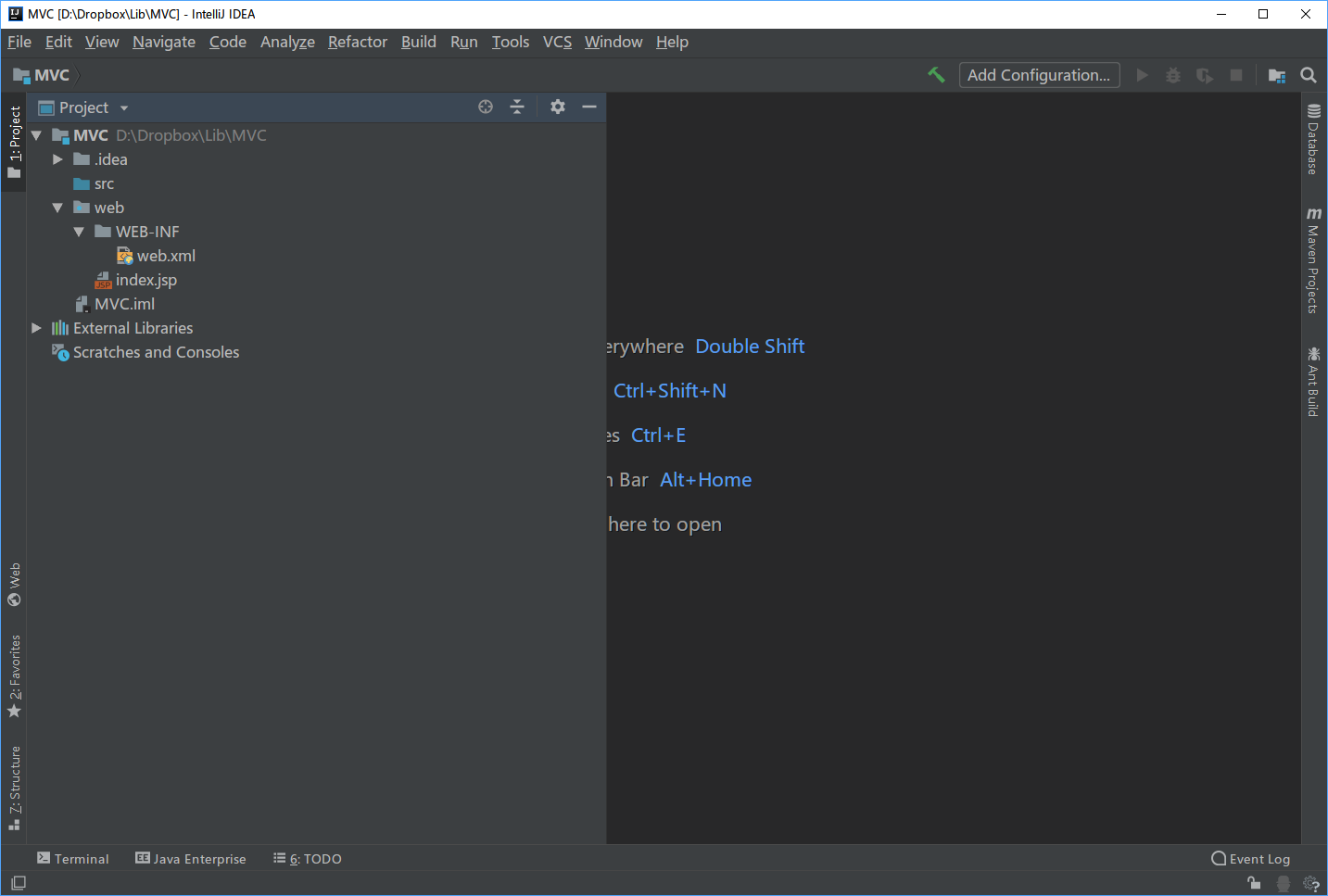
**实验七 JSP+JavaBean+Servlet开发**

### 一、实验目的

基于MVC模式，使用JSP+JavaBean+Servlet+JDBC技术完成之前实验的功能呢。

### 二、实验项目内容

* **新建web项目：**

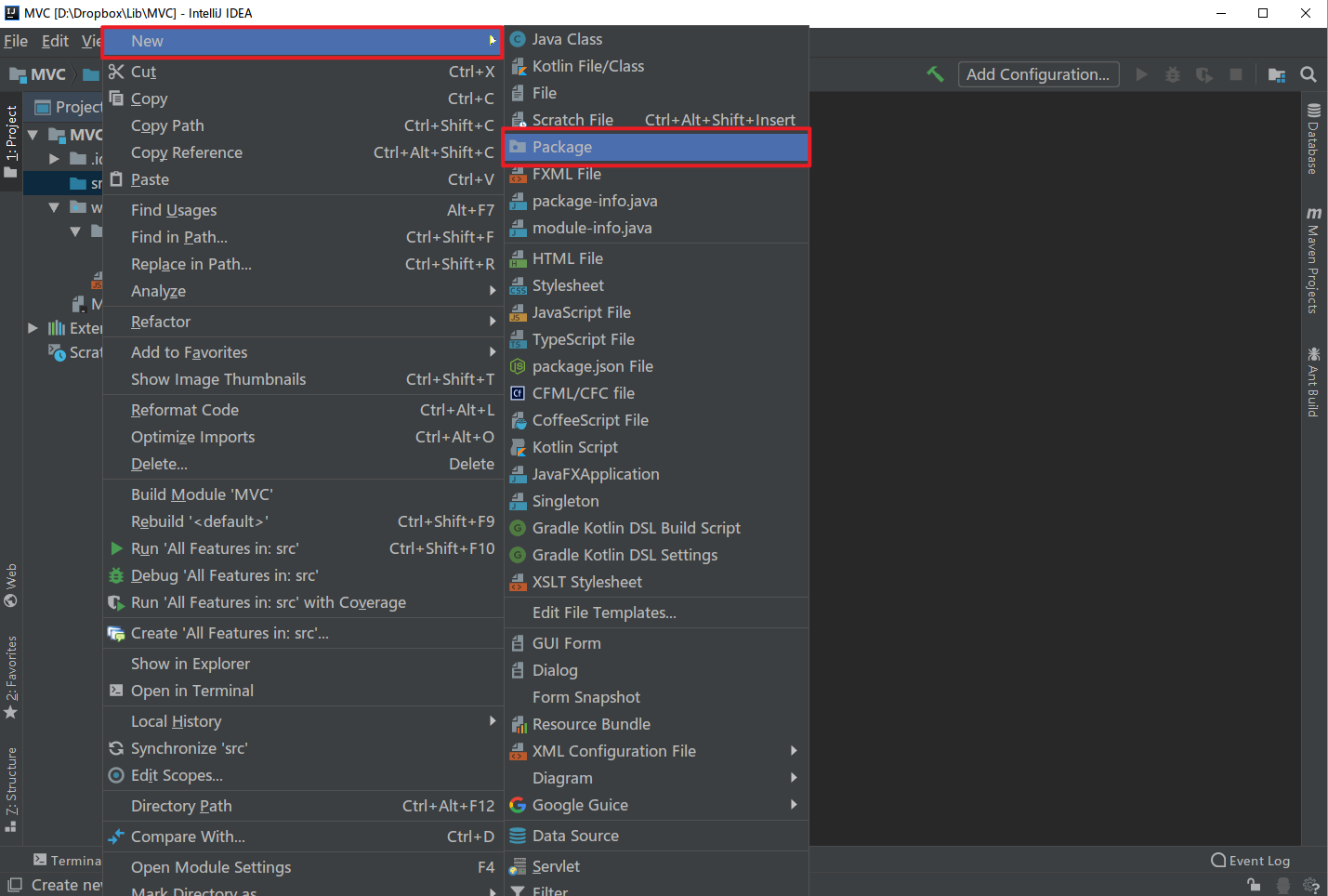


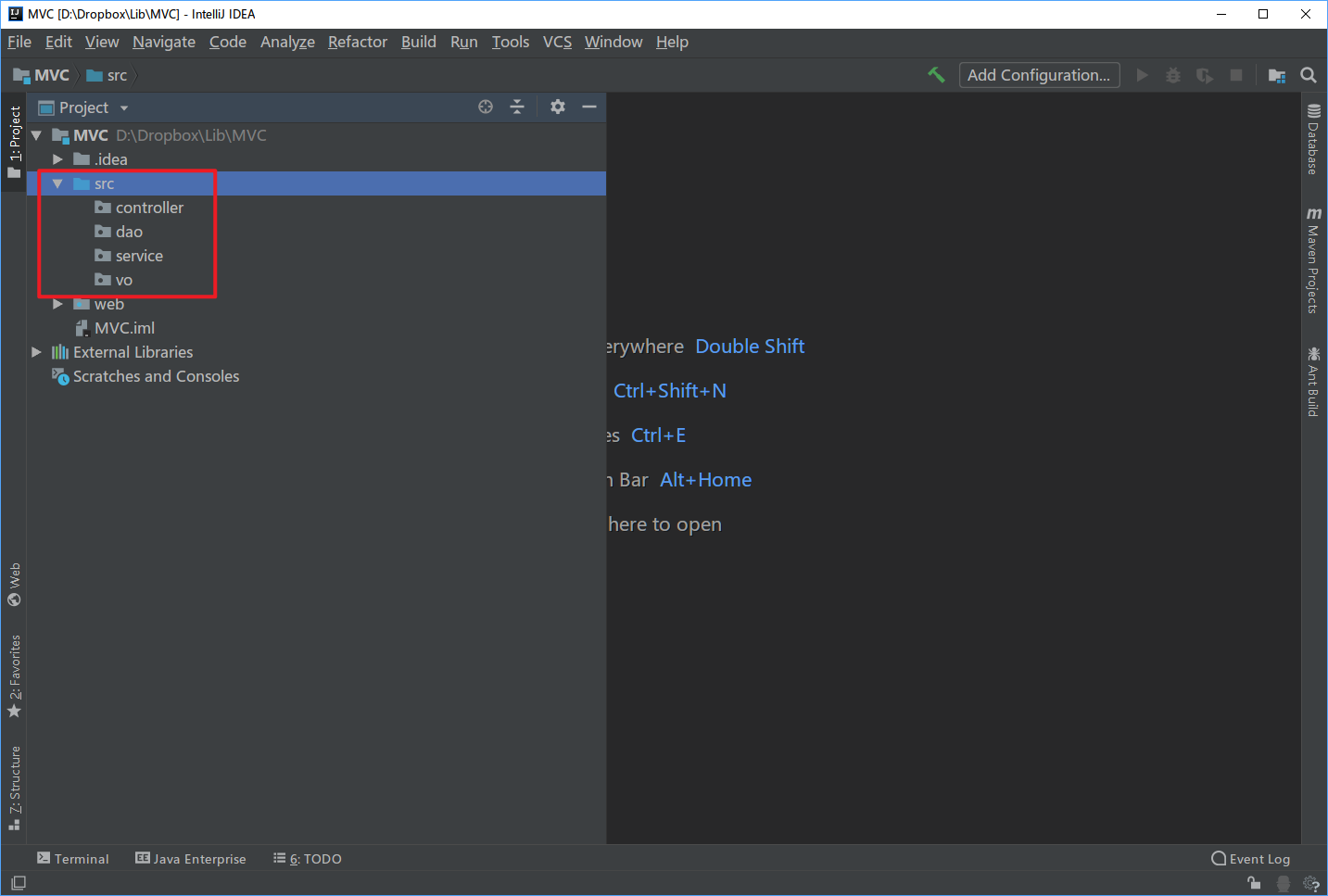
* **web目录下新建项目所需文件夹**

分别建立css、js、images文件夹及相关文件（同实验六）

* + 建立vo、dao、service3、controller4个package

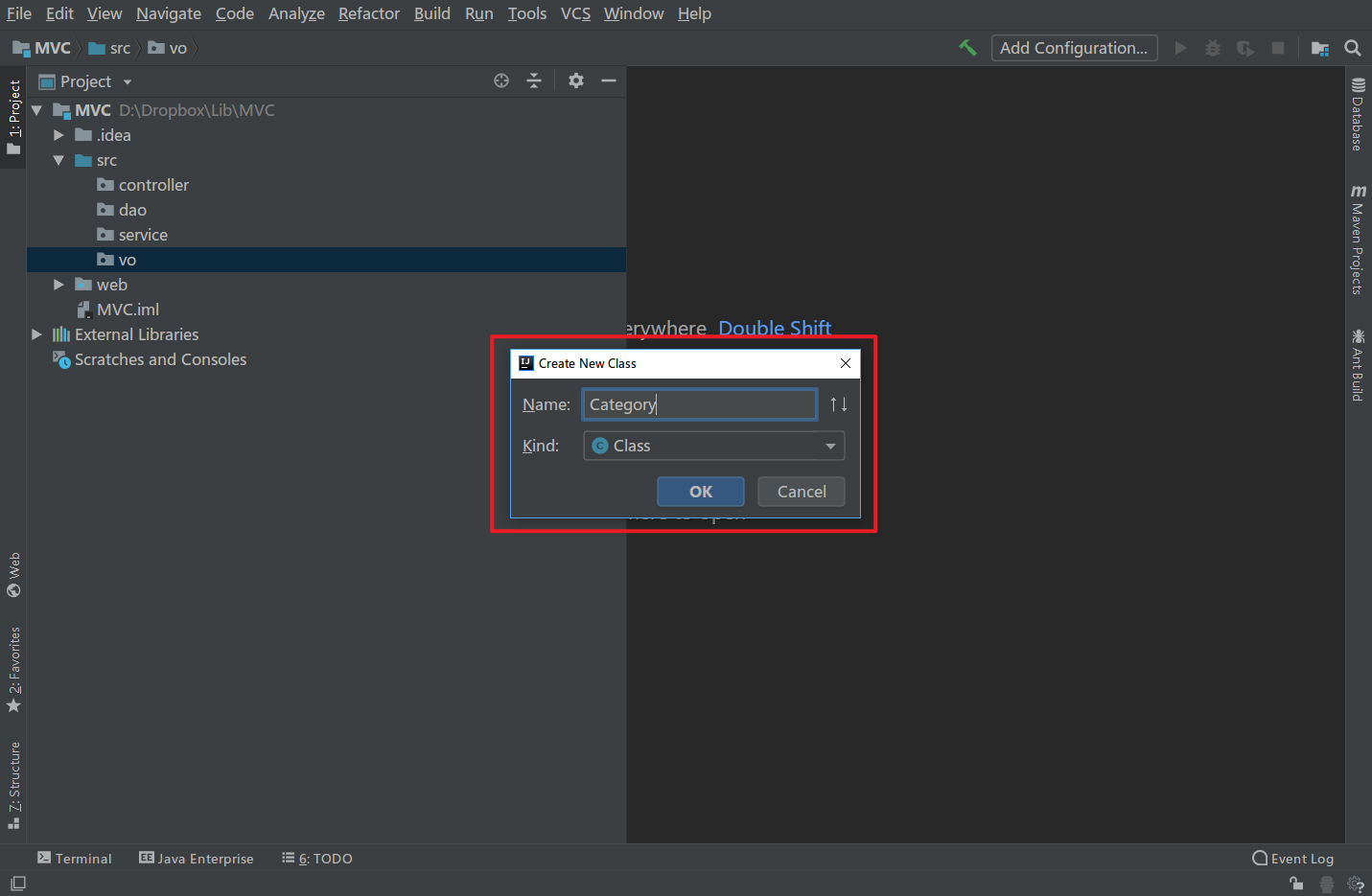
在src目录下建立4个package



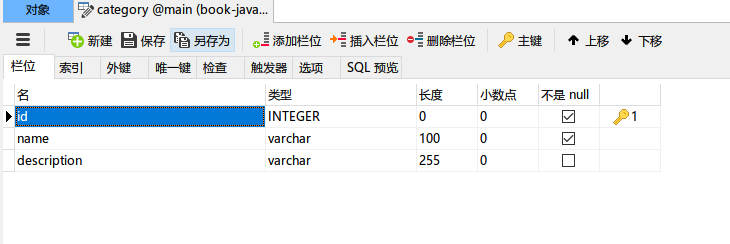


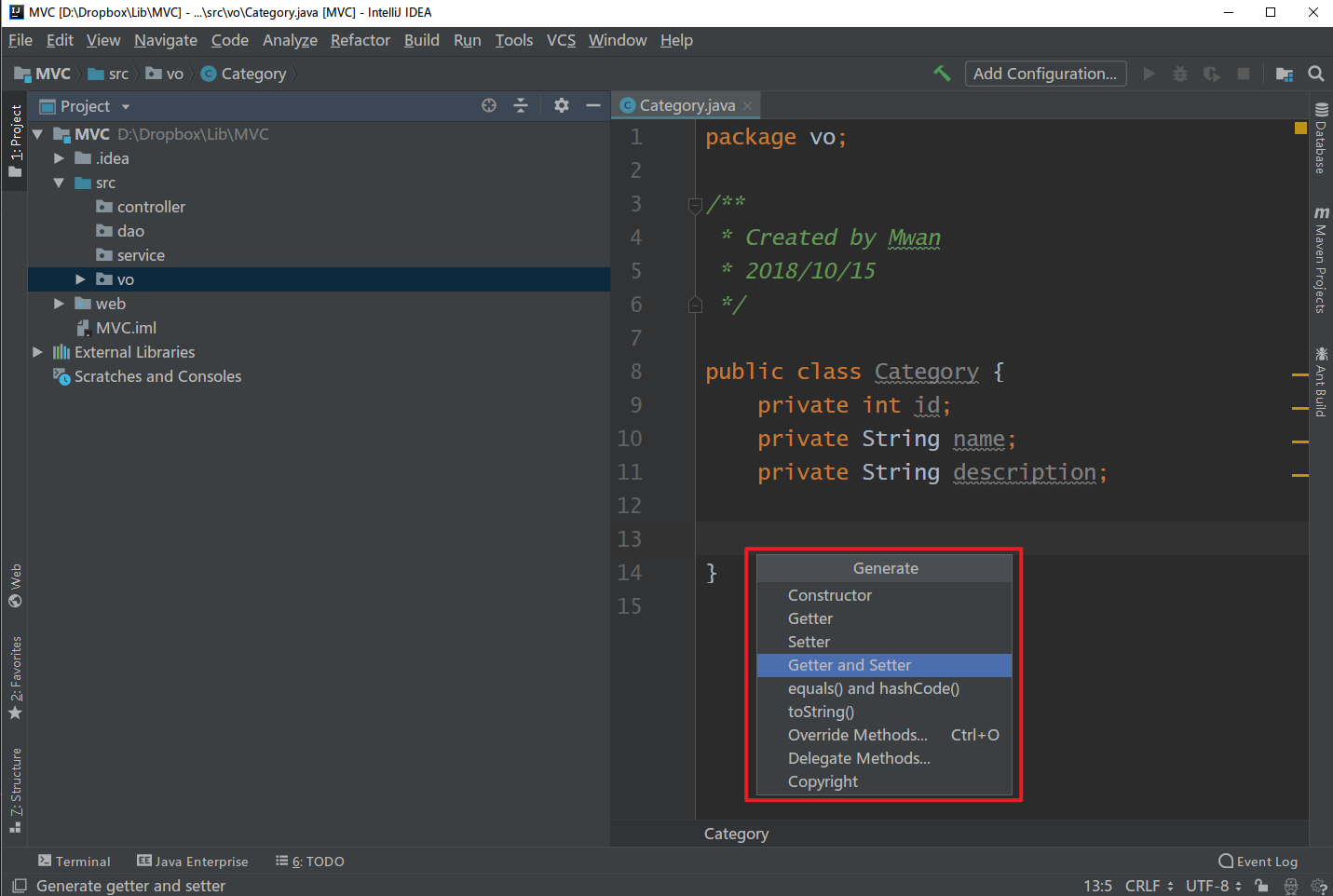
* 创建category对应的vo
  + 先完成类别查询功能，先创建类别的vo

在vo下创建category类



根据表结构建立相应属性





自动生成Getter和Setter方法

* 完成category对应的dao
  + 首先完成JDBC公共工具类，JDBCUtil

在dao下建立JDBCUtil类

public class JDBCUtil {

private static String DB\_URL ="jdbc:sqlite:D:/dropbox/bookstore.db";

private static String DB\_DRIVER = "org.sqlite.JDBC";

private static Connection connection = null;

public static Connection getConnection() {

try {

Class.forName(DB\_DRIVER);

connection = DriverManager.getConnection(DB\_URL);

} catch (ClassNotFoundException e) {

System.out.println("ClassNotFoundException");

e.printStackTrace();

} catch (SQLException e) {

System.out.println("连接数据库异常");

e.printStackTrace();

}

return connection;

}

public static void closeConnection(Connection connection) {

if(connection!=null) {

try {

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

* + 完成categoryDao接口

public interface CategoryDao {

public void add(Category category);

public void delete(Category category);

public List<Category> findAll();

}

注意引入正确的包

* + 完成categoryDao实现类

在dao下新建impl package，然后新建categoryDaoImpl类，实现对应的方法。

@Override

public List<Category> findAll() {

Connection connection = null;

Statement stat = null;

List<Category> list = new ArrayList<Category>();

try {

connection = JDBCUtil.getConnection();

stat = (Statement)connection.createStatement();

String sql = "select id,name,description from category";

ResultSet resultSet = (ResultSet)stat.executeQuery(sql);

while (resultSet.next()){

Category category = new Category();

category.setId(resultSet.getInt("id"));

category.setName(resultSet.getString("name"));

category.setDescription(resultSet.getString("description"));

list.add(category);

}

} catch (SQLException e) {

e.printStackTrace();

}

finally {

JDBCUtil.closeConnection(connection);

}

return list;

}

（注：后面例子中只用到了查询方法，故知列出findAll方法）

这时已完成关于category的model代码

* 完成category对应的service

本例业务简单，故不按照接口-实现方式完成，直接编写实现方法

在service中新建CategoryService类

public class CategoryService {

private CategoryDao category = new CategoryDaoImpl();

public List<Category> findAll() {

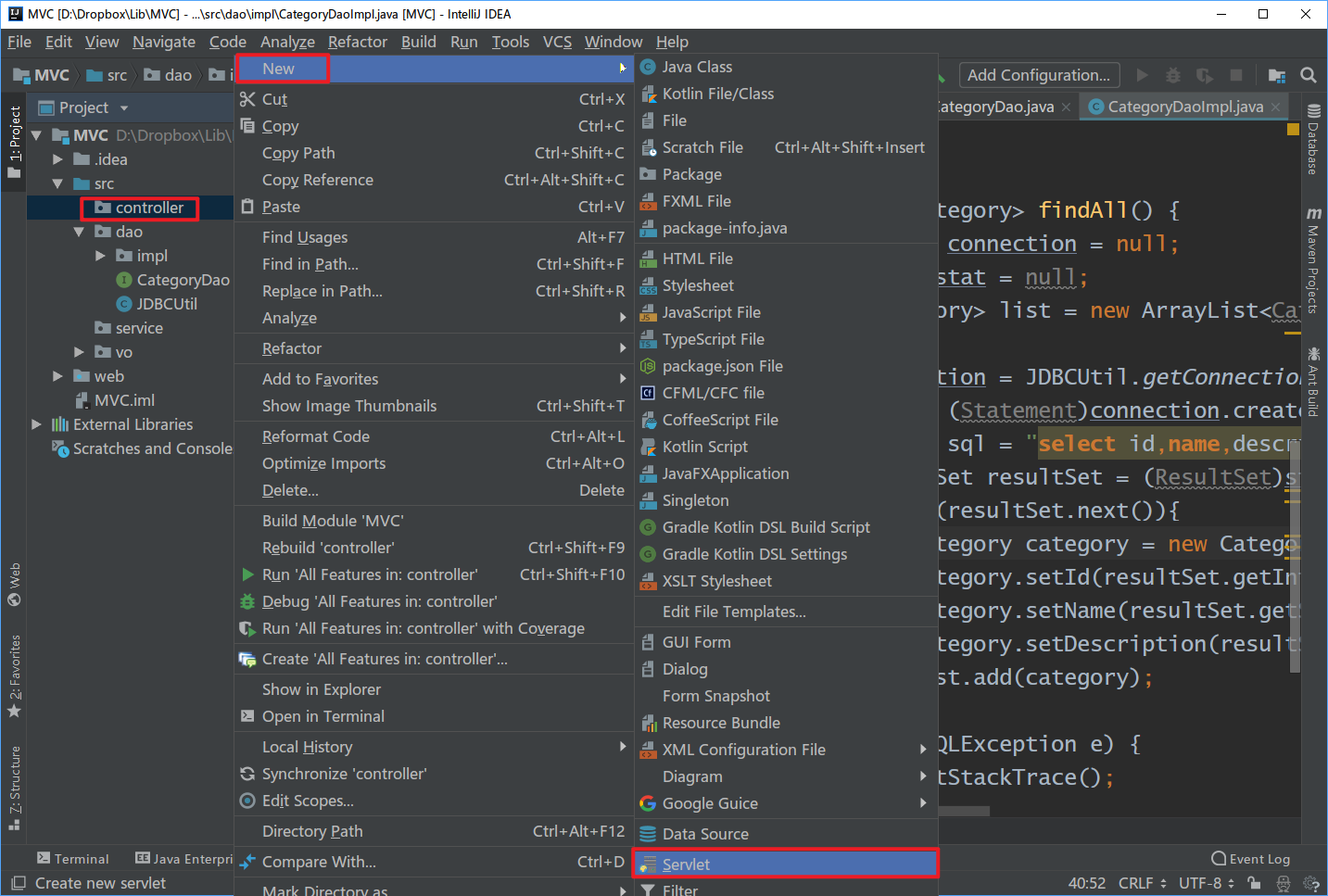
return category.findAll();

}

}

* 实现controller功能

在controller包下新建一个servlet



通过@WebServlet注解实现servlet和url的映射路径 /category

request对象中放入JavaBean对象，并传至前端JSP页面

@WebServlet("/category")

public class CategoryController extends javax.servlet.http.HttpServlet {

protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {

}

protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {

CategoryService categoryService = new CategoryService();

List<Category> list = categoryService.findAll();

request.setAttribute("categories",list);

request.getRequestDispatcher("/main.jsp").forward(request,response);

}

}

至此controller端完成

* view端开发

在web目录下建立main.jsp文件，并将实验六中的index.jsp代码复制，删去java和ajax代码只保留样式和布局代码。并将login.html和showID.jsp文件一并拷贝

main.jsp代码：

<html>

<head>

<link href="css/bootstrap.min.css" rel="stylesheet"/>

<link href="css/style.css" rel="stylesheet"/>

<title>网上书店</title>

</head>

<body onload="initAJAX()">

<div class="modal fade" id="myModal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header"></div>

<div class="modal-body"></div>

<div class="modal-footer"></div>

</div>

</div>

</div>

<div class="header">

<div class="container">

<div class="row">

<div class="login span4">

<h1><a href=""> 欢迎来到<strong>我的</strong>书店</a>

<span class="red">.</span></h1>

</div>

<div class="links span8">

<a class="login" href="login.html" rel="tooltip" data-placement="bottom" data-toggle="modal" data-target="#myModal"></a>

<a class="register" href="" rel="tooltip" data\_placement="bottom"></a>

</div>

</div>

</div>

</div> <%--header--%>

<div class="row"> <%--下方左右div控制--%>

<div class="col-md-3"> <%--左侧菜单div控制--%>

<ul class="nav nav-list">

<li class="nav-header">书籍类型</li>

</ul>

</div><%--左侧菜单div控制--%>

<div class="col-md-9" id="book"><%-- 书籍布局控制--%>

</div> <%-- 书籍布局控制--%>

</div><%--下方左右div控制--%>

<script src="js/jquery.min.js" ></script>

<script src="js/bootstrap.min.js"></script>

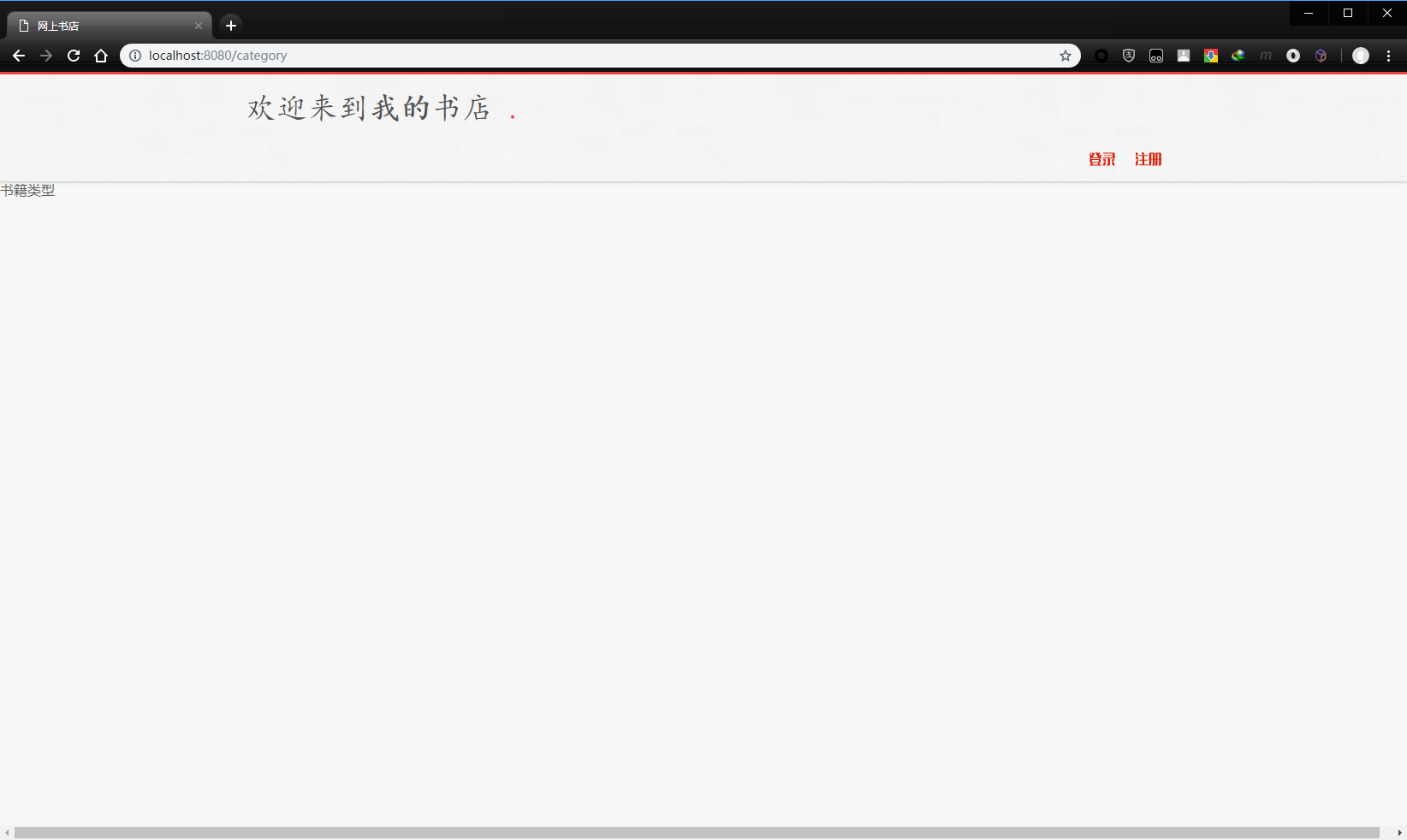
<script src="js/commons.js"></script>

</body>

</html>

配置IDEA服务器等设置，以及导入sqlite jdbc驱动jar包，运行项目，访问地址：

<http://localhost:8080/category> 查看显示是否正常，乱码问题解决同实验六



编辑main.jsp代码，接收并解析request数据，在左侧菜单div中编辑如下代码

<div class="row"> <%--下方左右div控制--%>

<div class="col-md-3"> <%--左侧菜单div控制--%>

<ul class="nav nav-list">

<li class="nav-header">书籍类型</li>

<%

Object categories = request.getAttribute("categories");

List<Category> categoryList = null;

if (categories instanceof List) {

categoryList = (List<Category>)categories;

}

if (categories != null) {

for (Category category:categoryList) {

%>

<li>

<a href="<%=category.getId()%>")>

<%=category.getName()%>

</a>

</li>

<%

}

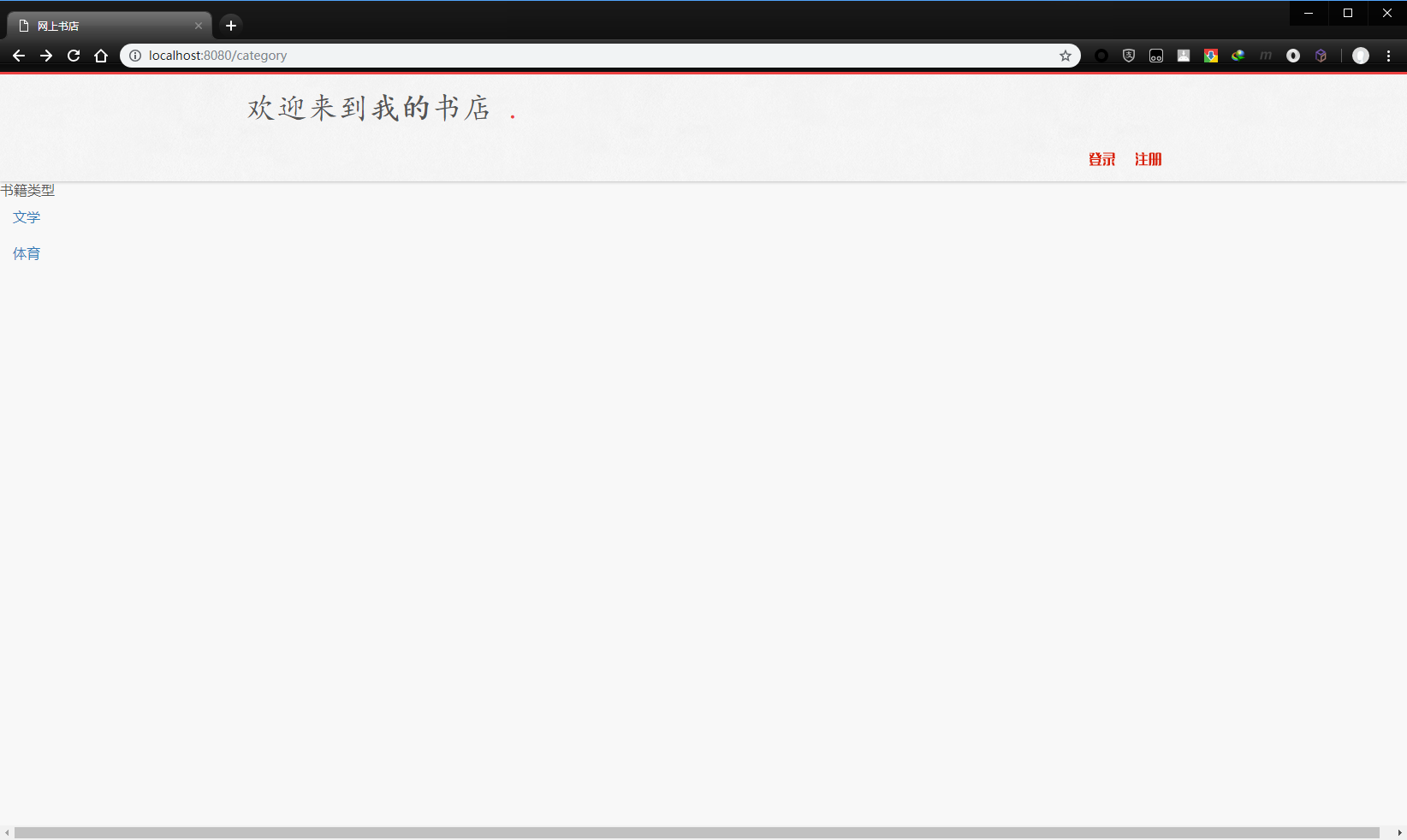
}

%>

</ul>

</div><%--左侧菜单div控制--%>

运行项目查看显示是否正常



* category显示功能完成，开始进行book查询、显示功能开发

自行完成book的vo、dao、service

在book的dao中需提供一个按指定类别ID的书籍信息查询方法，代码如下：

public class BookDaoImpl implements BookDao {

@Override

public List<Book> getBookByCategory(int category\_id) {

List<Book> list = new ArrayList<Book>();

Connection connection = null;

PreparedStatement preparedStatement = null;

ResultSet rs = null;

try {

connection = JDBCUtil.getConnection();

String sql = "select id,name,author,price,image,description,category\_id from book where category\_id = ?";

preparedStatement = connection.prepareStatement(sql);

preparedStatement.setInt(1,category\_id);

rs = preparedStatement.executeQuery();

while (rs.next()) {

Book book = new Book();

book.setId(rs.getInt("id"));

book.setName(rs.getString("name"));

book.setAuthor(rs.getString("author"));

book.setPrice(rs.getDouble("price"));

book.setImage(rs.getString("image"));

book.setDescription(rs.getString("description"));

book.setCategory\_id(rs.getString("category\_id"));

list.add(book);

}

} catch (SQLException e) {

e.printStackTrace();

}

finally {

JDBCUtil.closeConnection(connection);

}

return list;

}

// 其他方法

}

编写book的controller，在controller包下新建servlet，代码如下：

@WebServlet("/getBookByCategoryID")

public class BookController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

BookService bookService = new BookService();

int id = Integer.parseInt(request.getParameter("id"));

List<Book> bookList = bookService.getBookByCategoryID(id);

request.setAttribute("books",bookList);

request.getRequestDispatcher("/main.jsp").forward(request,response);

}

}

* 进行book查询功能的view开发

首先修改左侧类别显示的href

%>

<li>

<a href="/getBookByCategoryID?id=<%=category.getId()%>")>

<%=category.getName()%>

</a>

</li>

<%

修改main.jsp中book显示布局部分代码

<div class="col-md-9"><%--右侧书本div控制--%>

<%

Object books = request.getAttribute("books");

List<Book> listBook = null;

if (books instanceof List) {

listBook =(List<Book>)books;

}

if(listBook!=null) {

for(Book book:listBook) {

String desc = book.getDescription();

if(desc.length() > 18) {

desc = desc.substring(1,18) + "...";

}

%>

<div class="col-sm-9 col-md-3">

<div class="thumbnail">

<img src="/images/book.jpg"

alt="通用的占位符缩略图">

<div class="caption">

<h4><%=book.getName()%></h4>

<p><%=desc%></p>

<p>

<a href="#" class="btn btn-primary" role="button">

按钮

</a>

<a href="#" class="btn btn-default" role="button">

按钮

</a>

</p>

</div>

</div>

</div>

<%

}

}

%>

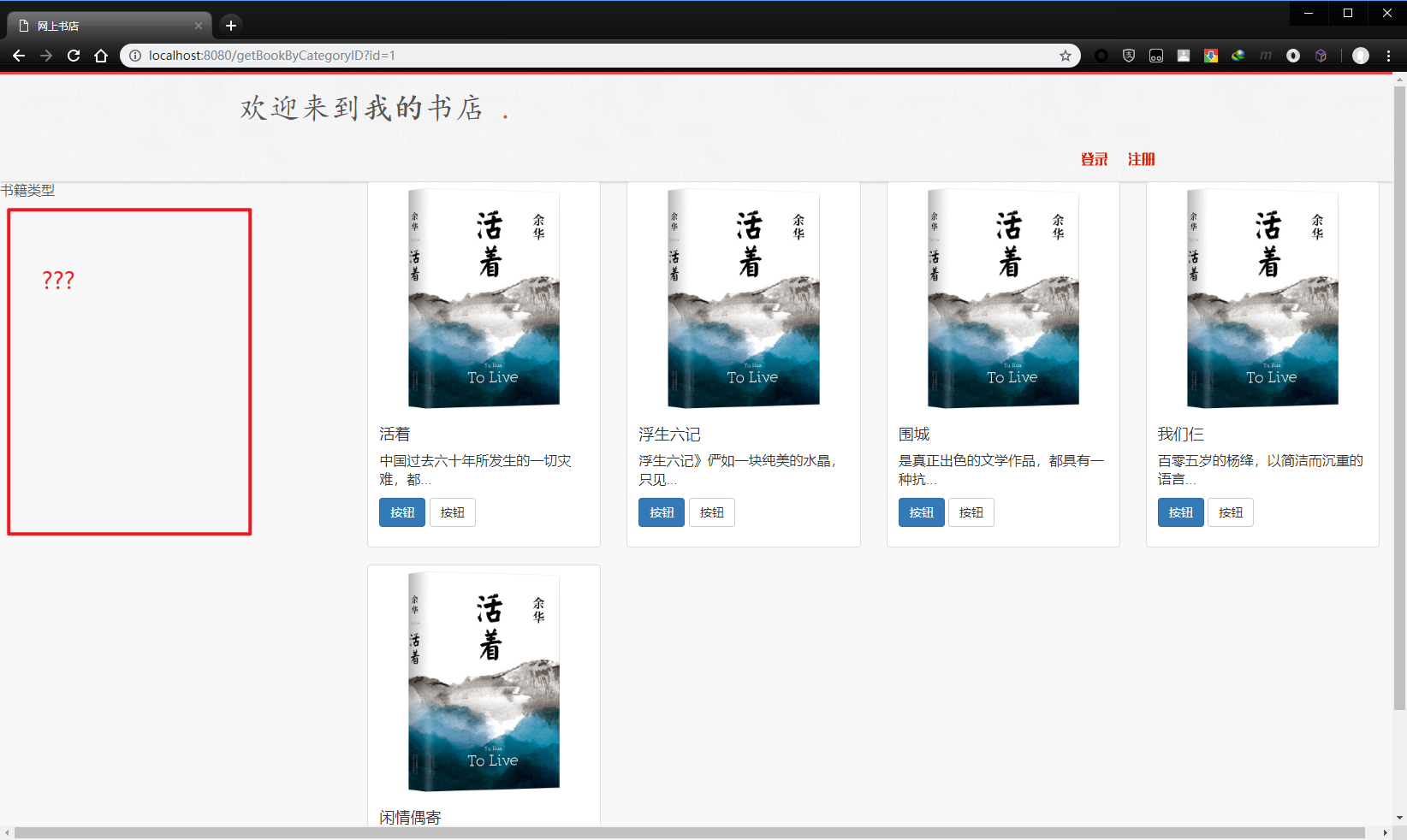
</div><%--右侧书本div控制--%>

运行项目，访问地址：

<http://localhost:8080/category>

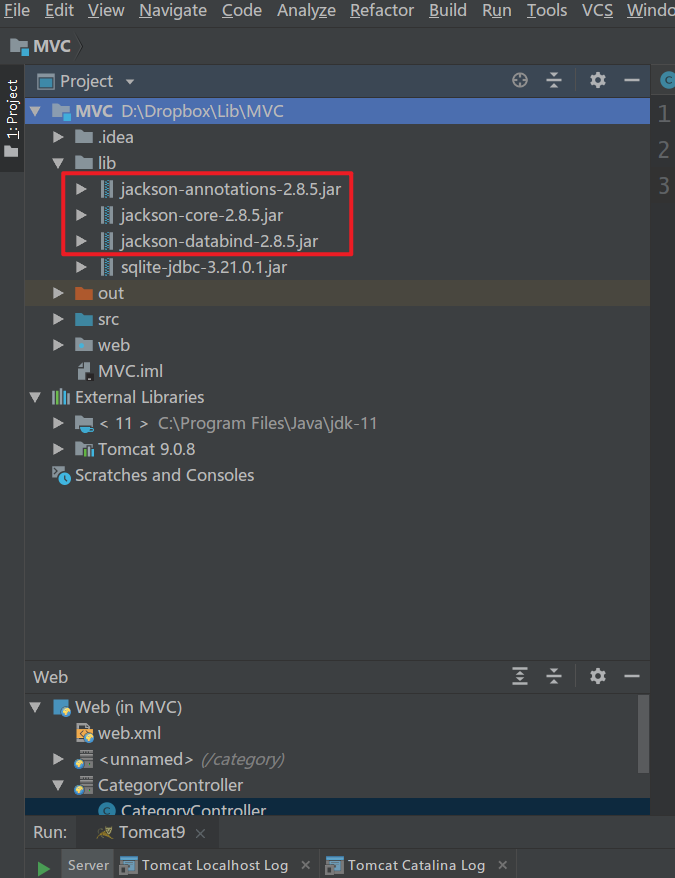
并点击左侧类别链接进行测试

发现book显示正常，但是类别信息会消失



思考题2：如何用最小代码修改量修复此bug？

* 使用json传输数据
  + 导入jaskson文件夹中3个jar包



* + 修改BookController为json返回

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

BookService bookService = new BookService();

int id = Integer.parseInt(request.getParameter("id"));

List<Book> bookList = bookService.getBookByCategoryID(id);

// request.setAttribute("books",bookList);

// request.getRequestDispatcher("/main.jsp").forward(request,response);

response.setCharacterEncoding("UTF-8");

ObjectMapper mapper = new ObjectMapper();

String jsonStr = mapper.writeValueAsString(bookList);

PrintWriter out = response.getWriter();

out.write(jsonStr);

}

运行项目访问地址

<http://localhost:8080/getBookByCategoryID?id=1>



* + 同样修改CategoryController代码

protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {

CategoryService categoryService = new CategoryService();

List<Category> list = categoryService.findAll();

ObjectMapper mapper = new ObjectMapper();

String jsonStr = mapper.writeValueAsString(list);

response.setCharacterEncoding("UTF-8");

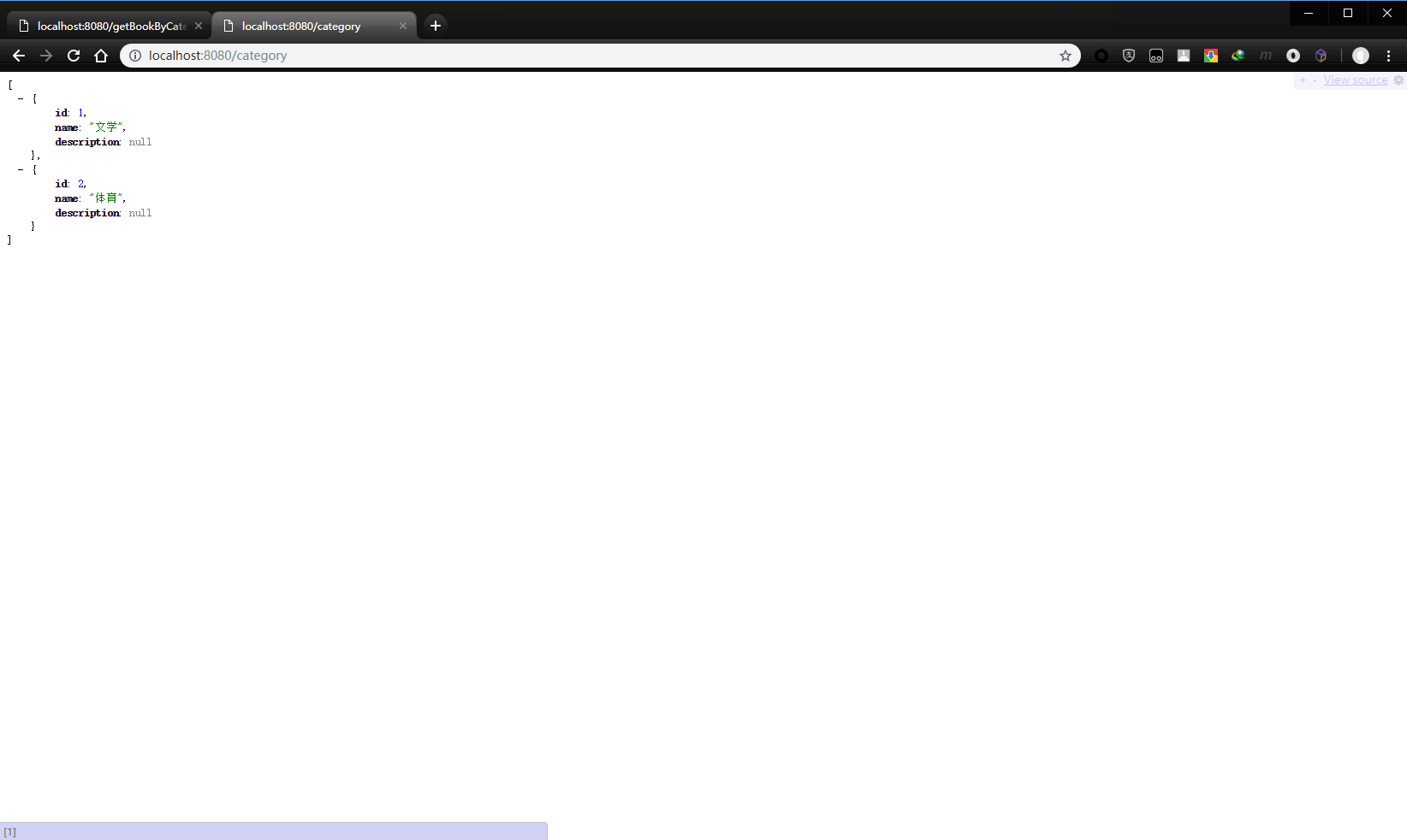
PrintWriter out = response.getWriter();

out.write(jsonStr);

}

访问地址：

http://localhost:8080/category



* + 修改main.jsp代码使用ajax方法请求数据

删除main.jsp中<div class="row"> </div>中所有java代码和部分html代码，最终代码为：

<div class="row"> <%--下方左右div控制--%>

<div class="col-md-3"> <%--左侧菜单div控制--%>

<ul class="nav nav-list" id="categorylist">

</ul>

</div><%--左侧菜单div控制--%>

<div class="col-md-9" id="book"><%--右侧书本div控制--%>

</div><%--右侧书本div控制--%>

</div><%--下方左右div控制--%>

添加ajax函数获取书籍分类信息

<script language="JavaScript">

function showCategory() {

xmlHttp.open("GET","/category",true);

xmlHttp.onreadystatechange = function () {

if(xmlHttp.readyState==4) {

var data = xmlHttp.responseText;

var obj = JSON.parse(data);

var listHtml = "<li class=\"nav-header\">书籍类别</li>\n" ;

for(var i in obj){

listHtml += "<li>\n" +

" <a href=\"javascript:showBook("+ obj[i].id + ")\">"+obj[i].name+"</a>\n" +

" </li>"

}

document.getElementById("categorylist").innerHTML=listHtml;

}

}

xmlHttp.send();

}

</script>

并在<body onload >时加载该方法

<body onload="initAJAX();showCategory()">

继续添加获取指定分类ID书籍信息方法

function showBook(categoryID) {

xmlHttp.open("GET", "getBookByCategoryID?id="+categoryID, true);

xmlHttp.onreadystatechange = function () {

if (xmlHttp.readyState == 4) {

var data = xmlHttp.responseText;

var obj = JSON.parse(data);

var listbook = '';

for(var i in obj){

var bookname = obj[i].name;

var desc = obj[i].description;

if(desc.length > 20) {

desc = desc.substring(0,17)

}

listbook += `<div class="col-sm-9 col-md-3"><div class="thumbnail" ><img src="images/book.jpg"><div class="caption"> <h4>`

+ bookname +

`</h4><p>`+desc+

`</p><p><a href="#" class="btn btn-primary" role="button">加入购物车</a> <a href="#" class="btn btn-default" role="button">查看详情`+

`</a></p></div></div> </div>`;

}

document.getElementById("book").innerHTML = listbook;

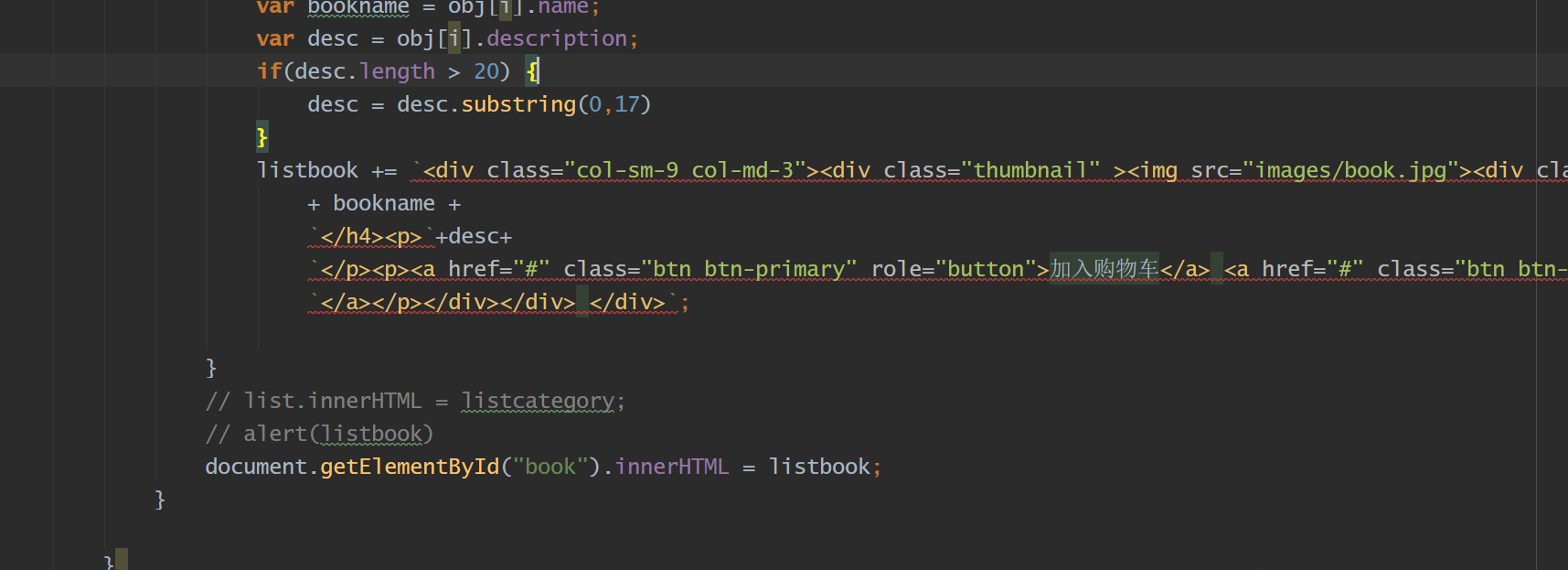
}

}

xmlHttp.send();

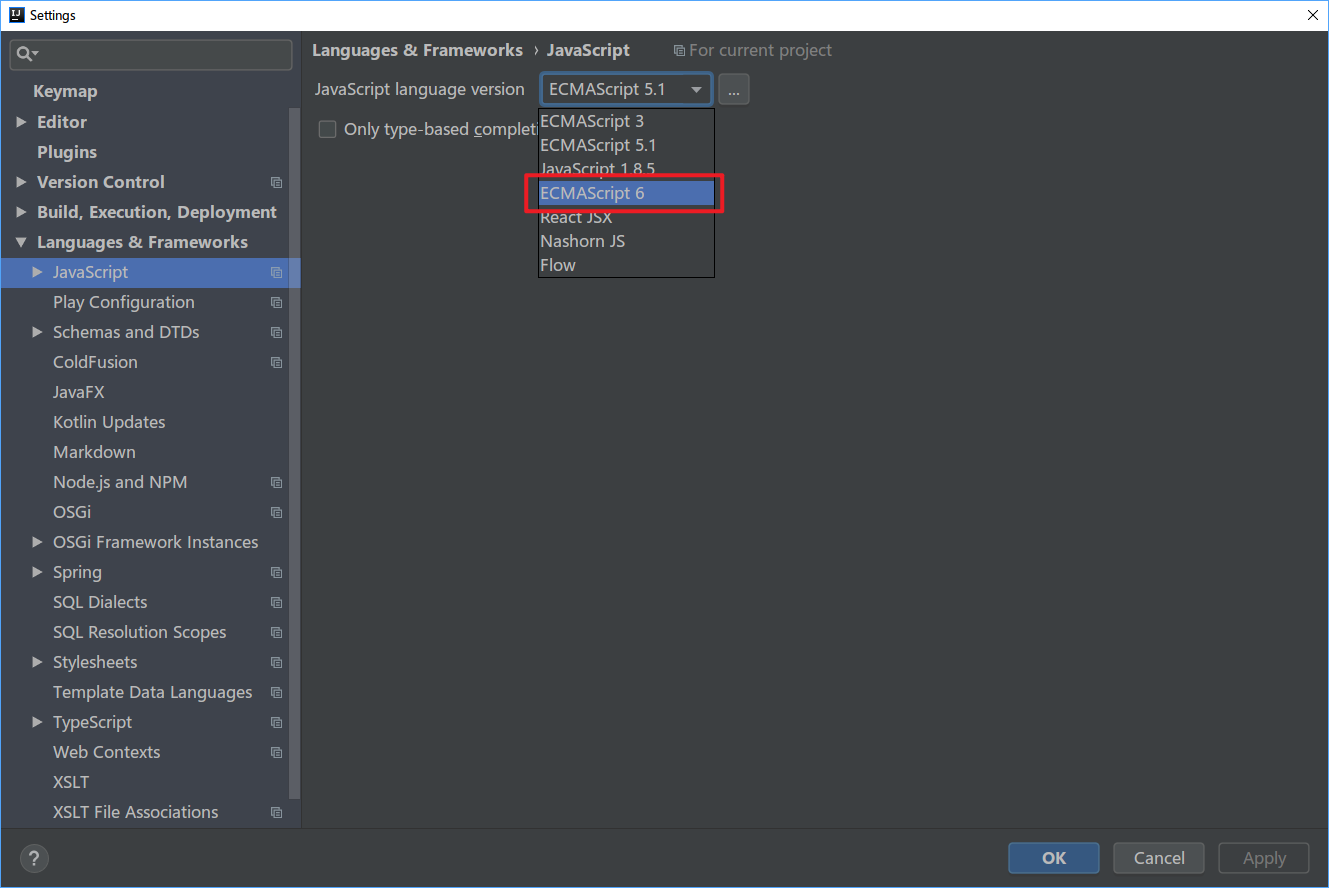
}

在IDEA中可能会出现以下错误提示：



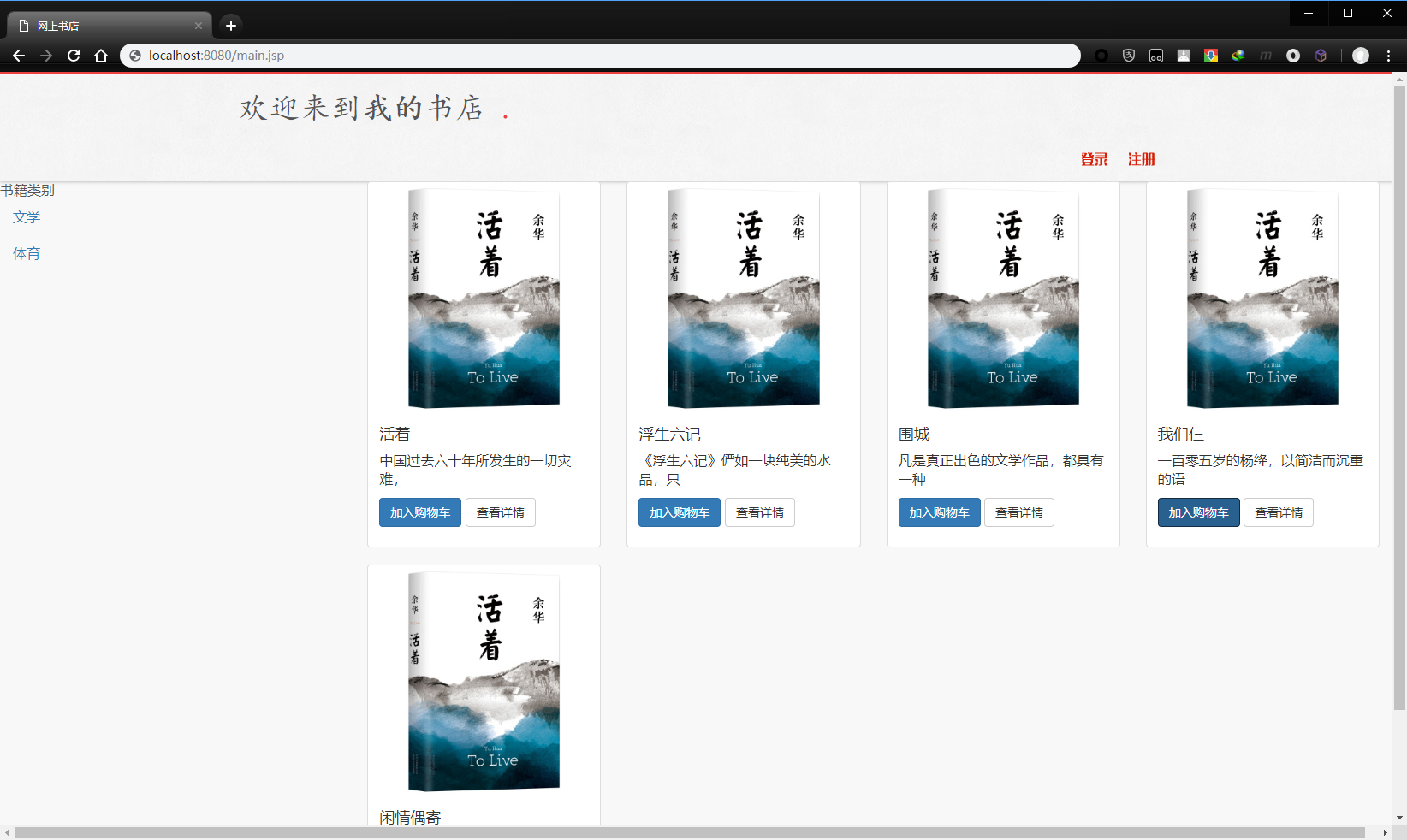
修改IDEA支持ES6语法即可

修改settings->language->javascript 支持ES6



至此前端代码中不再包含java代码，整个项目通过json传输数据的方式将前端开发和后端开发分开。

访问项目地址：<http://localhost:8080/main.jsp>



### 三、思考题

1、JDBCUtil的作用，以及它好处和不足？

2、在完成book功能查询和显示时发现类别信息会丢失，如何用最小代码修改量修复此bug？

3、在controller中转换java对象为json数据格式的代码可否放到service中？为什么？。