Project1_Unit5

By Yi (Elliot) Cao

Andrew: yc2

Overall understanding of unit5:
As for this unit5, I strictly follow the requirements and give adequate test.   The client can Exit, Upload, Get and Select. Beside, the getModel. java , displayOpset. java extends abstract class MyServlet.java so that operates as servlets. The results were returned to the web interface with JSP.
The Server side can access multiple clients' request and operated as requested. As for the test, we can open the server first and start several clients to take different different operations. We have 6 packages for client and server.

How to Run My Code:

1. Start the server.java in the package project1_unit5_server
2. Start the client.java in the package project1_unit5_client
3. Choose upload operation. Eg. Upload BMW.txt , Audi.txt
4. Start getModel.java in the package project1_unit5_clientside
5. Choose the model in the available models under the url:
   http://localhost:8080/project1_unit5_client/getModel
6. Choose the options and then it would show the details and total price of the option

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;

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;

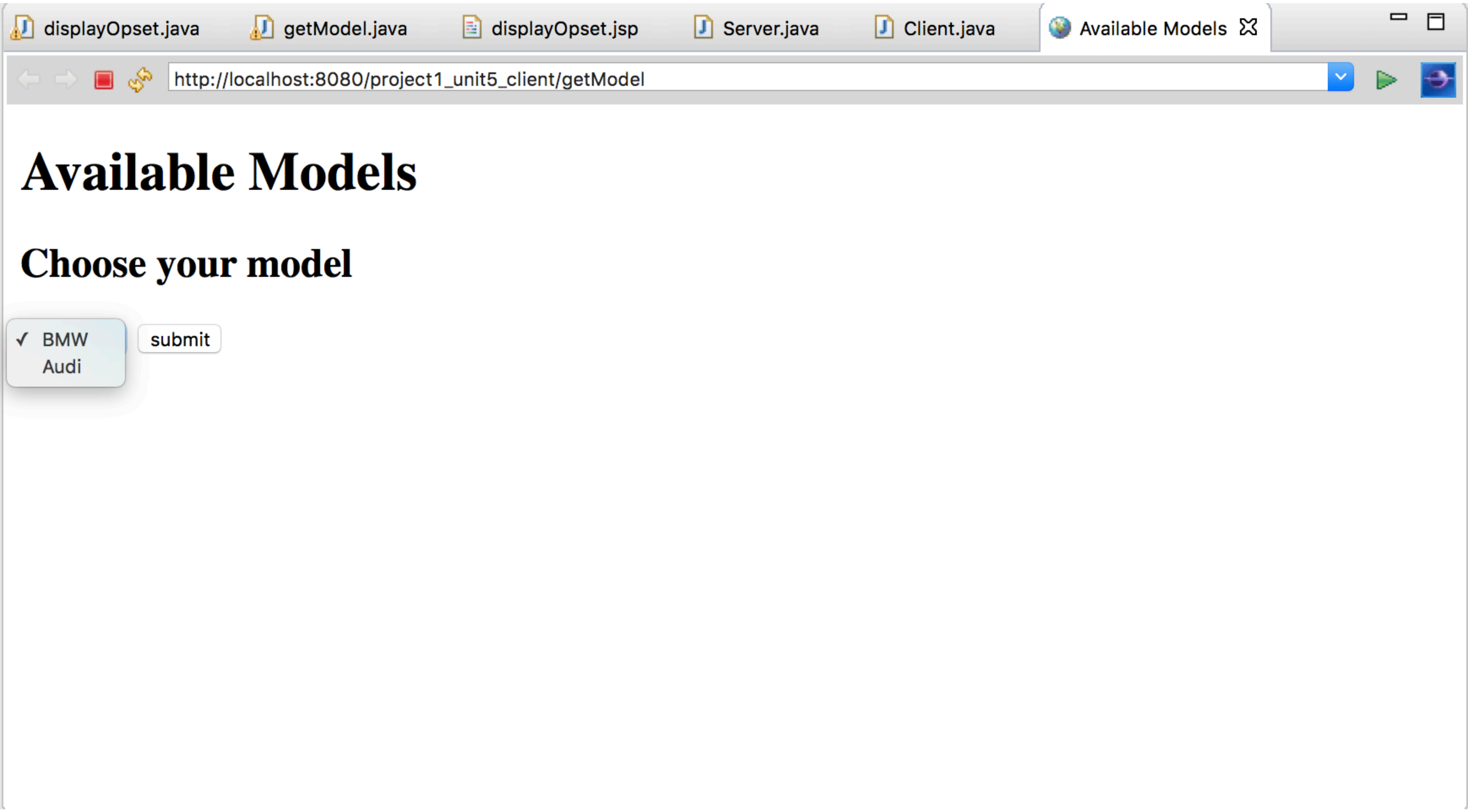
Web:

MyServlet: the abstract class includes doGet and doPost functions

getModel: list all the available models

displayModel: display the options

List available models:



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

http://localhost:8080/project1_unit5_client/getModel

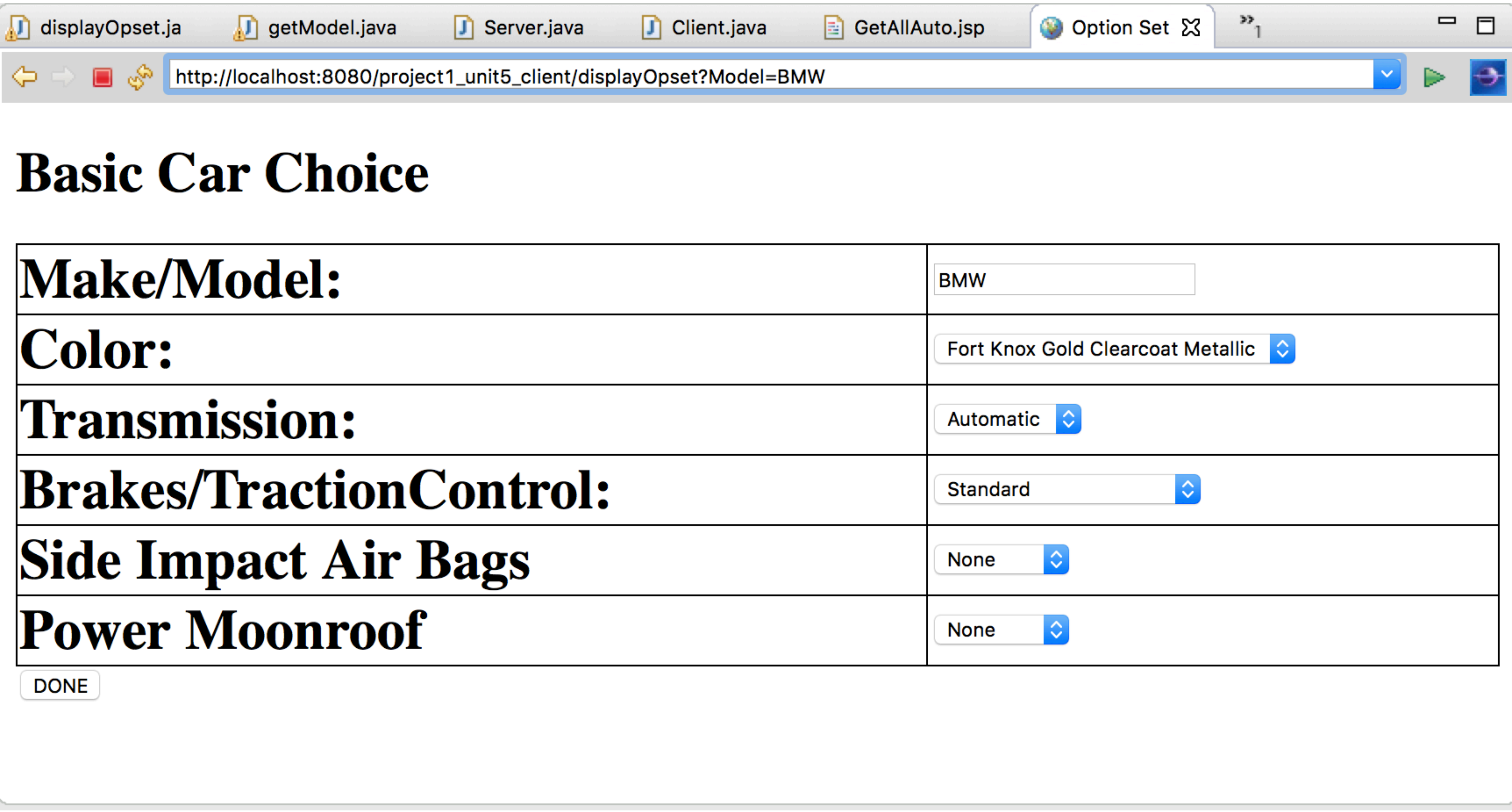
Available Models

Choose your model

✓ BMW
Audi

submit

List the option set:



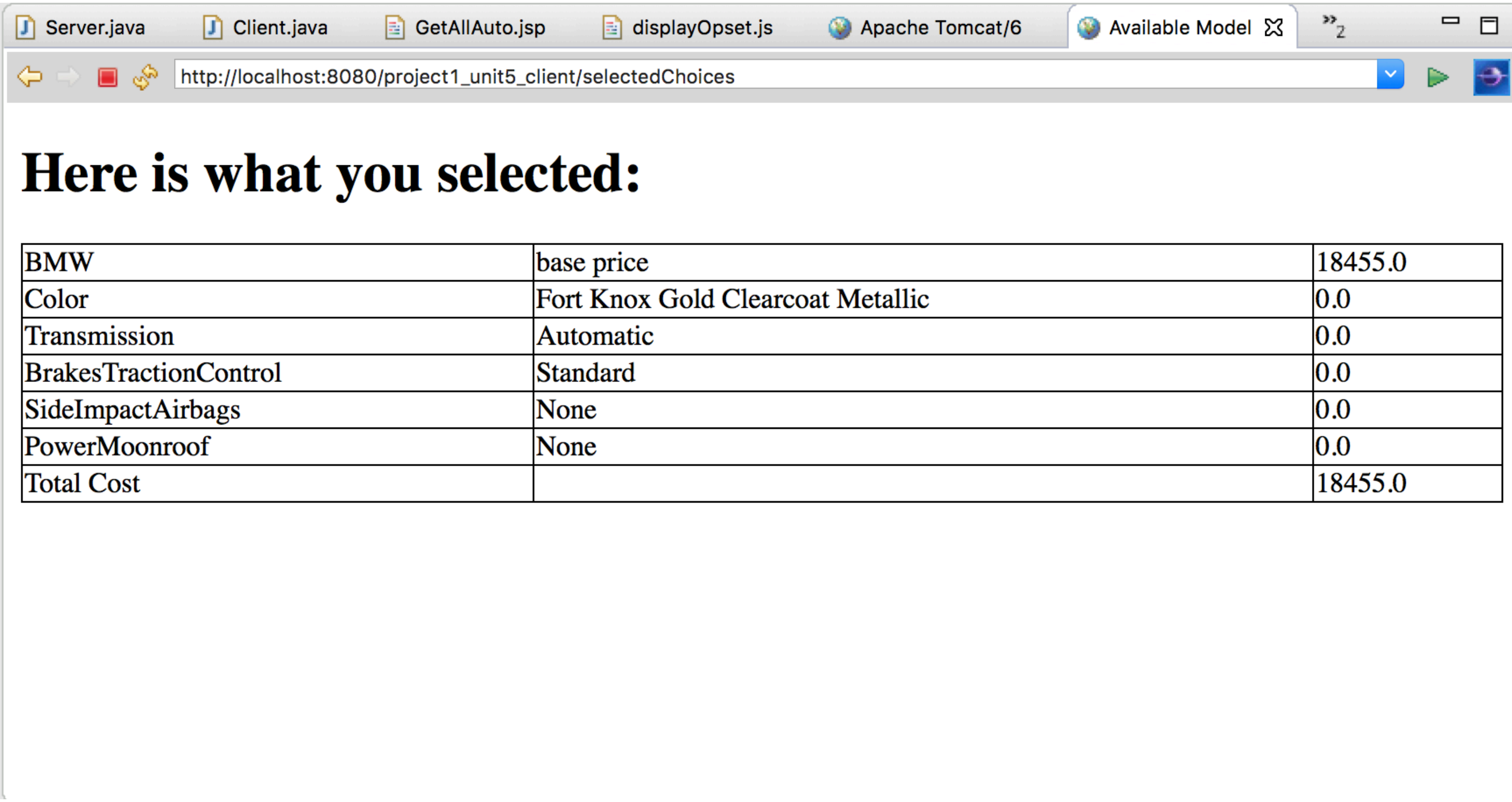
http://localhost:8080/project1_unit5_client/displayOpset?Model=BMW

Basic Car Choice

Make/Model:	BMW
Color:	Fort Knox Gold Clearcoat Metallic
Transmission:	Automatic
Brakes/TractionControl:	Standard
Side Impact Air Bags	None
Power Moonroof	None

DONE

List the options and total price



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