

# Web应用开发 之 会话管理

赵小敏

浙江工业大学计算机科学与技术学院

# 会话管理

- 4.1 会话管理
- 4.2 使用会话实现购物车
- 4.3 Cookie 及其应用
- 4.4 URL 重写与隐藏表单域

# 4.1会话管理

- 4.1.1 理解状态与会话
- 4.1.2 会话管理机制
- 4.1.3 HttpSession API
- 4.1.4 使用HttpSession对象
- 4.1.5 会话超时与失效
- 4.1.6 会话示例
- 4.1.7 session和servletContext的比较

## 4.1.1 理解状态与会话

- 如何区分身份？

✓ 日常生活中办理业务  身份证

✓ 网络中的计算机  IP地址

✓ 一台机器中的进程  端口号

✓ 系统中的用户  帐号

- 问题：web系统中如何区分用户的操作呢？

 客户状态跟踪  会话机制

## 4.1.1 理解状态与会话

- 在很多情况下，Web服务器必须能够跟踪客户的状态。
- 跟踪客户状态可以使用数据库实现，但在Servlet容器中通常使用会话机制维护客户状态。

## 4.1.1 理解状态与会话

- 协议记住客户及其请求的能力称为状态。按这个观点可将协议分成两种类型：
  - 有状态的
  - 无状态的
- HTTP协议是一种无状态的协议，服务器对客户的每个请求和响应都是作为一个分离的事务对待。
- 服务器本身不能在多个请求中维护客户的状态。

## 4.1.1 理解状态与会话

- 有些Web应用不需要服务器记住客户，例如，在线查询系统就不需要维护客户的状态。
- 某些应用中客户与服务器的交互中就需要有状态的。典型的例子是[购物车应用](#)。
- 为实现购物应用，服务器必须跟踪所有的[请求](#)并把它们与客户关联。

## 4.1.1 理解状态与会话

- 会话 (session) 是客户与服务器之间的不间断的请求-响应序列。
- 当客户向服务器发送第一个请求时就开始一个会话。对该客户之后的每个请求，服务器能够识别出请求来自于同一个客户。
- 当客户明确结束会话或服务器在一个预定义的时限内没从客户接收任何请求时，会话就结束了。当会话结束后，服务器就忘记了客户以及客户的请求。



## 4.1.2 会话管理机制

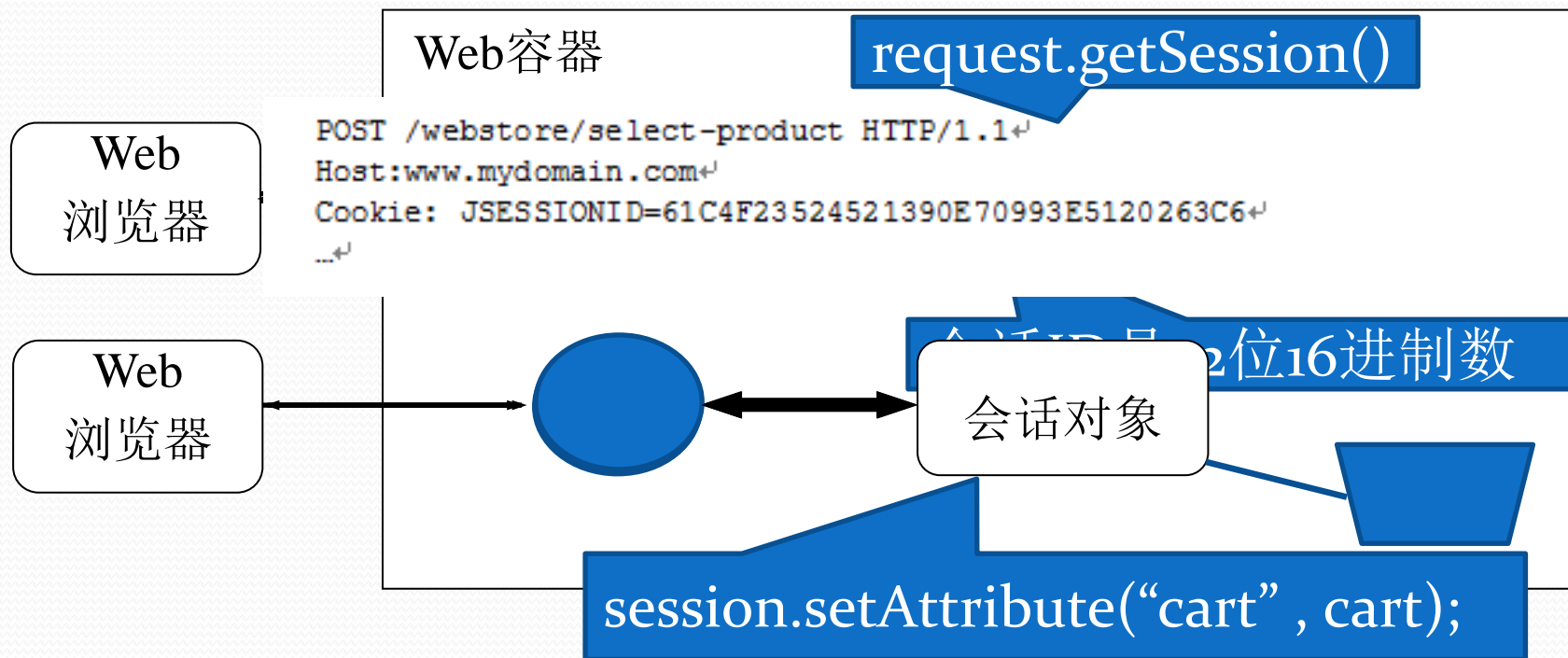
- 通过会话机制可以实现购物车应用。当用户登录购物网站时，Web容器就为客户创建一个HttpSession对象。
  - ① 实现购物车的Servlet使用该会话对象存储用户的购物车对象，购物车中存储着用户购买的商品列表。
  - ② 当客户向购物车中添加商品或删除商品时，Servlet就更新该列表。当客户要结账时，Servlet就从会话中检索购物车对象，从购物车中检索商品列表并计算总价格。
  - ③ 一旦客户结算完成，容器就会关闭会话。

## 4.1.2 会话管理机制

- 不能使用客户的IP地址唯一标识客户的原因：
  - ✓ 客户可能是通过局域网访问Internet。
  - ✓ 尽管在局域网中每个客户有一个IP地址，但对于服务器来说，客户的实际IP地址是路由器的IP地址，所以该局域网的所有客户的IP地址都相同！因此也就无法唯一标识客户。
  - ✓ 可以伪造客户端的IP。

## 4.1.2 会话管理机制

- Web容器使用HttpSession表示会话对象。容器管理会话的过程如下图所示。



## 4.1.3 HttpSession API

- HttpSession接口中定义的常用方法：
- **public String getId()**: 返回为该会话指定的唯一标识符，它是一个32位的十六进制数。
- **public long getCreationTime()**: 返回会话创建的时间。时间为从1970年1月1日午夜到现在的毫秒数。
- **public long getLastAccessedTime()**: 返回会话最后被访问的时间。
- **public boolean isNew()**: 如果会话对象还没有同客户关联，则返回true。

## 4.1.3 HttpSession API

- **public ServletContext getServletContext():** 返回该会话所属的ServletContext对象。
- **public void setAttribute (String name, Object value):** 将一个指定名称和值的属性存储到会话对象上。
- **public Object getAttribute(String name):** 返回存储到会话上的指定名称的属性值，如果没有指定名称的属性，则返回null。
- **public Enumeration getAttributeNames():** 返回存储在会话上的所有属性名的一个枚举对象。

## 4.1.3 HttpSession API

- **public void setMaxInactiveInterval(int interval)**: 设置在容器使该会话失效前客户的两个请求之间最大间隔的时间，单位为秒。参数为负值表示会话永不失效。
- **public int getMaxInactiveInterval()**: 返回以秒为单位的最大间隔时间，在这段时间内，容器将在客户请求之间保持该会话打开状态。
- **public void invalidate()**: 使会话对象失效并删除存储在其上的任何对象。

## 4.1.4 使用HttpSession对象

➤ 通常需要三步：

- 1) 创建或返回与客户请求关联的会话对象。
- 2) 在会话对象中添加或删除“名/值”对属性。
- 3) 如果需要可使会话失效。

➤ 创建或返回会话使用request对象的getSession()方法

- HttpSession getSession(boolean create)
- HttpSession getSession()

## 4.1.5 会话超时与失效

- 可以在web.xml文件中设置会话超时时间。

```
<session-config>
```

```
    <session-timeout>30</session-timeout>
```

```
</session-config>
```

- <session-timeout>元素中指定的以分钟为单位的超时期限。0或小于0的值表示会话永不过期。
- 如果用户在指定期间内没有执行任何动作，服务器就认为用户处于不活动状态并使会话对象无效，默认情况下是30分钟

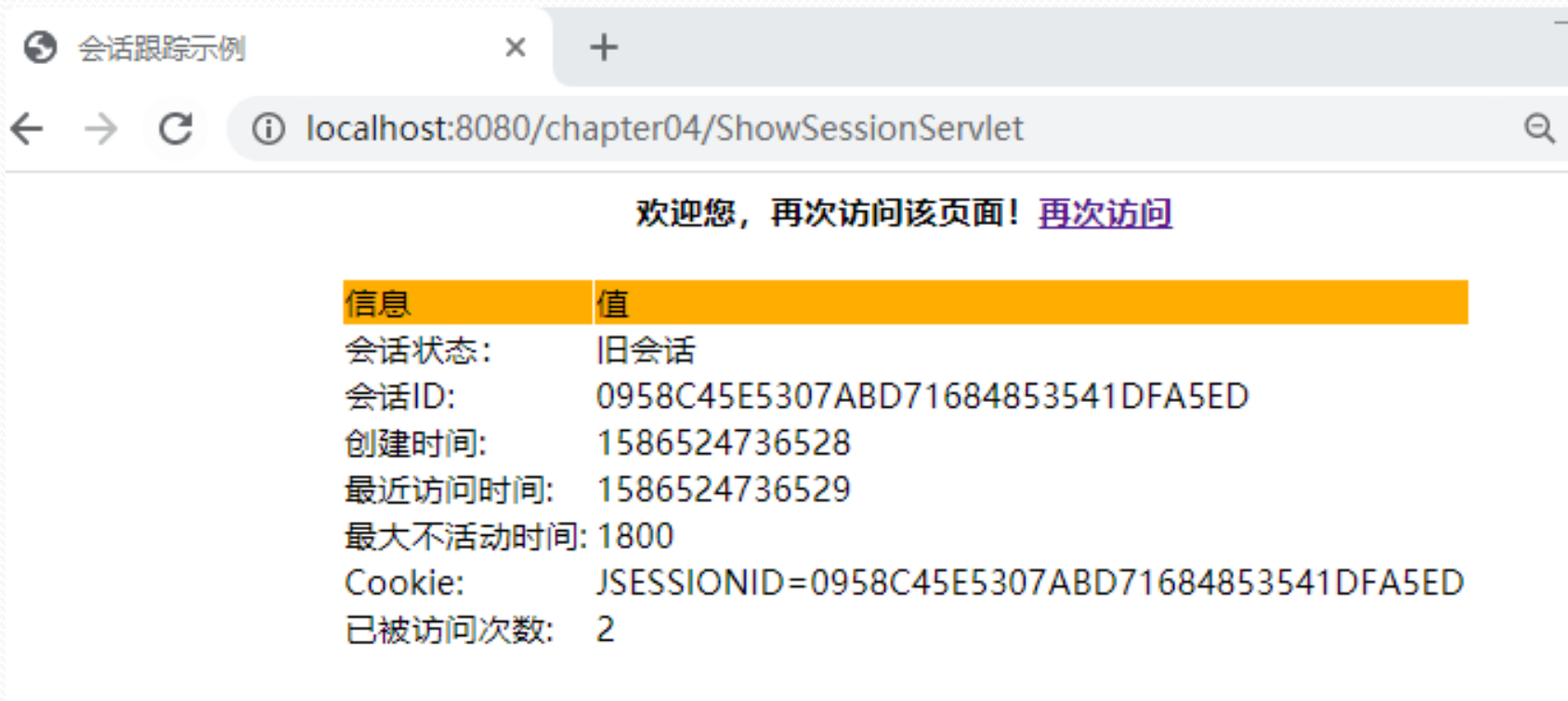


## 4.1.5 会话超时与失效

- 在web.xml文件中设置的会话超时时间针对Web应用程序中的所有会话对象，但有时可能需要对特定的会话对象指定超时时间，可使用会话对象的 `setMaxInactiveInterval()`。
- 使用 `HttpSession` 接口的 `invalidate()` 结束会话。例如，在购物车的应用中，我们希望在用户付款处理完成后结束会话。这样，当客户再次发送请求时，就会创建一个购物车中不包含商品的新的会话。

## 4.4.6会话示例

### 1、会话跟踪示例



会话跟踪示例

localhost:8080/chapter04/ShowSessionServlet

欢迎您，再次访问该页面! [再次访问](#)

信息	值
会话状态:	旧会话
会话ID:	0958C45E5307ABD71684853541DFA5ED
创建时间:	1586524736528
最近访问时间:	1586524736529
最大不活动时间:	1800
Cookie:	JSESSIONID=0958C45E5307ABD71684853541DFA5ED
已被访问次数:	2

## 4.1.6 会话示例

### 2、猜数游戏

- ① 当使用GET请求访问它时，生成一个在0~100之间的随机整数，将其作为一个属性存储到用户的会话对象中，同时提供一个表单供用户输入猜测的数。
- ② 如果该Servlet接收到一个POST请求，它将比较用户猜的数和随机生成的数是否相等，若相等在响应页面中给出信息，否则，应该告诉用户猜的数是大还是小，并允许用户重新猜。

localhost:8080/chapter04/GuessNumberServlet

我想出一个0到100之间的数，请你猜！

确定

localhost:8080/chapter04/GuessNumberServlet

太小了! 请重猜!

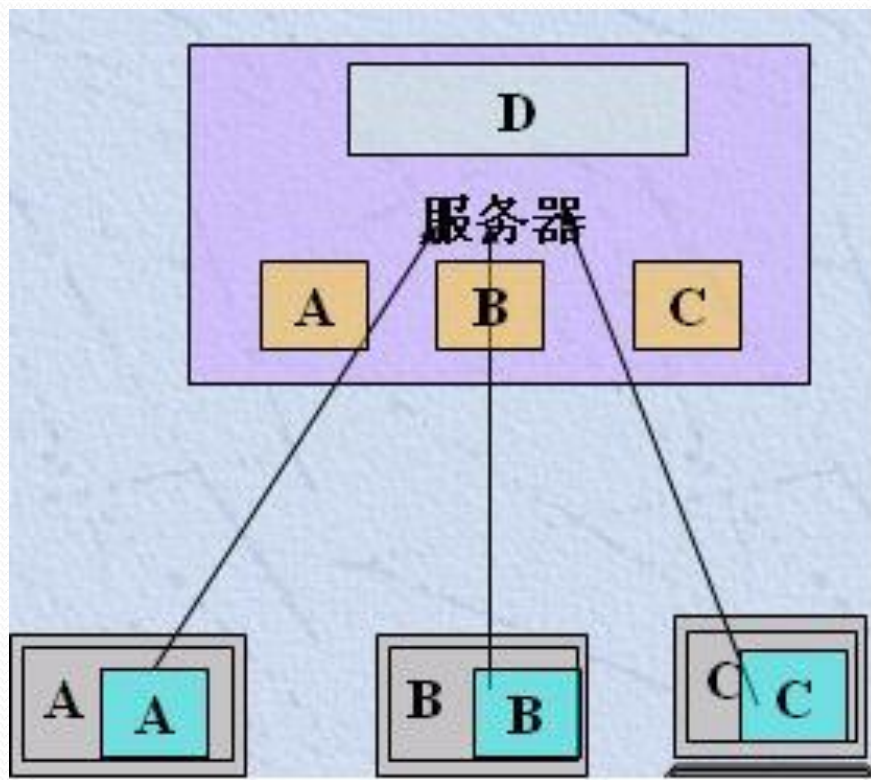
确定

## 4.1.6会话示例

- 3、对已登陆的用户，可以下载文件，未登陆的用户则无法下载
- 登陆页面：login.jsp
- 登陆处理：LoginServlet.java
- 下载文件：FileDownloadServlet.java

## 4.1.7 session和servletContext的比较

- session是一次会话过程，是局部的
- servletContext可以理解为服务器端的一个共享空间，它可以被所有的客户端访问，是全局的



A、B、C是代表客户端的cookie，服务器端的A、B、C代表session，D代表servletContext

## 例：用session实现页面访问次数

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, java.io.IOException {
    response.setContentType(" text/html");
    HttpSession session = request.getSession();
    String heading;
    Integer accessCount =(Integer)session.getAttribute(" accessCount");
    if(accessCount == null) {
        accessCount = new Integer(1);
        heading = " Welcom,You are first time to visit!";
    }
    else {
        heading = " Welcome Backer";
        accessCount = new Integer(accessCount.intValue()+1);
    }
    session.setAttribute(" accessCount",accessCount);
    PrintWriter out = response.getWriter();
    out.println(" The title:"+heading);
    out.println(" Access count: "+accessCount);
}
```

结果：不同用户访问，  
都会重新计数

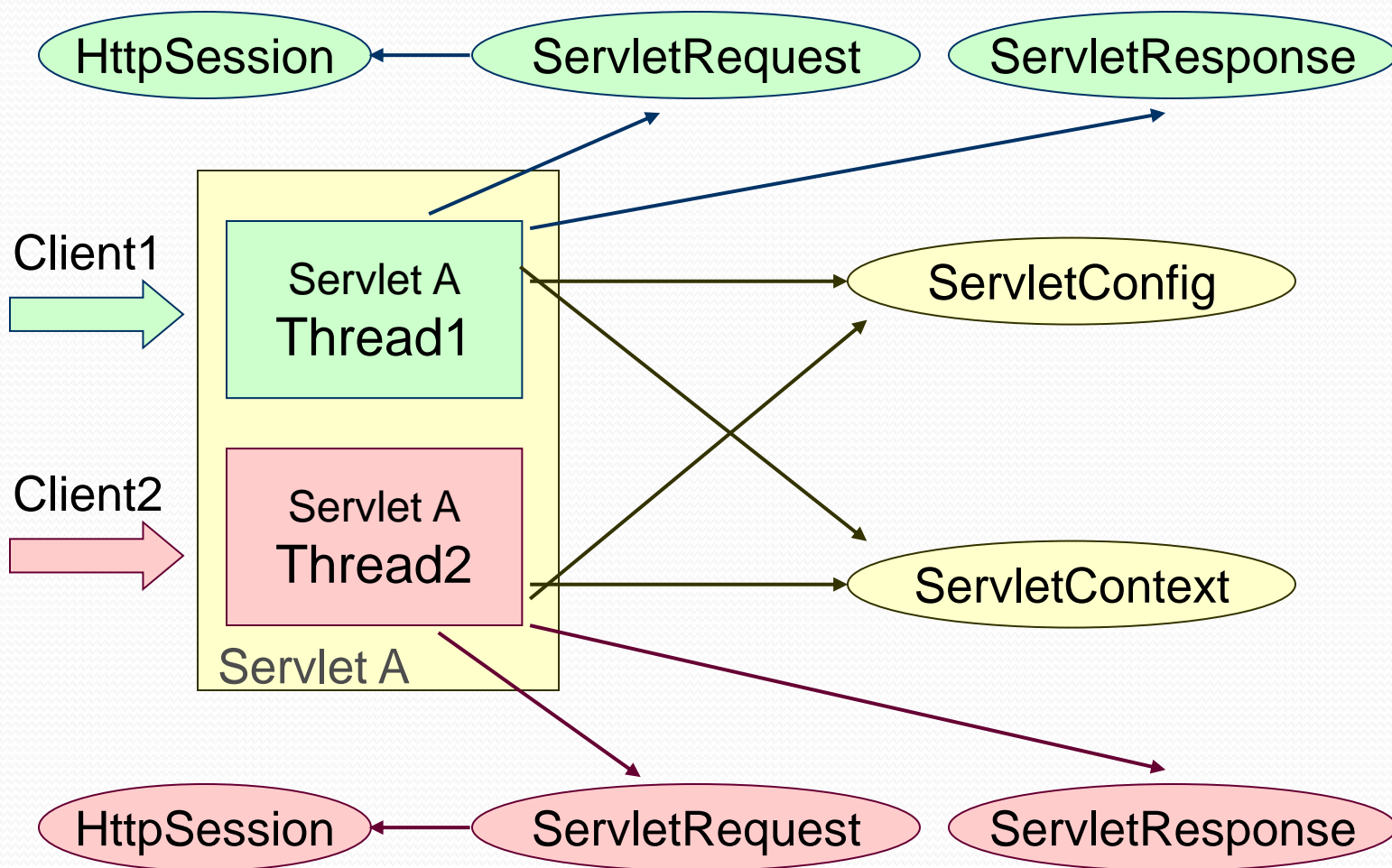
## 例：用ServletContext实现页面访问次数

```
public void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, java.io.IOException {
    ServletContext servletContext = this.getServletContext();
    Integer accessCount =(Integer)servletContext.getAttribute(" accessCount");
    if(accessCount == null){
        accessCount = 0;
    }else{
        accessCount = accessCount + 1;
    }
    servletContext.setAttribute(" accessCount", accessCount);
    response.setCharacterEncoding("gb2312");
    PrintWriter out = response.getWriter();
    out.println("<html><body><h2>访问:"+accessCount+"次</body></html> ");
}
```

结果：不同用户访问，不会重新计数  
会累加计数

# 进一步理解会话 (1)

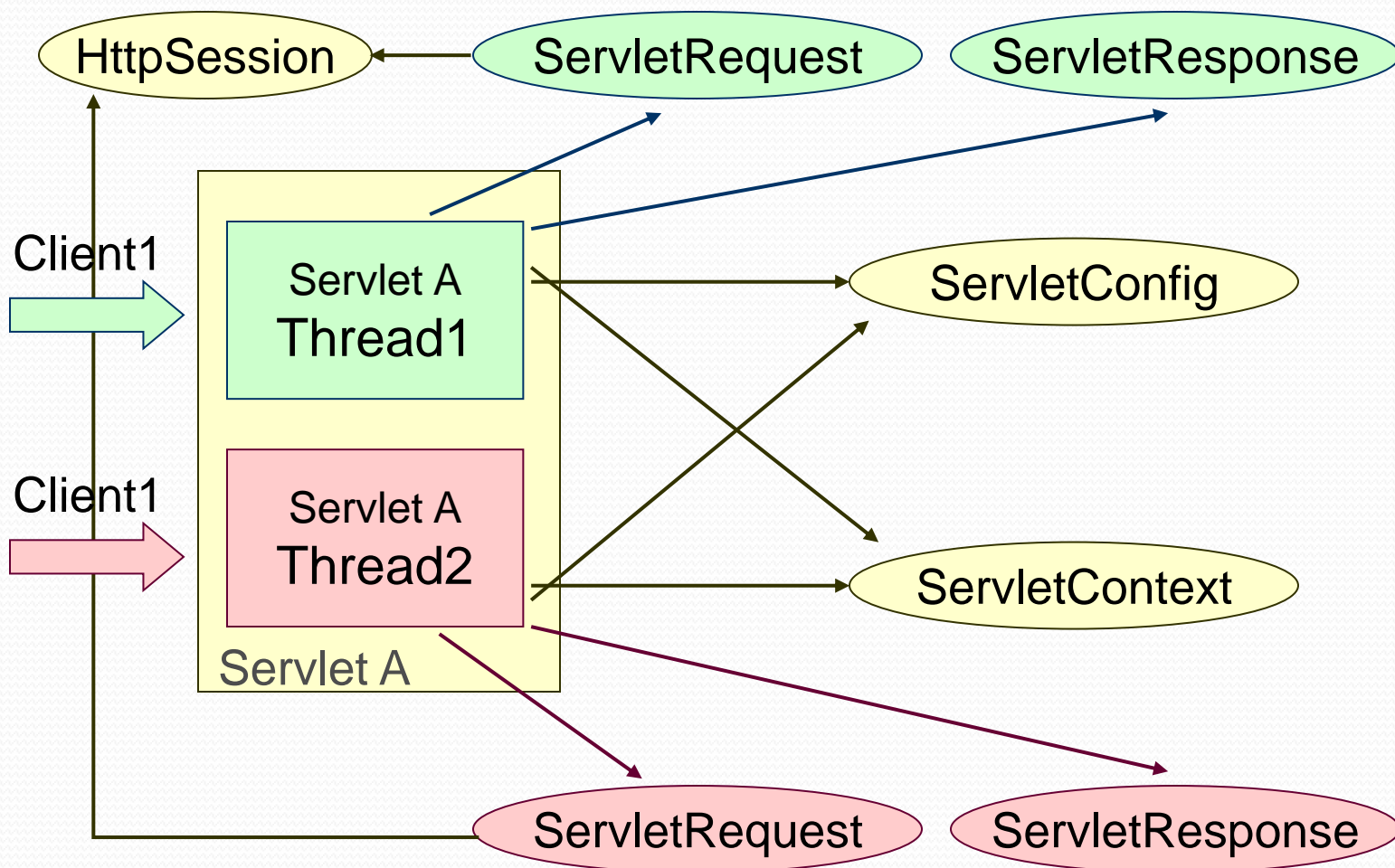
- Servlet对象





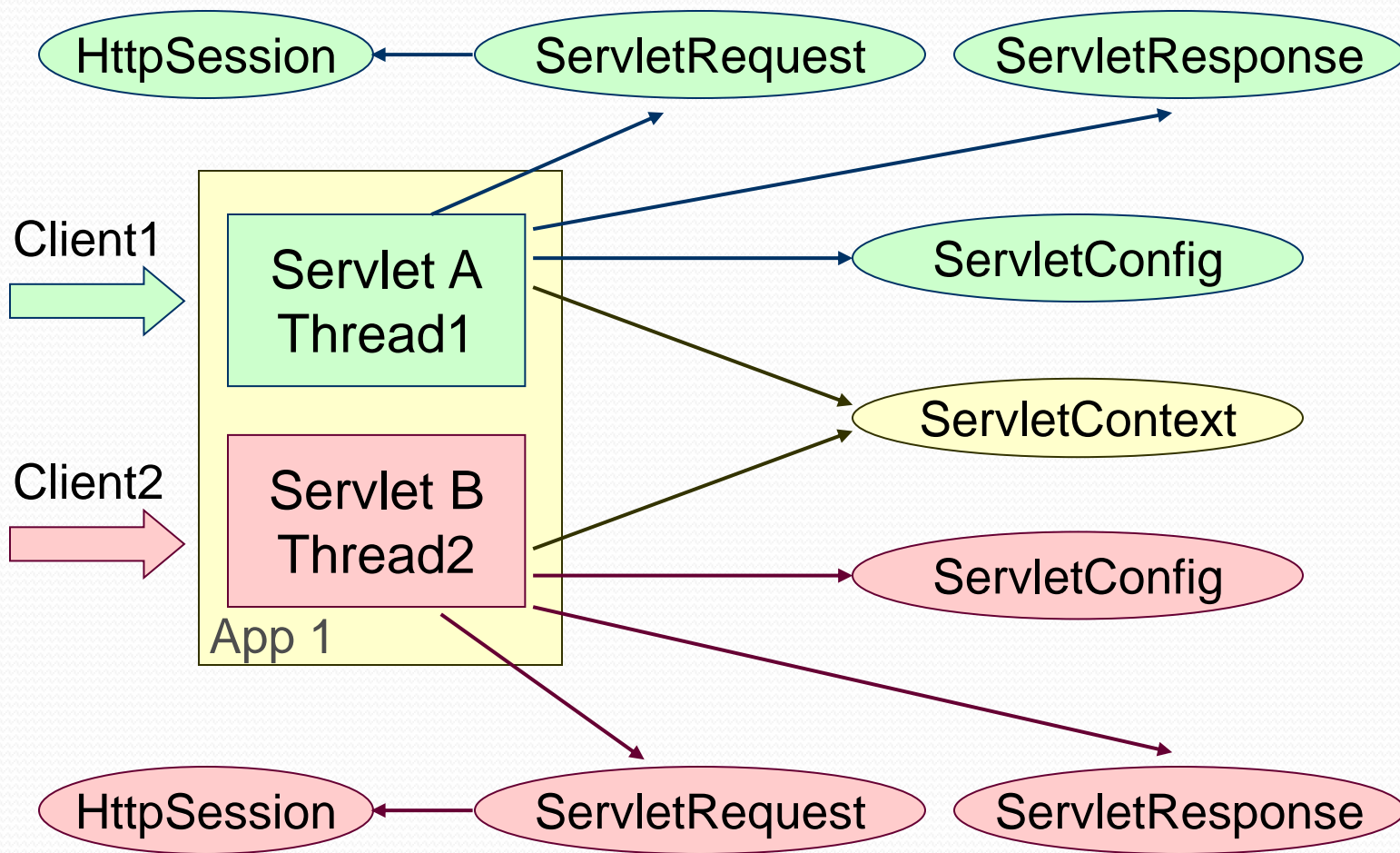
# 进一步理解会话 (2)

- Servlet对象



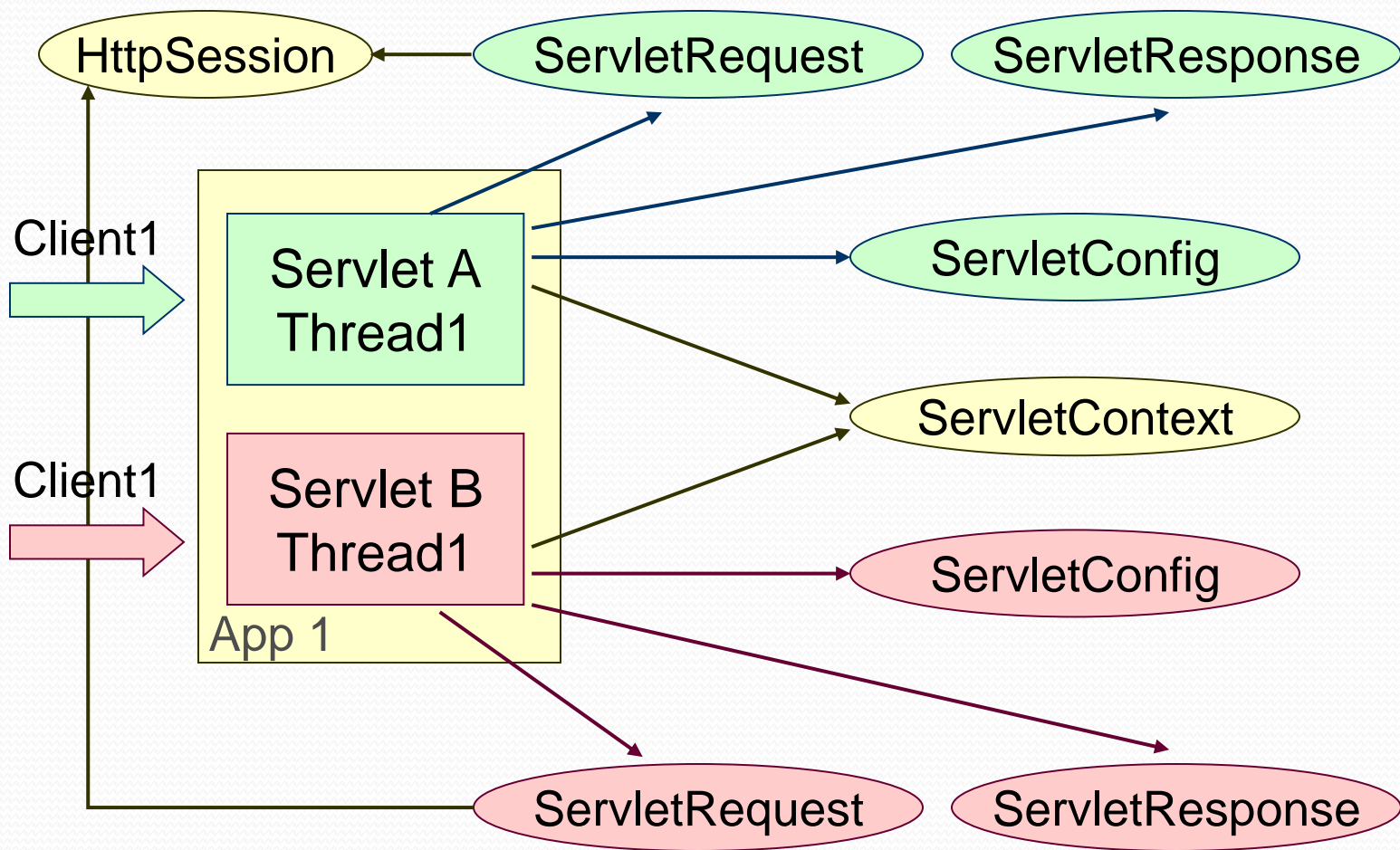
# 进一步理解会话 (3)

- Servlet对象



# 进一步理解会话 (4)

- Servlet对象



## 4.2使用会话实现购物车

- 1 模型类设计
- 2 购物车类设计
- 3 视图设计
- 4 控制器设计

# 1、模型类设计

➤ 商品类 **Product**，它表示商品信息。

- **private int id;** // 商品编号
- **private String pname;** // 商品名称
- **private double price;** // 商品价格
- **private int stock;** // 商品库存量
- **private String type;** // 商品类别

➤ **GoodsItem**类，用来表示购物车中一种商品。

- **private Product product;** // 商品信息
- **private int quantity;** // 商品数量

## 2、购物车类设计

- 购物车是购物系统中最重要的类，它用来临时存放用户购买的商品（Product）信息，购物车对象将被存储到用户的会话对象中。
- ShoppingCart类，包括添加商品、删除一件商品、计算购物车所有商品价格、清空购物车等方法

# 3、视图设计

➤ 本示例视图共包含3个JSP页面，

- [index.jsp](#)用于显示商品信息；
- [showProduct.jsp](#)用于显示一件商品详细信息；
- [showCart.jsp](#)用于显示购物车中商品信息，并提供删除一件商品功能。

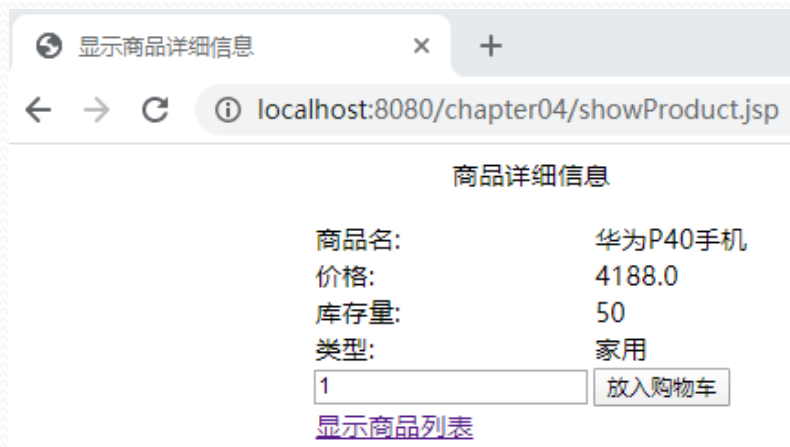


购物系统首页

localhost:8080/chapter04/index.jsp

商品号	商品名	价格	库存量	类型	详细信息
101	单反相机	4199.99	10	家用	<a href="#">详细信息</a>
102	华为P40手机	4188.0	50	家用	<a href="#">详细信息</a>
103	笔记本电脑	5299.99	30	电子	<a href="#">详细信息</a>
104	平板电脑	2900.0	20	电子	<a href="#">详细信息</a>

[查看购物车](#)



显示商品详细信息

localhost:8080/chapter04/showProduct.jsp

商品名:	华为P40手机
价格:	4188.0
库存量:	50
类型:	家用
<input type="text" value="1"/>	<input type="button" value="放入购物车"/>

[显示商品列表](#)



用户购物车信息

localhost:8080/chapter04/showCart.jsp

商品	价格	数量	小计	是否删除
华为P40手机	4188.0	2	8376.0	<a href="#">删除</a>
笔记本电脑	5299.99	1	5299.99	<a href="#">删除</a>
平板电脑	2900.0	1	2900.0	<a href="#">删除</a>
总计:16575.99				

[返回继续购物](#)

# 4、控制器设计

- 使用Servlet作为控制器，ControllerServlet类主要处理商品显示、查看购物车、添加商品到购物车等动作。
- 控制器主要处理3个动作
  - 查看商品详细信息 (/showProduct)
  - 从购物车中删除一件商品 (/deleteItem) ， 这两个动作使用doGet()方法处理。
  - 向购物车中添加商品动作 (/addToCart) 由doPost()方法处理。



# 如何存储商品信息

- 使用**ArrayList**对象设计一个上下文监听器的类存放商品信息，该对象在应用程序启动时创建。
- 设计一个读写文件的类读取存储在文件中的商品信息
- 设计一个读写数据库的类读取存储在数据库中的商品信息

# 小 结

- 通过使用HttpSession对象可以跟踪客户与服务器的交互，Web应用程序需要在本来无状态的HTTP协议上实现状态。
- 服务器通过为其指定一个唯一的标识符实现一个会话。

# 作业

- 1、开发一个下载文件WEB-INF/files/servlet.pdf功能的FileDownloadServlet：要求只有登录用户才能下载该文件，若用户没有登录，将请求转发到登录页面login.html，用户输入用户名和密码后，控制转到LoginServlet，用户名和密码分别为admin和12345678a，如果用户合法再把控制转到FileDownloadServlet下载文件。

# 作业

- **2、系统登陆功能。**在某教务系统中，系统会根据用户类型(教师/学生)登录后显示不同的页面。为模拟该功能，需要定义设计如下页面或类：

(1)登录页面login.jsp: 包括用户类型名（默认为学生类型）、用户名(用学号或工号)、用户密码，用javaScript判断用户名和密码不能为空，否则给出提示信息；

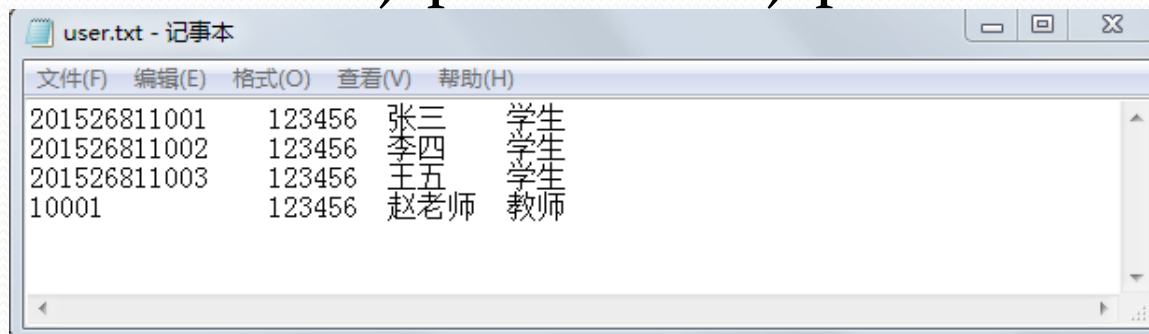
(2)用户登录处理页面LoginActionServlet, 根据用户类型判断，如果是学生用户，且用户名和密码正确，则把用户名写入到session中，同时转发到student.jsp页面输出“×××同学，欢迎您！”，否则页面跳转回到login.jsp；如果是教师用户，且用户名和密码正确，则把用户名写入到session中，同时转发到teacher.jsp页面输出“×××老师，欢迎您！”，否则页面跳转回到login.jsp。

(3)学生用户页面student.jsp: 若session存在，则输出“×××同学，欢迎您！”，否则页面返回到login.jsp

(4)教师用户页面teacher.jsp: 若session存在，则输出“×××老师，欢迎您！”，否则页面返回到login.jsp

# 作业

- 注：学生和教师的信息包括学号或工号、密码、姓名和用户类型等都保存在WEB-INF/user.txt文件中，每个用户信息一行，信息之间用制表符(\t)隔开,如下图所示。在LoginActionServlet判断用户名和密码时，需与user.txt文件中学生学号或教师工号和密码一致，若成功，则创建UserBean对象（至少有学号或工号、密码、姓名和用户类型等属性）保存至session，然后跳转至student.jsp或teacher.jsp。

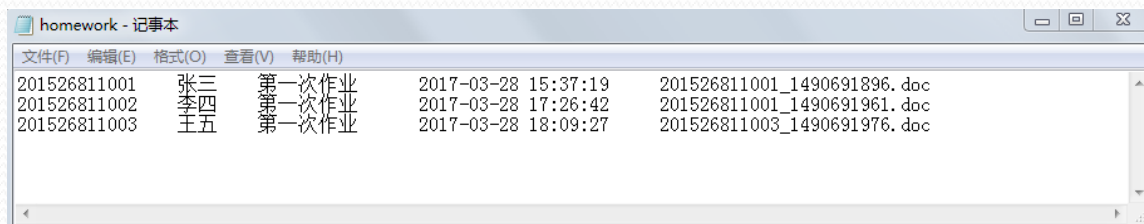


# 作业

- 3、编写一个简单的作业上传与下载系统。具体要求如下：

(1) 学生通过登录页面login.jsp进行登录，用户登录处理页面为LoginActionServlet，登录名和密码根据保存在WEB-INF/user.txt（具体内容与第2题一致）的学号和密码进行判断。登录成功后将学生用户对象存入session，并跳转至操作页面studentOperation.jsp，该页面包括上传作业、查看所有作业和我的作业等三个功能，具体以超链接形式。

(2) 学生通过上传作业页面uploadHomework.jsp将作业以文件形式上传，先判断用户是否登录（判断session是否存在），若未登录，则跳转至登录页面。上传时要求把作业标题和作业以附件形式上传，上传成功后将学号、姓名、作业标题和上传时间以及作业文件名（学号\_当前时间戳的值命名文件，其中当前时间戳的值=new java.util.Date().getTime()/1000）存入homework.txt文件中，如下图：



文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
201526811001	张三	第一次作业	2017-03-28 15:37:19	201526811001_1490691896.doc
201526811002	李四	第一次作业	2017-03-28 17:26:42	201526811001_1490691961.doc
201526811003	王五	第一次作业	2017-03-28 18:09:27	201526811003_1490691976.doc

# 作业

(3) 学生查看所有作业页面AllHomeworkServlet（映射为homework.html）时，从homework.txt文件中读出所有已上传的作业并显示，见下图。



序号	学号	姓名	作业标题	上传时间	下载
1	201526811001	张三	第一次作业	2017-03-28 15:37:19	<a href="#">下载</a>
2	201526811002	李四	第一次作业	2017-03-28 17:26:42	<a href="#">下载</a>
3	201526811003	王五	第一次作业	2017-03-28 18:09:27	<a href="#">下载</a>

[返回操作页面](#)

(4) 我的作业页面MyHomeworkServlet仅显示当前学生自己的作业。

(5) 教师通过登录页面login.jsp进行登录，用户登录处理页面为LoginActionServlet，登录名和密码根据保存在user.txt（具体内容如下）的学号 and 密码进行判断。登录成功后将教师用户对象存入session，并跳转至操作页面AllHomeworkServlet（映射为homework.html）查看所有学生的作业。



# 作业

(6) 定义一个UserBean对象（至少有学号或工号、姓名和用户类型等属性），用户登录成功后可保存至session进行数据共享。

- 要求：学生用户点击下载链接时，若该作业是自己上传的，则允许下载，若不是自己的作业，则提示“您没有权限下载该作业”，而教师可以下载所有的学生作业。
- 提示：下载的超连接为  
downloadHomework.do?studentid=201526811001&filename=201526811001\_1490691896.doc，在下载超链接的DownloadHomeworkServlet页面（映射地址为downloadHomework.do）判断当前用户类型，如果是教师，则运行下载，如果是学生，则判断该文件是否与当前学生学号studentid匹配，不匹配则不允许下载。



# 作业

- 4（选做）、修改会话实现购物车的例子，要求商品信息存在文件中，可扩充商品信息（如商品介绍、图片），美化页面。

**Thank You !**