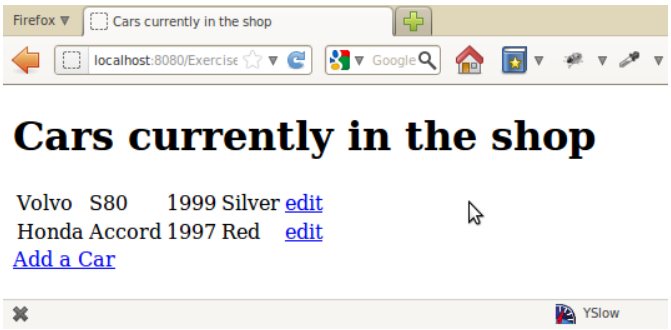
Lab 3

The Setup:

In this exercise we will create a simple CRUD (Create, Retrieve, Update, Delete) application with Spring MVC. Start by opening **exercise8.1** and add the following dependencies to the project's pom.xml:

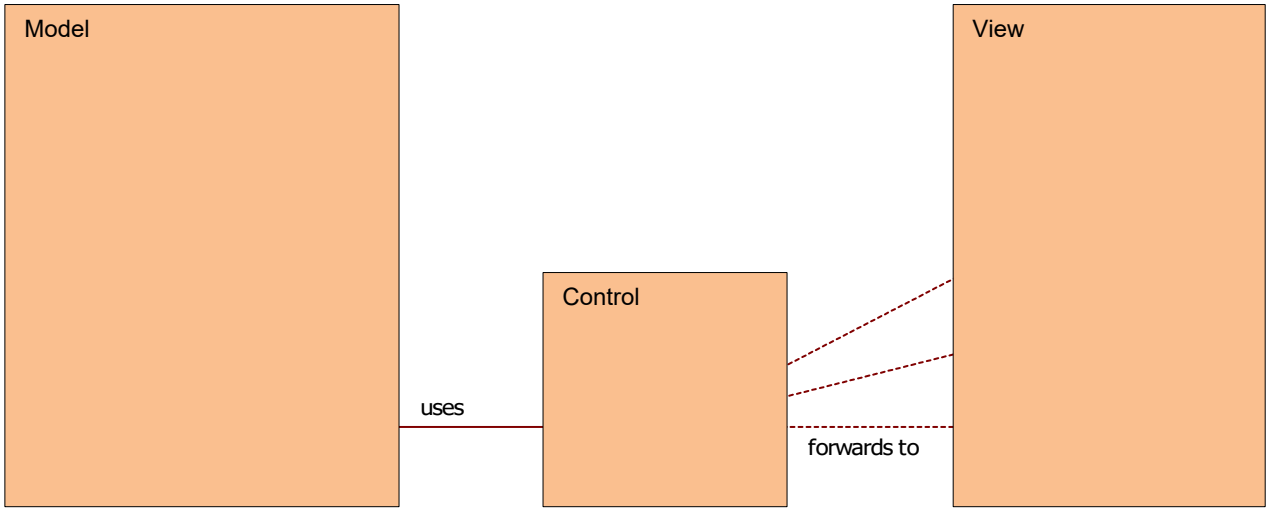
```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>4.3.8.RELEASE</version>
 </dependency>
 <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
 </dependency>
<dependency>
   <groupId>jstl</groupId>
   <artifactId>jstl</artifactId>
   <version>1.2</version>
</dependency>
```

Once everything is setup running the project on the tomcat server should open the following page in your browser:



The Application:

The provided code is reasonably simple, **CarController** uses **CarDao** to create, retrieve, update and delete **Car** objects, after which it forwards to one of the views.



The Exercise:

The goal of this exercise is to make a Book store CRUD application similar to the Car Shop curd application. We've provided the basic **Book** and **BookDao** classes along with the exercise skeleton code.

The core items that you will need to make are the **BookController** class, and a new set of views related to the book store e.g. BookList, BookDetail, AddBook.

You can either copy paste many of the files from the car shop application and update them, or for a greater (although potentially somewhat more frustrating) learning experience you can start from scratch.

Note: Please be aware that **CarController.java** has a mapping for "/", if you directly copy paste CarController.java to **BookController.java** you will end up with a "/" mapping in BookController.java as well. Spring does not allow having the same path mapped to two different handlers. In other words, remember to remove one of the two, they cannot exist simultaneously.