value="BMW"/>
Color:	<input type="text" value="Fort Knox Gold Clearcoat Metallic"/>
Transmission:	<input type="text" value="Automatic"/>
Brakes/TractionControl:	<input type="text" value="Standard"/>
Side Impact Air Bags	<input type="text" value="None"/>
Power Moonroof	<input type="text" value="None"/>

DONE

List the options and total price

Server.java

Client.java

GetAllAuto.jsp

displayOpset.js

Apache Tomcat/6

Available Model

2

← → ⏏ ⏮ ⏭

http://localhost:8080/project1_unit5_client/selectedChoices

⏏ ⏭

Here is what you selected:

BMW	base price	18455.0
Color	Fort Knox Gold Clearcoat Metallic	0.0
Transmission	Automatic	0.0
BrakesTractionControl	Standard	0.0
SideImpactAirbags	None	0.0
PowerMoonroof	None	0.0
Total Cost		18455.0