Spring MVC4 –i18n 、登录、拦截器

创建工程 estore4

将 estore3 相关文件复制过来，包名称中 estore3 修改为 estore4,能正常运行，进行下一步

数据库设计

会员表

```
-- Table structure for table `member`
--

DROP TABLE IF EXISTS `member`;
CREATE TABLE `member` (
  `memid` char(12) NOT NULL,
  `mempwd` char(12) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `regdate` datetime DEFAULT NULL,
  `logindate` datetime DEFAULT NULL,
  `logintimes` int(11) DEFAULT NULL,
  PRIMARY KEY (`memid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

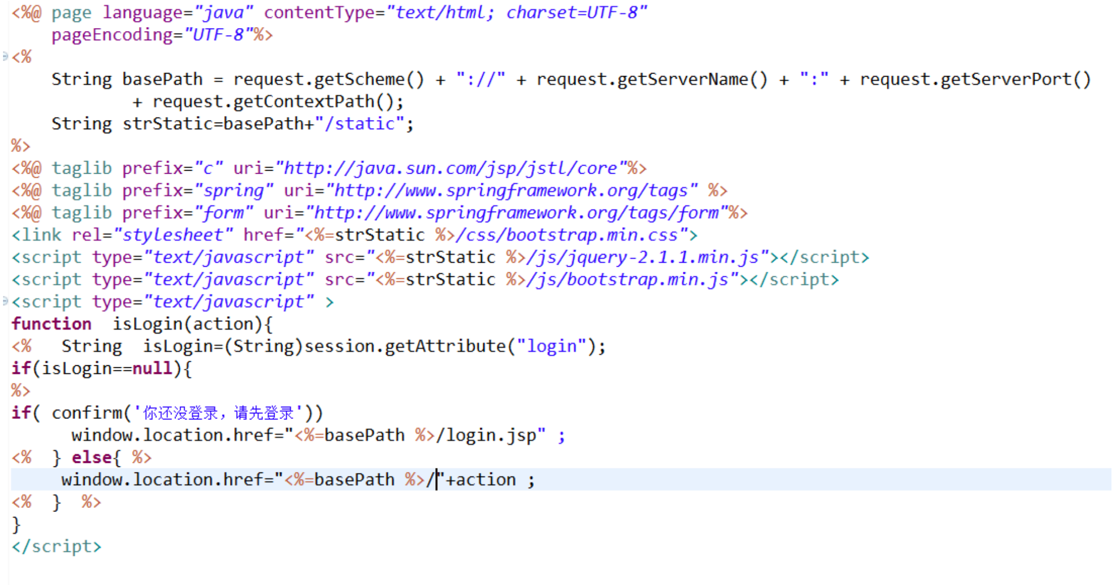
购物车表（暂时不用，选购商品中用会员 id 表示购物车 id）

```
-- Table structure for table `cart`

--

DROP TABLE IF EXISTS `cart`;

CREATE TABLE `cart` (

    `cartid` char(12) NOT NULL,

    `totalprice` float DEFAULT '0',

    PRIMARY KEY (`cartid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

选购商品表

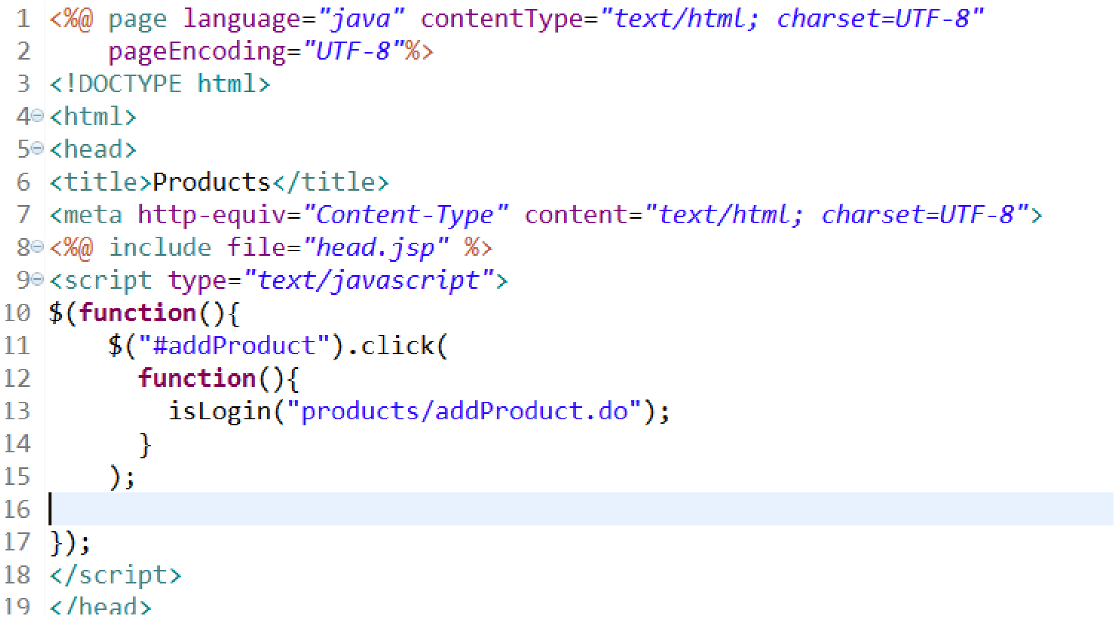
```
-- Table structure for table `cartitem`

--



DROP TABLE IF EXISTS `cartitem`;

CREATE TABLE `cartitem` (

    `id` int(11) NOT NULL AUTO_INCREMENT,

    `productid` char(20) DEFAULT NULL,

    `productcnt` int(11) DEFAULT NULL,
    `price` float DEFAULT NULL,

    `totalprice` float DEFAULT NULL,

    `cartid` char(12) DEFAULT NULL,

    PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

创建 head.jsp 文件（提取每个 jsp 中公用代码）



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort()
            + request.getContextPath();
    String strStatic=basePath+"/static";
%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<link rel="stylesheet" href="<%=strStatic %>/css/bootstrap.min.css">
<script type="text/javascript" src="<%=strStatic %>/js/jquery-2.1.1.min.js"></script>
<script type="text/javascript" src="<%=strStatic %>/js/bootstrap.min.js"></script>
<script type="text/javascript" >
function  isLogin(action){
<%   String  isLogin=(String)session.getAttribute("login");
if(isLogin==null){
%>
if( confirm('你还没登录，请先登录'))
      window.location.href="<%=basePath %>/login.jsp" ;
<%  } else{ %>
     window.location.href="<%=basePath %>/"+action ;
<%  }  %>
}
</script>
```

相应调整每个 jsp，比如 products.jsp（包含登录判断）



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Products</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<%@ include file="head.jsp" %>
<script type="text/javascript">
$(function(){
    $("#addProduct").click(
      function(){
        isLogin("products/addProduct.do");
      }
    );

});
</script>
</head>
```

参考之前分页，以分页的方式显示产品（每页 4 个商品）

i18n

在 src 下创建文件夹 resources，在文件夹下创建属性文件

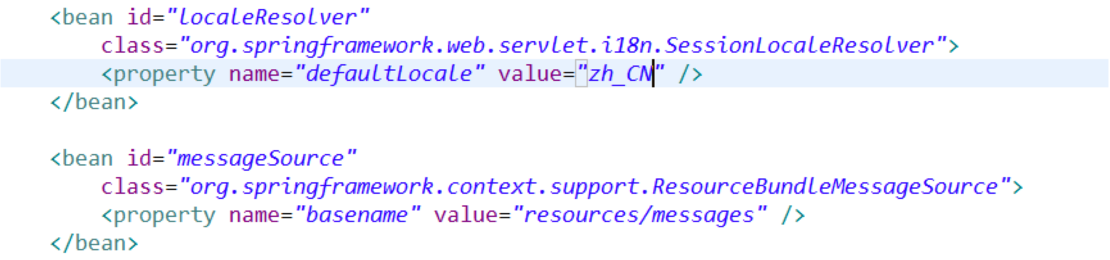
messages_en.properties

```
addProduct.form.productId.label = Product ID(en i18n)

addProduct.form.name.label = Product Name

addProduct.form.unitPrice.label = unitPrice

addProduct.form.description.label = description

addProduct.form.manufacturer.label = manufacturer

addProduct.form.category.label =     category

addProduct.form.unitsInStock.label = unitsInStock
addProduct.form.condition.label = condition

addProduct.form.productImage.label = productImage
```

messages_zh_CN.properties

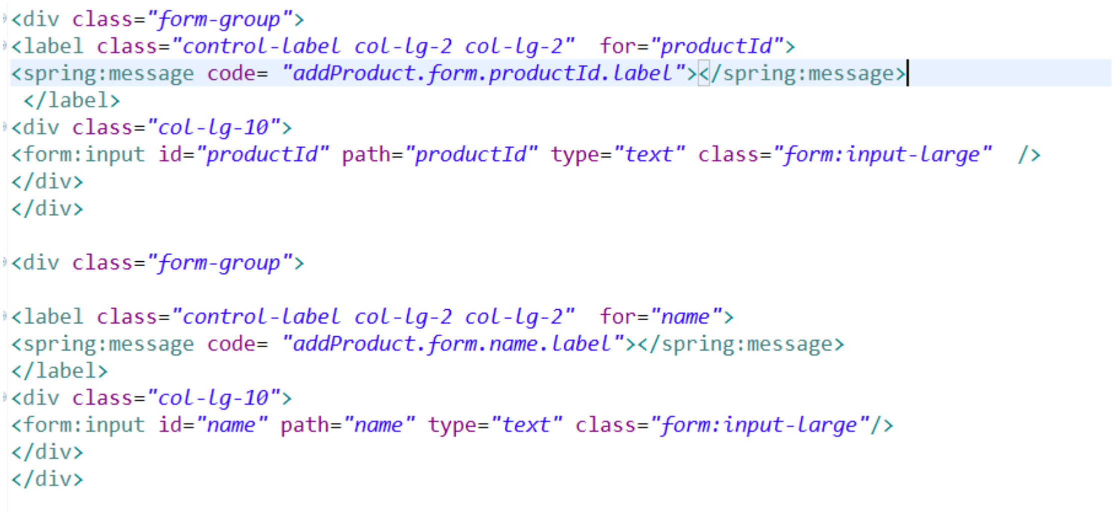
```
addProduct.form.productId.label =

\u4EA7\u54C1\u7F16\u53F7

addProduct.form.name.label = \u4EA7\u54C1\u540D\u79F0

addProduct.form.unitPrice.label = \u4EF7\u683C

addProduct.form.description.label = \u63CF\u8FF0

addProduct.form.manufacturer.label = \u5236\u9020\u5546

addProduct.form.category.label = \u7C7B\u522B

addProduct.form.unitsInStock.label =

\u5E93\u5B58\u6570\u91CF

addProduct.form.condition.label = condition

addProduct.form.productImage.label = productImage
```

编辑框架配置文件



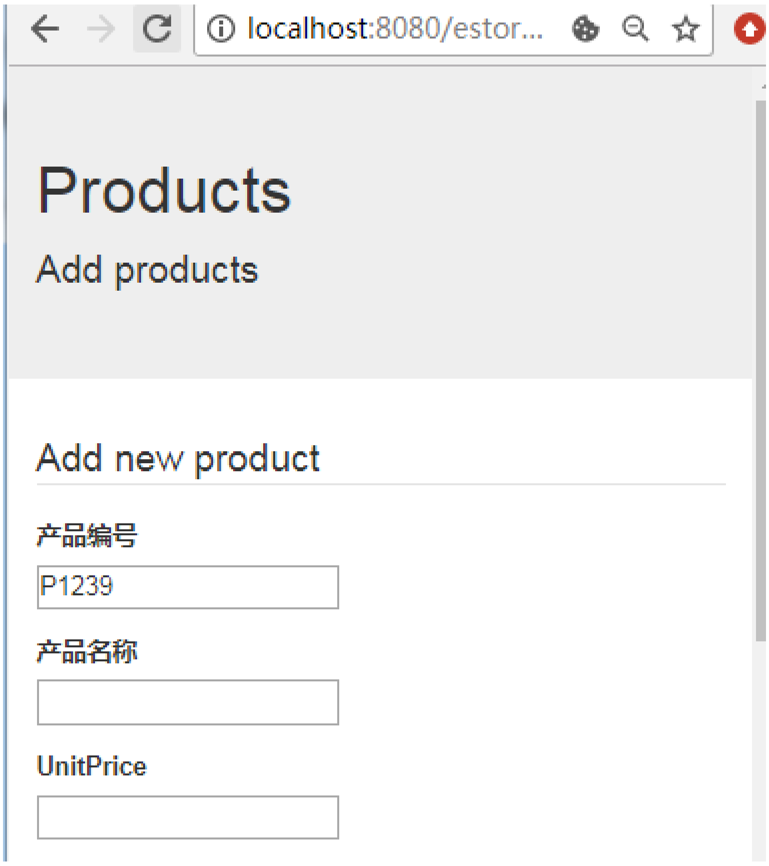
```
<bean id="localeResolver"
    class="org.springframework.web.servlet.i18n.SessionLocaleResolver">
    <property name="defaultLocale" value="zh_CN" />
</bean>

<bean id="messageSource"
    class="org.springframework.context.support.ResourceBundleMessageSource">
    <property name="basename" value="resources/messages" />
</bean>
```

编辑 addProduct.jsp



```
<div class="form-group">
<label class="control-label col-lg-2 col-lg-2"  for="productId">
<spring:message code= "addProduct.form.productId.label"></spring:message>
 </label>
<div class="col-lg-10">
<form:input id="productId" path="productId" type="text" class="form:input-large"  />
</div>
</div>

<div class="form-group">

<label class="control-label col-lg-2 col-lg-2"  for="name">
<spring:message code= "addProduct.form.name.label"></spring:message>
</label>
<div class="col-lg-10">
<form:input id="name" path="name" type="text" class="form:input-large"/>
</div>
</div>
```

访问增加产品页面（显示中文）



Products

Add products

Add new product

产品编号

P1239

产品名称

UnitPrice

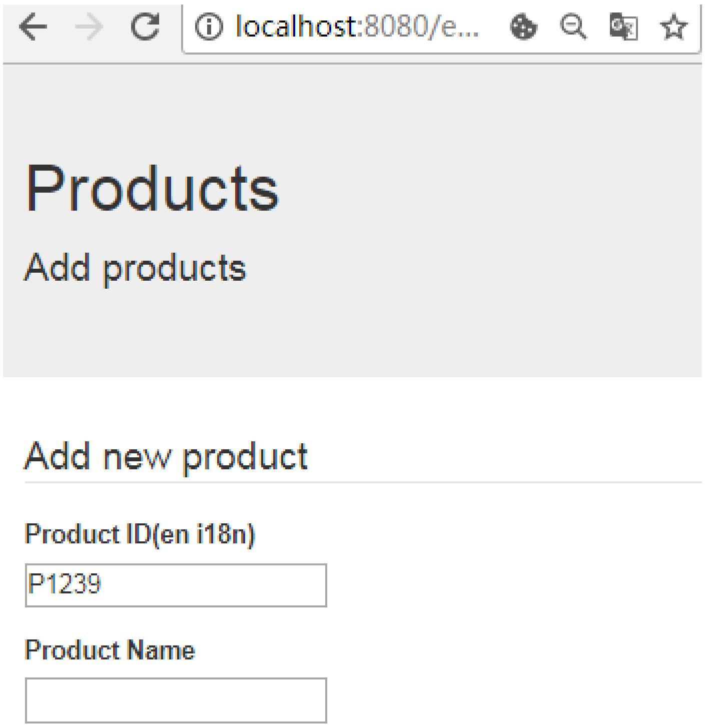
```
<bean id="localeResolver"
    class="org.springframework.web.servlet.i18n.Session
```

```
LocaleResolver">
      <property name="defaultLocale" value="zh_CN" />
   </bean>
```

由zh_CN修改为en

```
<property name="defaultLocale" value="en" />
```

访问增加产品页面（显示英文）



Products

Add products

Add new product

Product ID(en i18n)

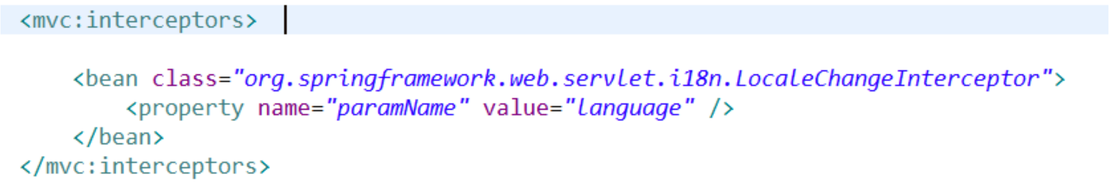
P1239

Product Name

实现中英文切换

编辑框架配置文件

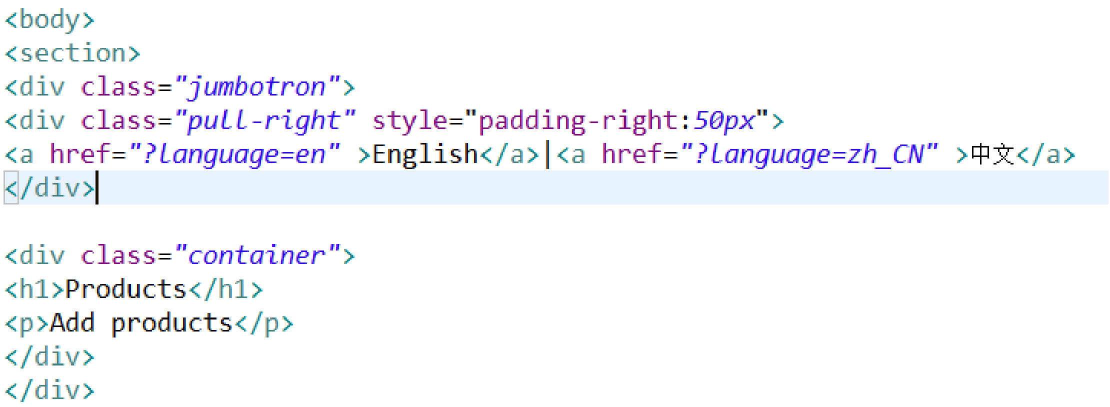
配置一个本地化拦截器



```
<mvc:interceptors>

    <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
        <property name="paramName" value="language" />
    </bean>
</mvc:interceptors>
```

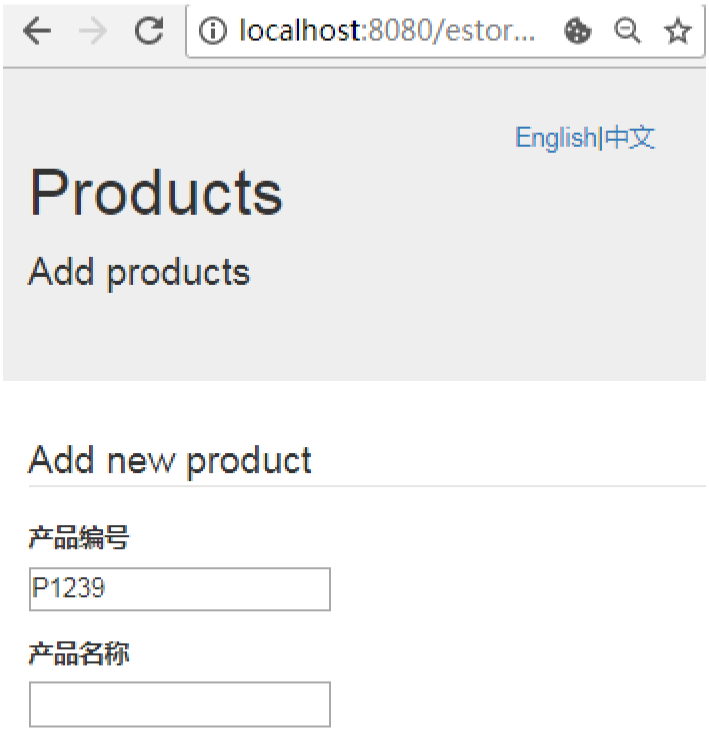
编辑 addProduct.jsp

增加一个中英文切换



```
<body>
<section>
<div class="jumbotron">
<div class="pull-right" style="padding-right:50px">
<a href="?language=en" >English</a>|<a href="?language=zh_CN" >中文</a>
</div>

<div class="container">
<h1>Products</h1>
<p>Add products</p>
</div>
</div>
```

访问效果



English|中文

Products

Add products

Add new product

产品编号

P1239

产品名称

实现登录功能

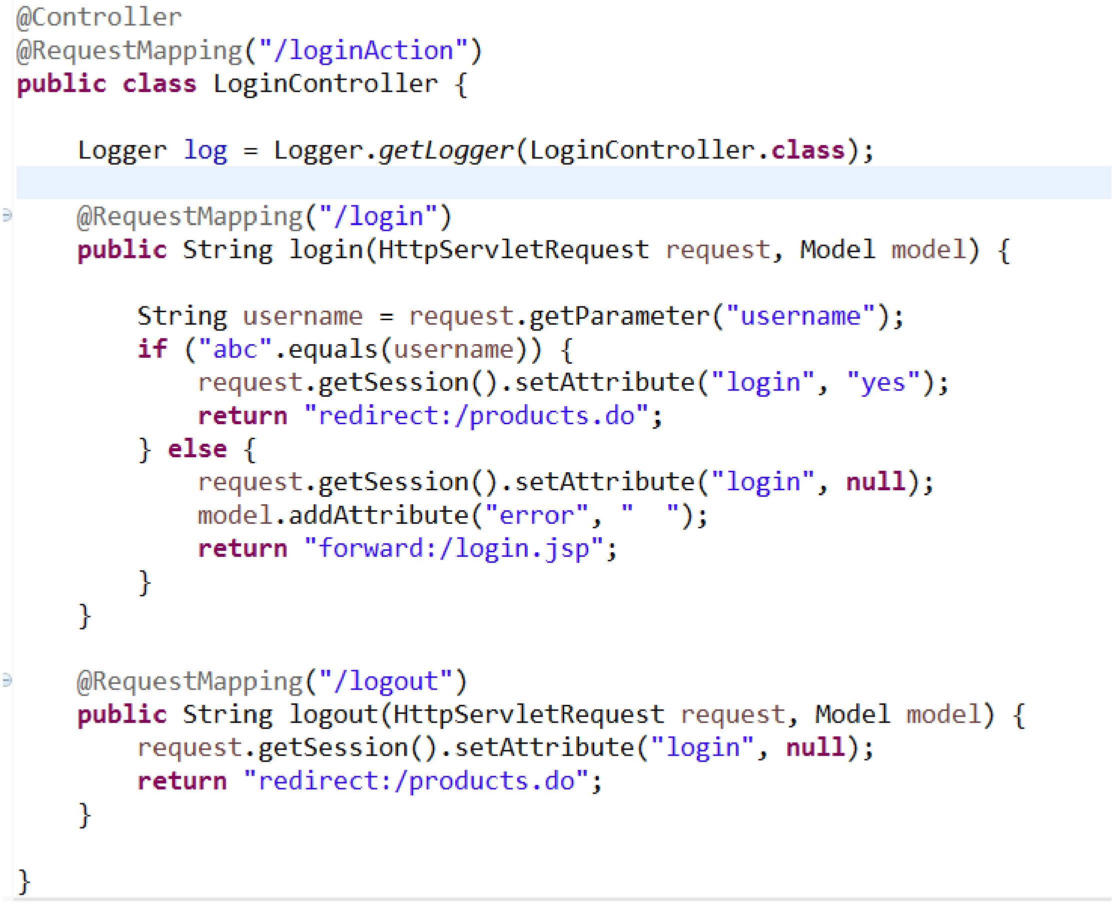
在 **webcontent** 下创建 **login.jsp**

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Products</title>
<%@ include file="WEB-INF/jsp/head.jsp" %>
</head>
<body>
<section>
<div class="jumbotron" style="background-color: white">
<div class="container">
<br><br><br>
</div>
</div>
```

```
</section>

<div class="container">
<div class="row">
<div class="col-md-4 col-md-offset-4">
<div class="panel panel-default">

<div class="panel-heading"><h3
class="panel-title">Please sign in</h3>
</div>
<div class="panel-body">

<c:if test="${not empty error}">
<div class="alert alert-danger">
<spring:message code="loginInfo.failure"/><br />
</div>
</c:if>

<form action="<c:url value=
"/loginAction/login.do"></c:url>" method="post">
<fieldset>
<div class="form-group">
```

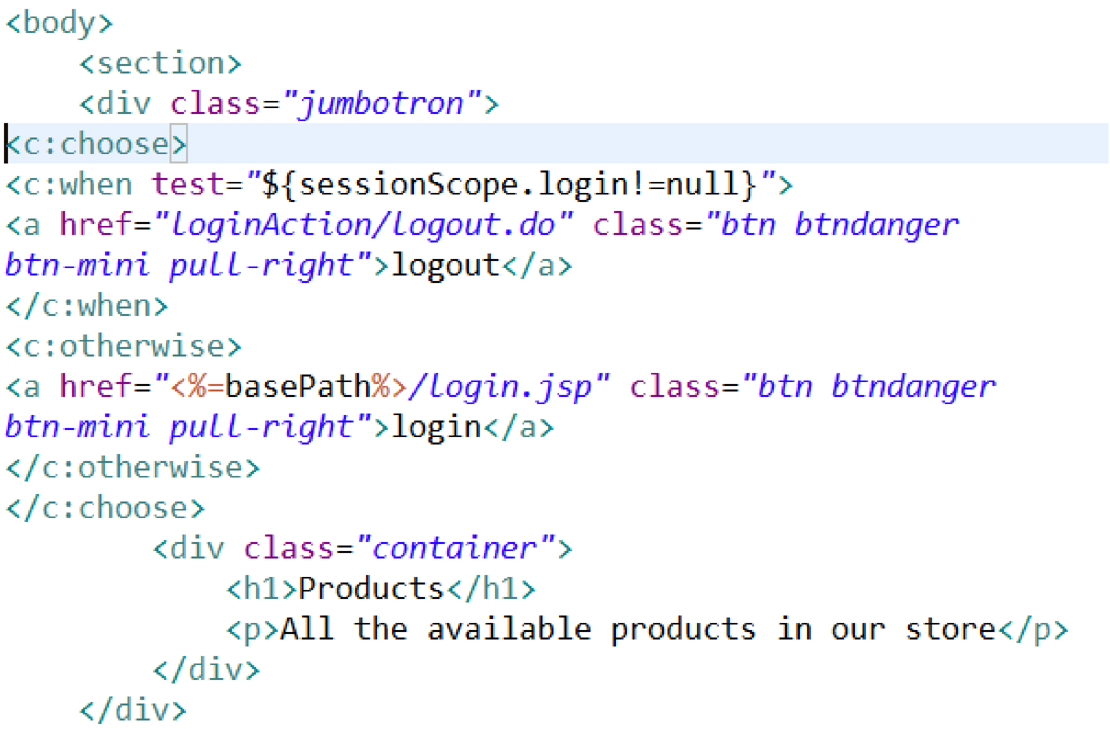
```
<input class="form-control" placeholder="User Name"

name='username' type="text">

</div>

<div class="form-group">

<input class="form-control" placeholder="Password"

name='password' type="password" value="">

</div>

<input class="btn btn-lg btn-success btn-block"

type="submit" value="Login">

</fieldset>

</form>

</div>


</div></div></div></div>

</body>

</html>
```

实现业务控制器（完成 service　dao 层）



```java
@Controller
@RequestMapping("/loginAction")
public class LoginController {

    Logger log = Logger.getLogger(LoginController.class);

    @RequestMapping("/login")
    public String login(HttpServletRequest request, Model model) {

        String username = request.getParameter("username");
        if ("abc".equals(username)) {
            request.getSession().setAttribute("login", "yes");
            return "redirect:/products.do";
        } else {
            request.getSession().setAttribute("login", null);
            model.addAttribute("error", "  ");
            return "forward:/login.jsp";
        }
    }

    @RequestMapping("/logout")
    public String logout(HttpServletRequest request, Model model) {
        request.getSession().setAttribute("login", null);
        return "redirect:/products.do";
    }

}
```

在显示产品页增加登录、登出（products.jsp）

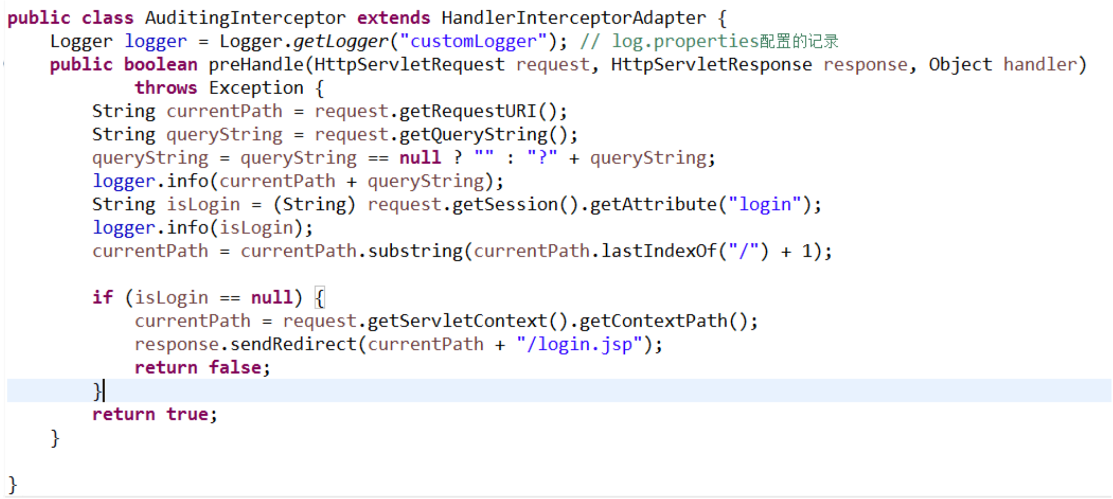


```
<body>
    <section>
    <div class="jumbotron">
<c:choose>
<c:when test="${sessionScope.login!=null}">
<a href="loginAction/logout.do" class="btn btndanger
btn-mini pull-right">logout</a>
</c:when>
<c:otherwise>
<a href="<%=basePath%>/login.jsp" class="btn btndanger
btn-mini pull-right">login</a>
</c:otherwise>
</c:choose>
        <div class="container">
            <h1>Products</h1>
            <p>All the available products in our store</p>
        </div>
    </div>
```

拦截器

控制登录拦截器

包 com.estore3.interceptor 下创建拦截器

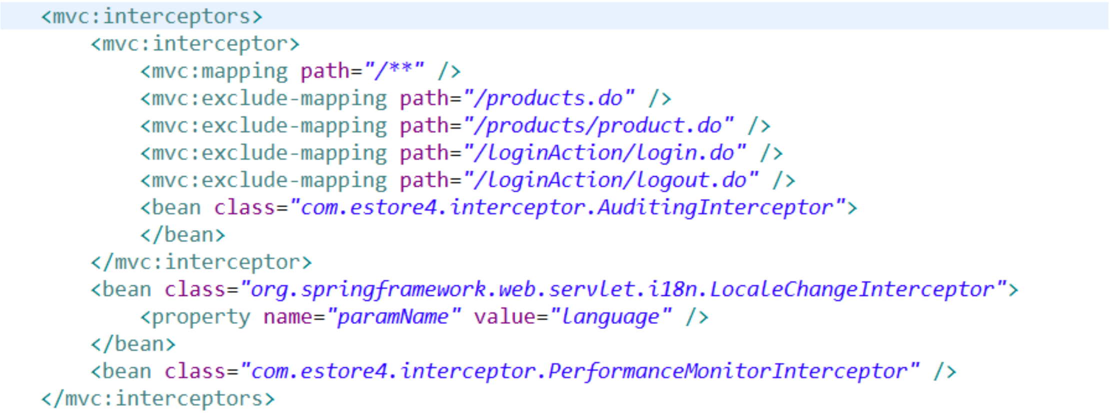


```
public class AuditingInterceptor extends HandlerInterceptorAdapter {
    Logger logger = Logger.getLogger("customLogger"); // log.properties配置的记录
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
            throws Exception {
        String currentPath = request.getRequestURI();
        String queryString = request.getQueryString();
        queryString = queryString == null ? "" : "?" + queryString;
        logger.info(currentPath + queryString);
        String isLogin = (String) request.getSession().getAttribute("login");
        logger.info(isLogin);
        currentPath = currentPath.substring(currentPath.lastIndexOf("/") + 1);

        if (isLogin == null) {
            currentPath = request.getServletContext().getContextPath();
            response.sendRedirect(currentPath + "/login.jsp");
            return false;
        }
        return true;
    }

}
```

性能拦截器

```
package com.estore4.interceptor;


import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.log4j.Logger;
import org.springframework.util.StopWatch;
import org.springframework.web.servlet.HandlerInterceptor;
import org.springframework.web.servlet.ModelAndView;


public class PerformanceMonitorInterceptor implements HandlerInterceptor {
    ThreadLocal<StopWatch> stopWatchLocal = new ThreadLocal<StopWatch>();
    Logger logger = Logger.getLogger("customLogger"); //log.properties配置的记录


    public boolean preHandle(HttpServletRequest request,
HttpServletResponse response, Object handler)
            throws Exception {
        StopWatch stopWatch = new StopWatch(handler.toString());
        stopWatch.start(handler.toString());
        stopWatchLocal.set(stopWatch);
        logger.info("Accessing URL path: " + getURLPath(request));
        logger.info("Request processing started on: " + getCurrentTime());
        return true;
    }

    public void postHandle(HttpServletRequest arg0,
HttpServletResponse response, Object handler,
            ModelAndView modelAndView) throws Exception {
        logger.info("Request processing ended on " + getCurrentTime());
    }

    public void afterCompletion(HttpServletRequest request,
HttpServletResponse response, Object handler,
            Exception exception) throws Exception {
        StopWatch stopWatch = stopWatchLocal.get();
        stopWatch.stop();
        logger.info("Total time taken for processing: " +
stopWatch.getTotalTimeMillis() + " ms");
        stopWatchLocal.set(null);

    logger.info("==============================================
=========");
    }

    private String getURLPath(HttpServletRequest request) {
        String currentPath = request.getRequestURI();
        String queryString = request.getQueryString();
        queryString = queryString == null ? "" : "?" + queryString;
        return currentPath + queryString;
    }

    private String getCurrentTime() {
        DateFormat  formatter  =  new  SimpleDateFormat("yyyy-MM-dd
hh:mm:ss");
        Calendar calendar = Calendar.getInstance();
        calendar.setTimeInMillis(System.currentTimeMillis());
        return formatter.format(calendar.getTime());
    }
}
```

配置拦截器（框架配置文件）



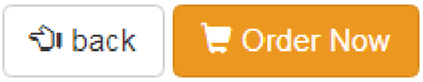
```
    <mvc:interceptors>
        <mvc:interceptor>
            <mvc:mapping path="/**" />
            <mvc:exclude-mapping path="/products.do" />
            <mvc:exclude-mapping path="/products/product.do" />
            <mvc:exclude-mapping path="/loginAction/login.do" />
            <mvc:exclude-mapping path="/loginAction/logout.do" />
            <bean class="com.estore4.interceptor.AuditingInterceptor">
            </bean>
        </mvc:interceptor>
        <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
            <property name="paramName" value="language" />
        </bean>
        <bean class="com.estore4.interceptor.PerformanceMonitorInterceptor" />
    </mvc:interceptors>
```

Spring MVC5 –注册、加入购物车

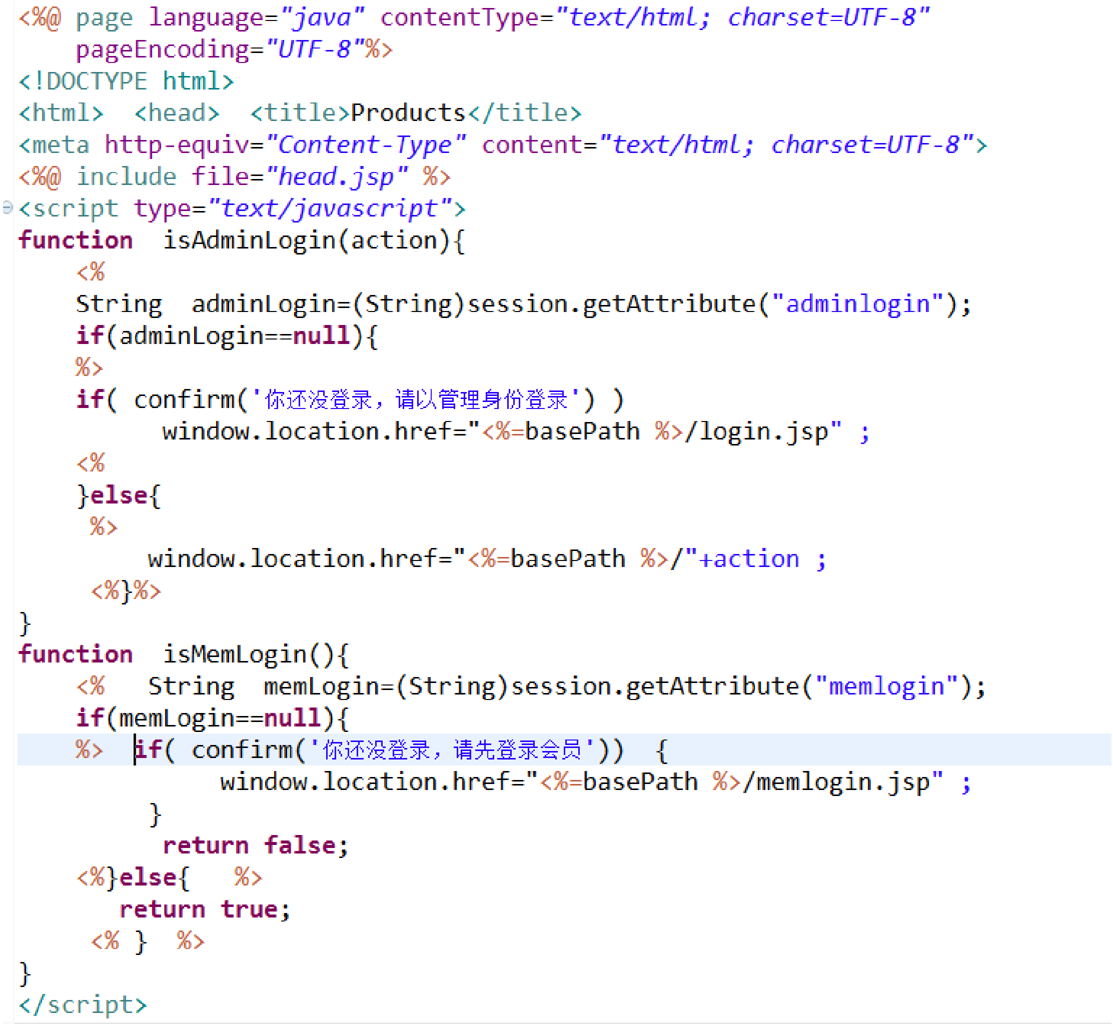
创建工程 estore5

将 estore4 相关文件复制过来，包名称中 estore4 修改为 estore5,能正常运行，进行下一步

实现选购商品



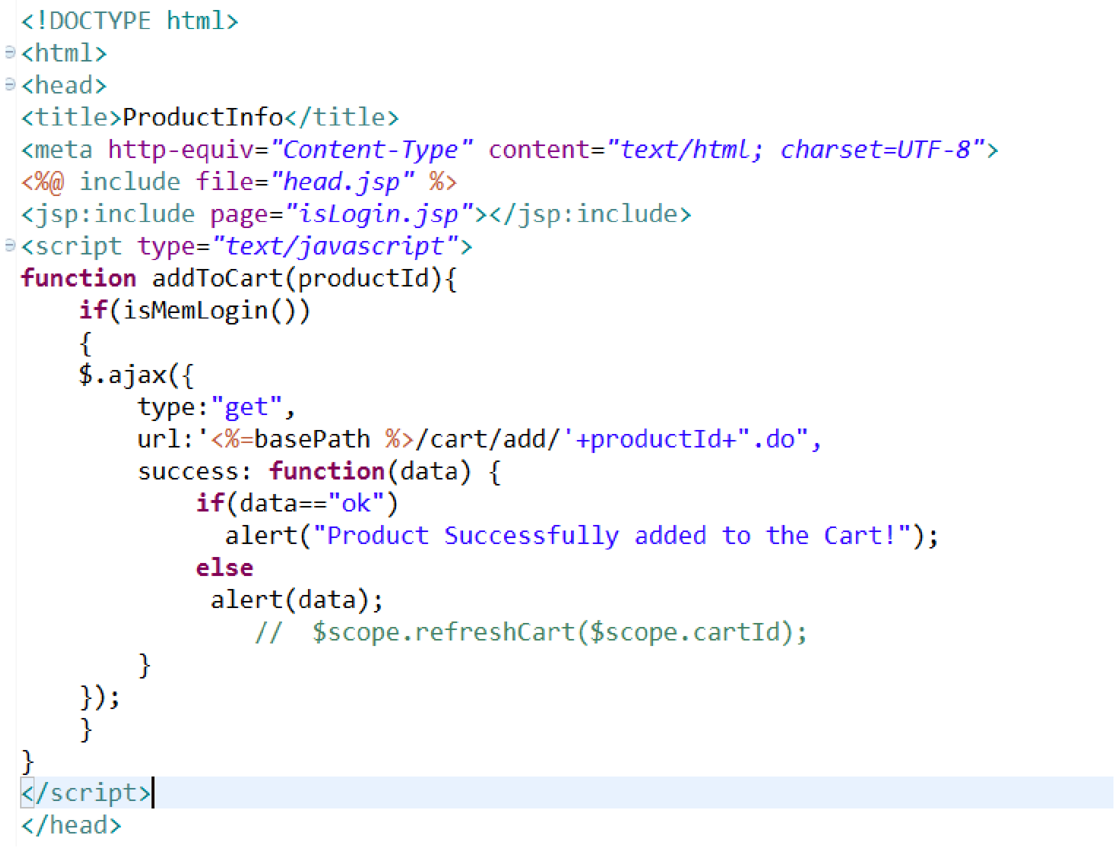
创建 isLogin.jsp 文件



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>  <head>  <title>Products</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<%@ include file="head.jsp" %>
<script type="text/javascript">
function  isAdminLogin(action){
    <%
    String  adminLogin=(String)session.getAttribute("adminlogin");
    if(adminLogin==null){
    %>
    if( confirm('你还没登录，请以管理身份登录') )
          window.location.href="<%=basePath %>/login.jsp" ;
    <%
    }else{
     %>
         window.location.href="<%=basePath %>/"+action ;
     <%}%>
}
function  isMemLogin(){
    <%   String  memLogin=(String)session.getAttribute("memlogin");
    if(memLogin==null){
    %>  if( confirm('你还没登录，请先登录会员'))  {
              window.location.href="<%=basePath %>/memlogin.jsp" ;
         }
          return false;
    <%}else{   %>
       return true;
     <% }  %>
}
</script>
```

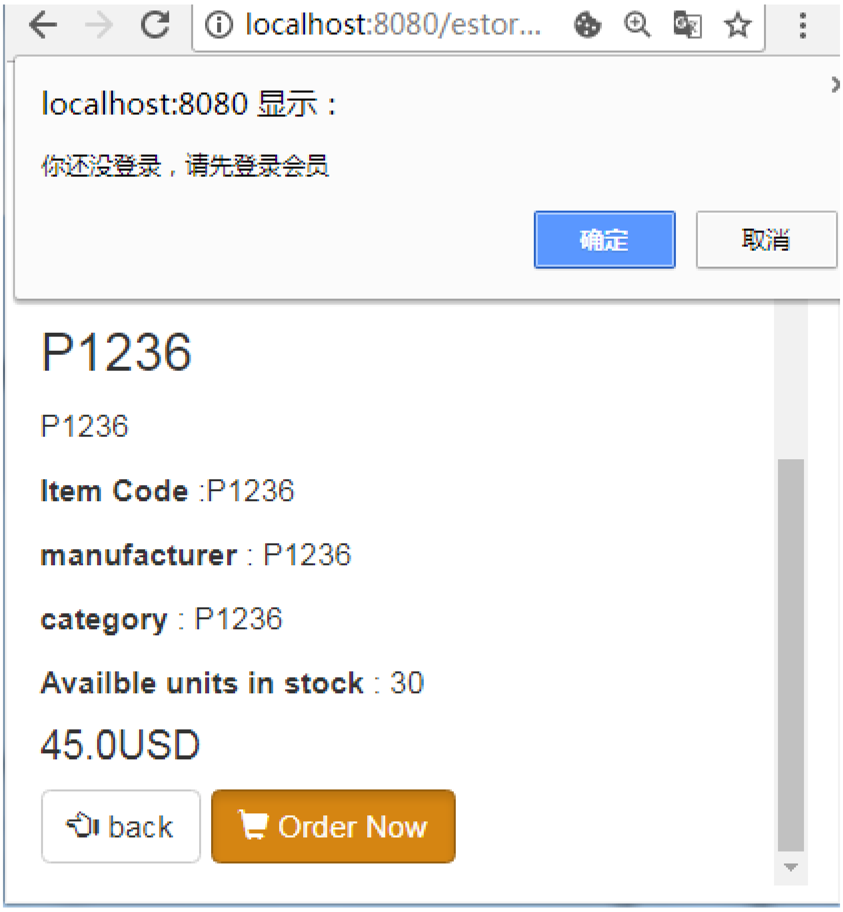
编辑 product.jsp

```
<a href="javascript:void(0)" class="btn btn-warning
btn-large"
onclick="addToCart('${product.productId}')" >
                <span  class="glyphicon-shopping-cart
glyphicon"></span>  Order Now
               </a>
```



```
<!DOCTYPE html>
<html>
<head>
<title>ProductInfo</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<%@ include file="head.jsp" %>
<jsp:include page="isLogin.jsp"></jsp:include>
<script type="text/javascript">
function addToCart(productId){
    if(isMemLogin())
    {
    $.ajax({
        type:"get",
        url:'<%=basePath %>/cart/add/'+productId+".do",
        success: function(data) {
            if(data=="ok")
              alert("Product Successfully added to the Cart!");
            else
             alert(data);
                //  $scope.refreshCart($scope.cartId);
        }
    });
    }
}
</script>
</head>
```

测试是否要登录



localhost:8080 显示：

你还没登录，请先登录会员

P1236

P1236

Item Code :P1236

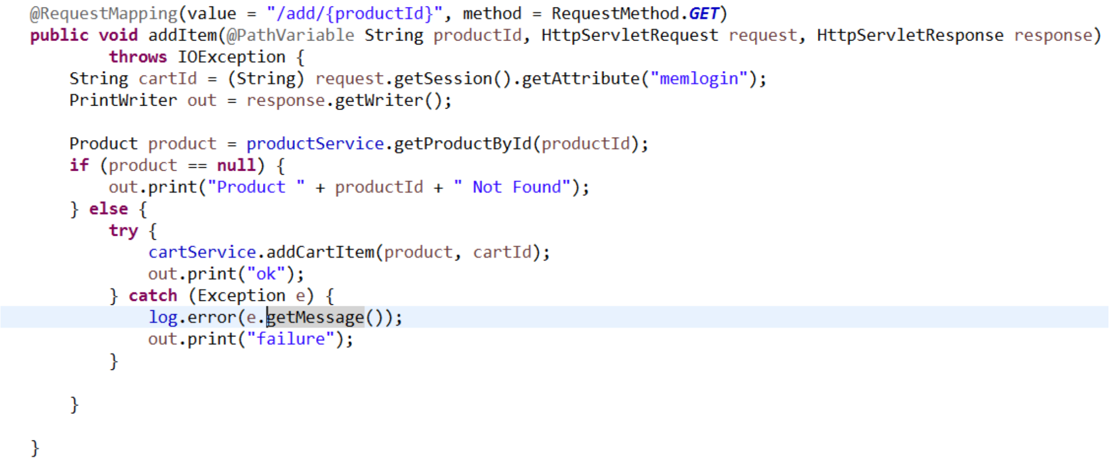
manufacturer : P1236

category : P1236

Availble units in stock : 30

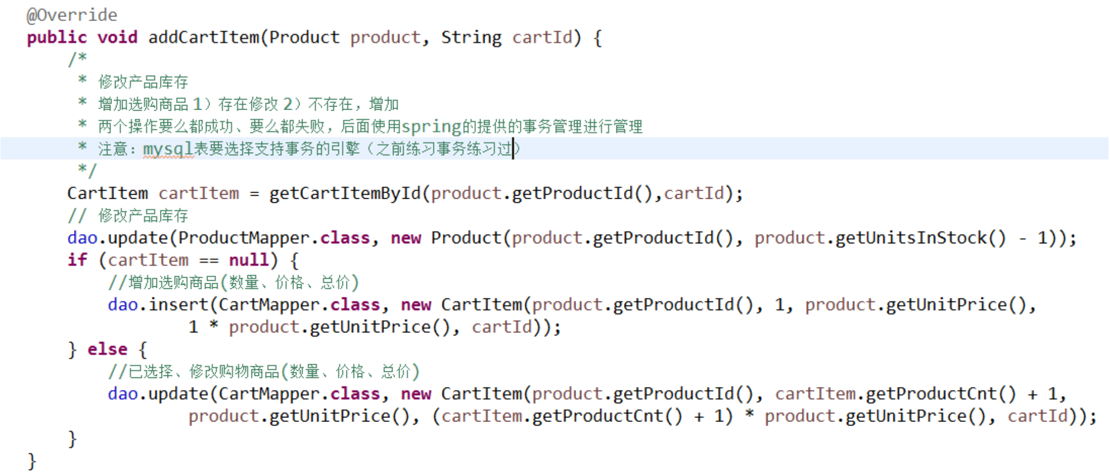
45.0USD

实现订购 action（控制器/业务控制器）



```
@RequestMapping(value = "/add/{productId}", method = RequestMethod.GET)
public void addItem(@PathVariable String productId, HttpServletRequest request, HttpServletResponse response)
        throws IOException {
    String cartId = (String) request.getSession().getAttribute("memlogin");
    PrintWriter out = response.getWriter();

    Product product = productService.getProductById(productId);
    if (product == null) {
        out.print("Product " + productId + " Not Found");
    } else {
        try {
            cartService.addCartItem(product, cartId);
            out.print("ok");
        } catch (Exception e) {
            log.error(e.getMessage());
            out.print("failure");
        }

    }

}
```

实现订购 service(业务逻辑器)



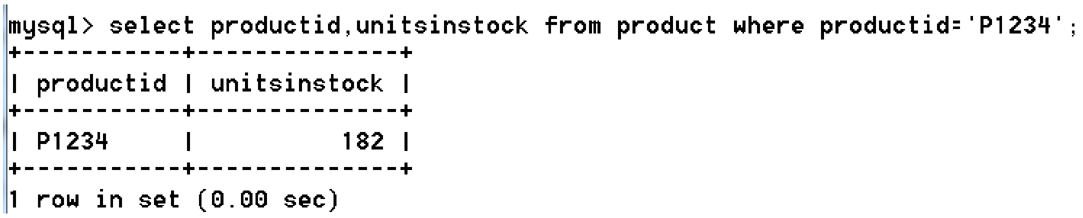
```
@Override
public void addCartItem(Product product, String cartId) {
    /*
     * 修改产品库存
     * 增加选购商品 1）存在修改 2）不存在，增加
     * 两个操作要么都成功、要么都失败，后面使用spring的提供的事务管理进行管理
     * 注意：mysql表要选择支持事务的引擎（之前练习事务练习过）
     */
    CartItem cartItem = getCartItemById(product.getProductId(),cartId);
    // 修改产品库存
    dao.update(ProductMapper.class, new Product(product.getProductId(), product.getUnitsInStock() - 1));
    if (cartItem == null) {
        //增加选购商品(数量、价格、总价)
        dao.insert(CartMapper.class, new CartItem(product.getProductId(), 1, product.getUnitPrice(),
                1 * product.getUnitPrice(), cartId));
    } else {
        //已选择、修改购物商品(数量、价格、总价)
        dao.update(CartMapper.class, new CartItem(product.getProductId(), cartItem.getProductCnt() + 1,
                product.getUnitPrice(), (cartItem.getProductCnt() + 1) * product.getUnitPrice(), cartId));
    }
}
```

数据库层

每个人根据选购商品表自己完成

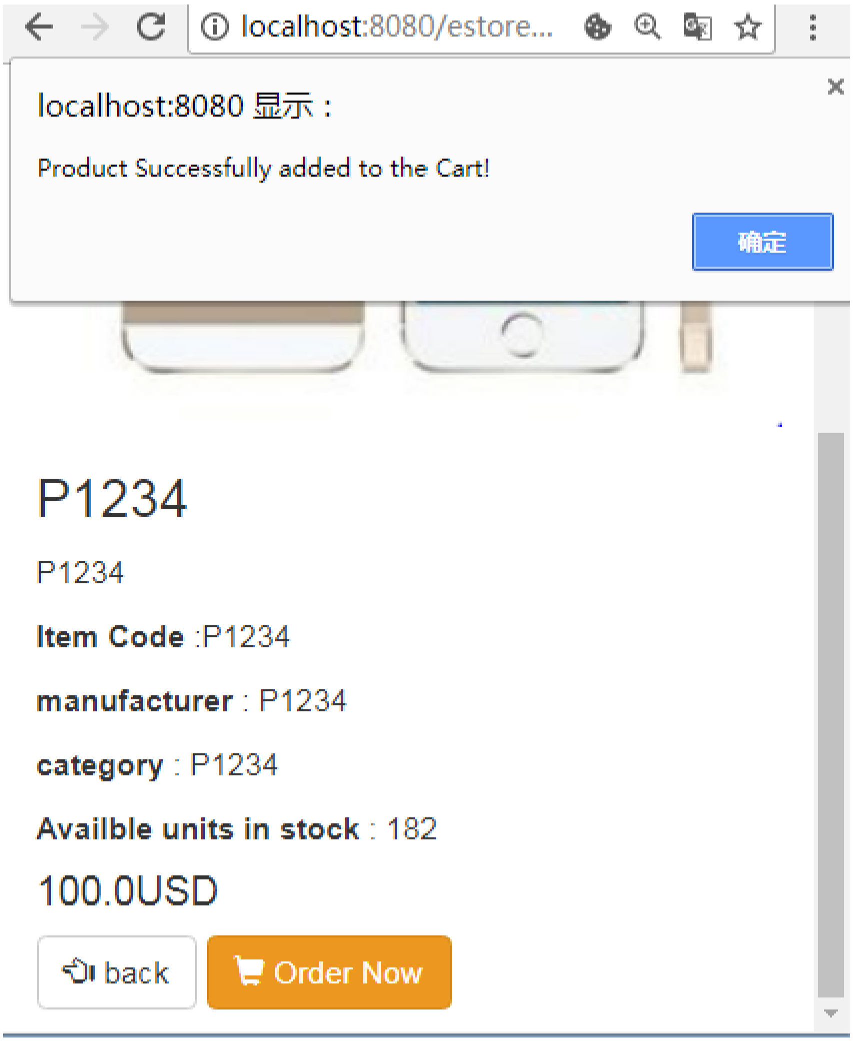
运行效果

商品选购前



```
mysql> select productid,unitsinstock from product where productid='P1234';
+-----------+--------------+
| productid | unitsinstock |
+-----------+--------------+
| P1234     |          182 |
+-----------+--------------+
1 row in set (0.00 sec)
```

选购商品



localhost:8080 显示：

Product Successfully added to the Cart!

P1234

P1234

Item Code :P1234

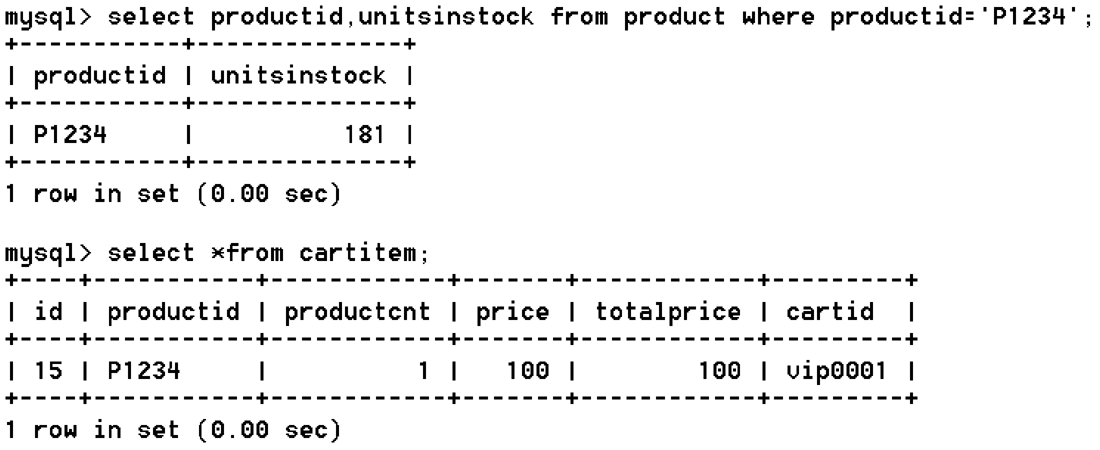
manufacturer : P1234

category : P1234

Availble units in stock : 182

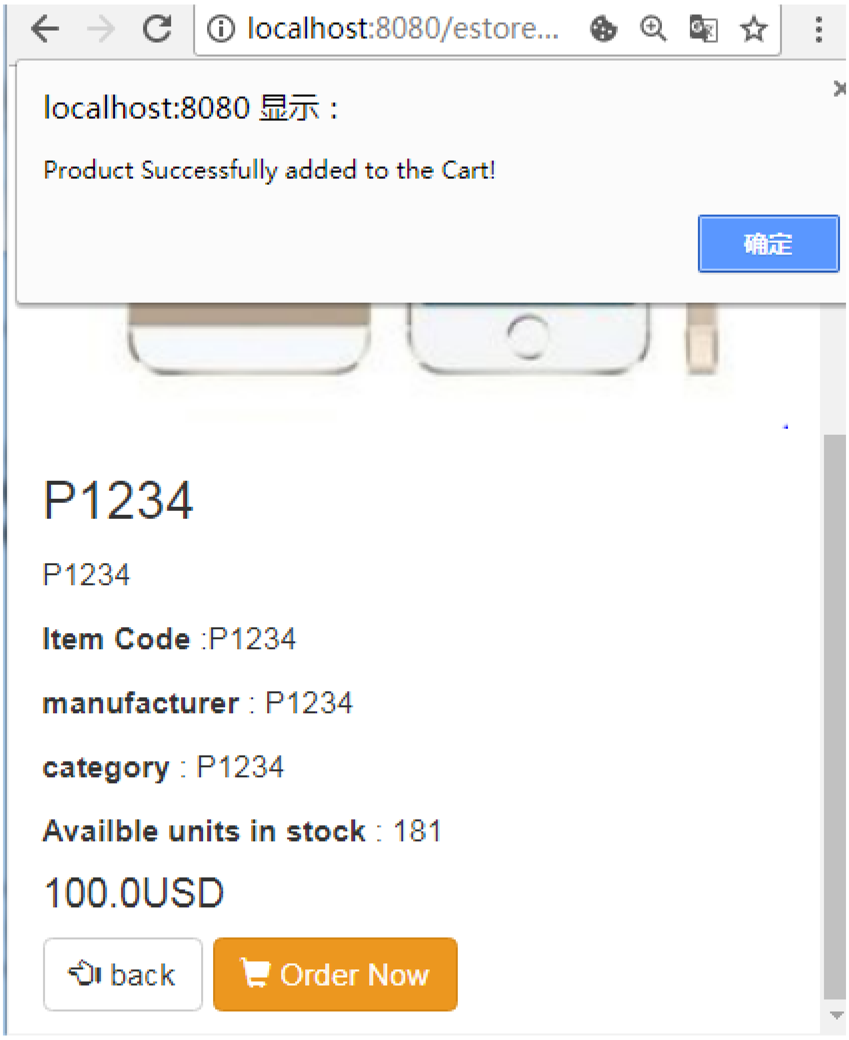
100.0USD

商品选购后



```
mysql> select productid,unitsinstock from product where productid='P1234';
+-----------+--------------+
| productid | unitsinstock |
+-----------+--------------+
| P1234     |          181 |
+-----------+--------------+
1 row in set (0.00 sec)

mysql> select *from cartitem;
+----+-----------+------------+-------+------------+---------+
| id | productid | productcnt | price | totalprice | cartid  |
+----+-----------+------------+-------+------------+---------+
| 15 | P1234     |          1 |   100 |        100 | vip0001 |
+----+-----------+------------+-------+------------+---------+
1 row in set (0.00 sec)
```

再次选购



localhost:8080 显示：

Product Successfully added to the Cart!

P1234

P1234

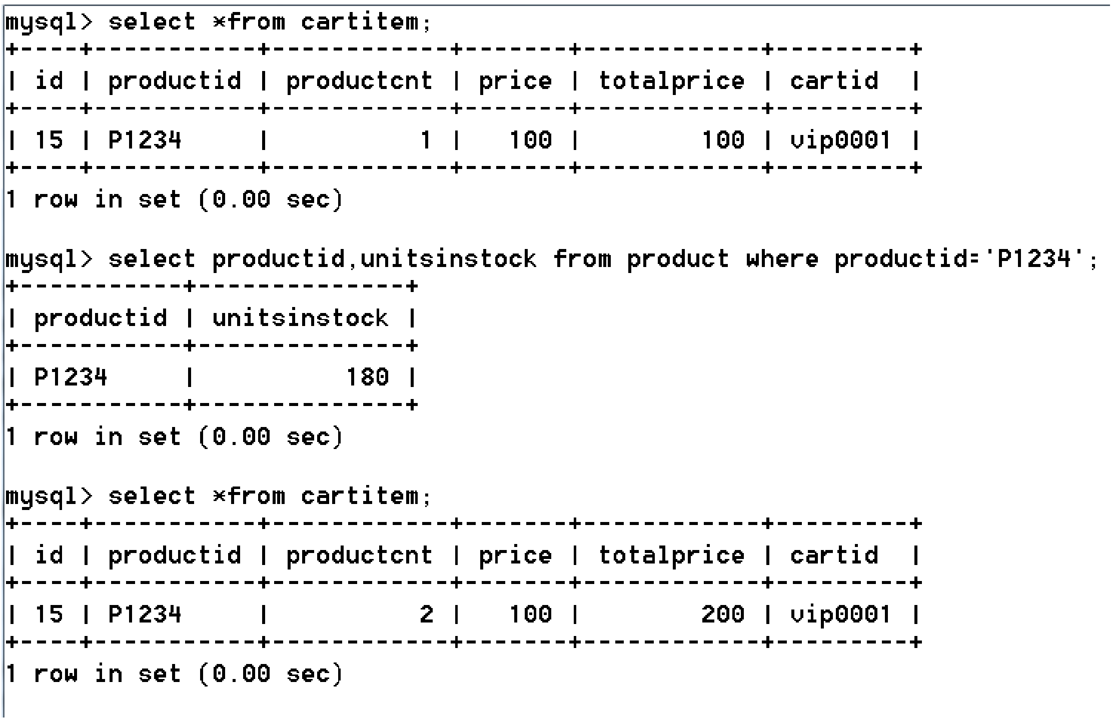
Item Code :P1234

manufacturer : P1234

category : P1234

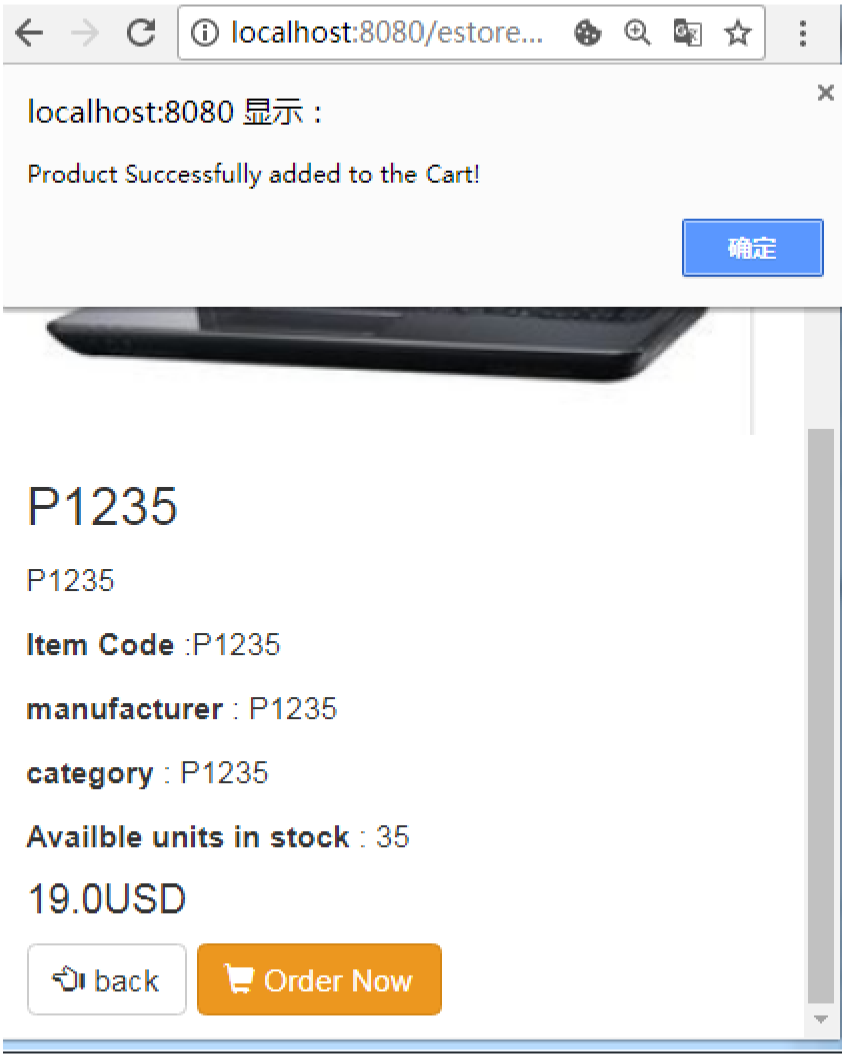
Availble units in stock : 181

100.0USD



```
mysql> select *from cartitem;
+----+-----------+------------+-------+------------+---------+
| id | productid | productcnt | price | totalprice | cartid  |
+----+-----------+------------+-------+------------+---------+
| 15 | P1234     |          1 |   100 |        100 | vip0001 |
+----+-----------+------------+-------+------------+---------+
1 row in set (0.00 sec)

mysql> select productid,unitsinstock from product where productid='P1234';
+-----------+--------------+
| productid | unitsinstock |
+-----------+--------------+
| P1234     |          180 |
+-----------+--------------+
1 row in set (0.00 sec)

mysql> select *from cartitem;
+----+-----------+------------+-------+------------+---------+
| id | productid | productcnt | price | totalprice | cartid  |
+----+-----------+------------+-------+------------+---------+
| 15 | P1234     |          2 |   100 |        200 | vip0001 |
+----+-----------+------------+-------+------------+---------+
1 row in set (0.00 sec)
```

选购其他商品



localhost:8080 显示：

Product Successfully added to the Cart!

P1235

P1235

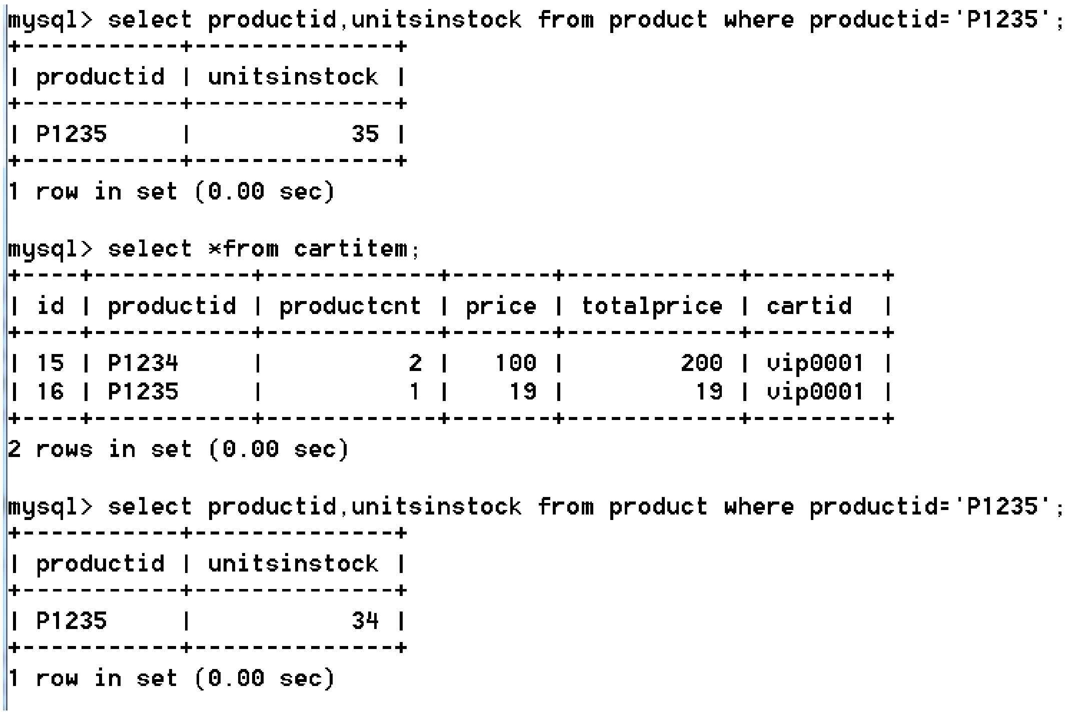
Item Code :P1235

manufacturer : P1235

category : P1235

Availble units in stock : 35

19.0USD



```
mysql> select productid,unitsinstock from product where productid='P1235';
+-----------+--------------+
| productid | unitsinstock |
+-----------+--------------+
| P1235     |           35 |
+-----------+--------------+
1 row in set (0.00 sec)

mysql> select *from cartitem;
+----+-----------+------------+-------+------------+---------+
| id | productid | productcnt | price | totalprice | cartid  |
+----+-----------+------------+-------+------------+---------+
| 15 | P1234     |          2 |   100 |        200 | vip0001 |
| 16 | P1235     |          1 |    19 |         19 | vip0001 |
+----+-----------+------------+-------+------------+---------+
2 rows in set (0.00 sec)

mysql> select productid,unitsinstock from product where productid='P1235';
+-----------+--------------+
| productid | unitsinstock |
+-----------+--------------+
| P1235     |           34 |
+-----------+--------------+
1 row in set (0.00 sec)
```

再重新打开浏览器，使用其他会员登录，选购商品看数据库变化