

TTC□ □ □ TTC

课程/项目名称	Javaweb	授课教师	赵千里	总学时：60学时
教学目的和要求	通过课程掌握Servlet&Jsp规范及API，熟悉MVC编程思想，熟悉HTTP规范。			
教学重点、难点	重点： servlet，jsp，过滤器，监听器，http协议 难点： cookie、session、Filter、Listener			
教学资源	教材及参考资料： 教材： 内部课件			
	其他教学资源：			
教学环境				

章节	第1章web应用开发入门
----	--------------

教学目标 与要求	了解web起源，web应用的历史；学习web应用三大标准：url统一资源定位符、http协议、html超文本标记语言；了解web服务器概念；了解JAVAE概念及容器概念，熟悉Servlet和Jsp规范、Servlet容器概念；掌握tomcat的安装和配置以及运行。
重点 难点	重点：url、http、tomcat的安装配置 难点：http协议请求及响应的组成

## 【教学进程安排】

Web应用介绍、Url、Http.html

4课时

JavaWeb应用、Tomcat

4课时

## 【课程主要内容】

### 1. WEB应用介绍，URL、HTTP、HTML

#### 1.1. 什么是web应用

说一下分布式系统概念以及C/S和B/S架构的各自特点。紧接着打开Chrome浏览器，在地址栏输入任意常用网站地址，引入url、http、html的概念，此处需要打开chrome调试器配合讲解，说明地址栏中输入的是url，调试器监视任意请求简要说一下http协议的内容，调试器查看任意页面的源码，简要阐述html的作用。至此说清楚web应用的大体概念和核心三标准。

#### 1.2. URL统一资源定位符

根据之前访问的地址栏地址，说明url的作用：资源定位，说明uri统一资源标识符的作用：描述资源，url是uri的子集，不仅可以描述资源，更重要的是定位资源，通过ppt中的案例解释url的各个组成部分，其中说明下http协议默认端口80的情况下，url中可省略端口号。

大致说明下绝对路径和相对路径的概念，在文件系统中建立1.html,2.html,sub/3.html三个文件，用于解释相对路径。



#### 1.3. http协议（参考简书<https://www.jianshu.com/p/80e25cb1d81a>，参考ietf rfc2612：<https://tools.ietf.org/html/rfc2616> abstract部分）

HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写，是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。

HTTP是一个基于TCP/IP通信协议来传递数据（HTML文件，图片文件，查询结果等）。

HTTP是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990年提出，经过几年的使用与发展，得到不断地完善和扩展。目前在WWW中使用的是HTTP/1.0的第六版，HTTP/1.1的规范化工作正在进行之中，而且HTTP-NG(Next Generation of HTTP)的建议已经提出。

HTTP协议工作于客户端-服务端架构为上。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。Web服务器根据接收到的请求后，向客户端发送响应信息。

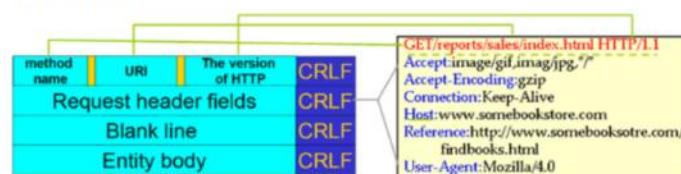
##### 1.3.1. http协议组成（参考rfc2616第1.4、4、5、6部分）

直接参考rfc2612 1.4对于http的定义：

The HTTP protocol is a request/response protocol. A client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity metainformation, and possible entity-body content.

参考4、5、6部分来说明http由请求和响应组成，以及请求和响应的分别组成部分，结合ppt中下图直观说明request和response的组成部分。

#### • HTTP请求格式



考核点	• 1
学员问题汇总	
作业	
课后 总结分析	

??????

章节	第2章servlet开发技术
教学目标 与要求	掌握javaweb标准目录结构、理解servlet基本原理，掌握servlet api的基本使用；ServletRequest和ServletResponse接口几个主要方法的使用，继承GenericServlet和HttpServlet实现servlet输出hello world的功能。掌握servlet生命周期和生命周期相关方法。
重点 难点	重点： servlet生命周期、request和response对象的使用 难点： servlet生命周期

## 【教学进程安排】

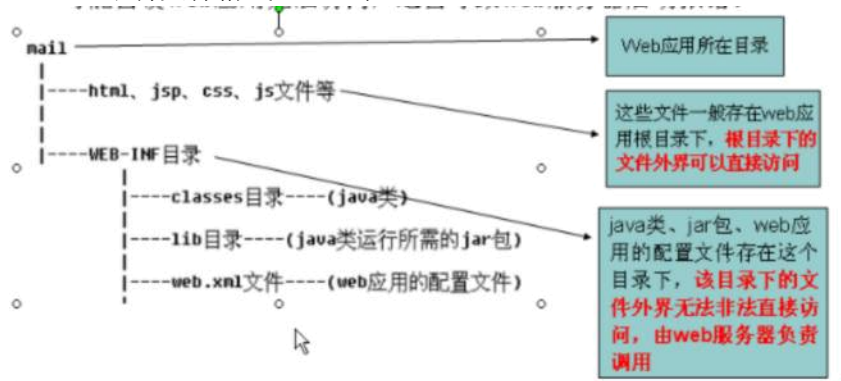
JavaWeb应用标准目录结构及Servlet介绍	2课时
Servlet生命周期详解	1课时
Servlet的其他知识点	1课时
HttpServlet	1课时

## 【课程主要内容】

### 3. JavaWeb、Servlet 简介

#### 3.1. JavaWeb目录结构

根据下图讲解一个标准javaweb应用的目录组成部分，着重讲解web-inf目录下的几个重要目录和文件：classes、lib、web.xml。在tomcat webapps目录下按照目录结构创建对应的文件夹和文件，以及web.xml文件，web.xml文件暂时不解释，先从examples中将文件格式copy过来。



#### 3.2. Servlet api

在带领学生编写第一个servlet之前，带着学生访问<https://javaee.github.io/javaee-spec/javadocs/>，通过查询官方doc来带领学生理解servlet的api及作用。

3.2.1. 通读Servlet接口的介绍，从中了解servlet接口的作用，实现servlet接口的最佳实践，关于生命周期的三个方法。以及其他几个方法，比如getServletConfig。

3.2.2. 通过doc了解Servlet最佳实践：继承GenericServlet或者HttpServlet，进入GenericServlet，然后通读GenericServlet的介绍，了解到继承service方法来实现具体处理请求的业务逻辑。

#### 3.2.3. 实现第一个Servlet

在之前tomcat webapps底下建立好的javaweb项目目录中web-inf/classes目录下用编辑器建立带领学生实现一个继承了GenericServlet并且覆盖了service方法的Servlet,并且输出“hello world”，代码如下：

```
import java.io.*;
import javax.servlet.*;

public class HelloServlet extends GenericServlet{

    public void service(ServletRequest req,ServletResponse res) throws
    ServletException, IOException{
        OutputStream out = res.getOutputStream();
        out.write("Hello World".getBytes());
    }
}
```

考核点	P
学员问题汇总	
作业	
课后 总结分析	

??????

章节	第三章 Servlet ServletContext、Session
教学目标 与要求	掌握ServletContext对象的使用，理解ServletContext域的含义 掌握cookie与session技术，了解cookie，session的生命周期，其他会话跟踪技术 掌握请求重定向与请求转发技术
重点 难点	重点： cookie与session的生命周期， ServletContext域 难点： cookie与session配合使用， ServletContext域

## 【教学进程安排】

ServletContext 2课时

Cookie技术 3课时

Session技术 3课时

## 【课程主要内容】

# 7. ServletContext对象

## 7.1. ServletContext介绍

讲解servletContext对象，讲解一下通过servletContext对象的几个方法，着重讲：[getInitParameter?](#) [getMimeType?](#) [getRealPath?](#) [getResourceAsStream?](#) [getResource?](#) [getResource?](#) [removeAttribute?](#) [setAttribute?](#) [getRequestDispatcher?](#)

## 7.2. Context-param运用

讲解一下其中的方法，顺带讲一下context-param标签，这里重新设计一个需求：试着用servlet按照/WEB-INF/web.xml的路径读取受保护的web.xml（学生练习），同时要说明context-param的作用，以spring举例子，后实现一个代码，读取context-param中jdbc.properties的位置，然后使用properties读取这个配置文件，然后拿到jdbc的参数（[jdbc.properties.note](#)），这里与getClassLoader().[getResourceAsStream](#)做一下对比，主要说明二者的不同，从面向对象的思想上去说，一个代表类加载器，一个代表web运行环境，因此二者获得的根目录位置自然不同。

## 7.3. ServletContext域

说明需求，存放服务的名称，通过servletContext域进行存储。

## 7.4. 文件下载实现

使用response完成一个文件下载任务，content-disposition详解，参考（<https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Headers/Content-Disposition>）

# 8. Cookie

## 8.1. Cookie介绍

直接打开rfc6265（<https://tools.ietf.org/html/rfc6265#section-3.1>），带着学生泛读一下abstract和Introduction部分，了解cookie的概念和作用：

### （1）简介

Cookie机制采用的是在客户端保持 HTTP 状态信息的方案。

Cookie实际上是一小段的文本信息。客户端请求服务器，如果服务器需要记录该用户状态，就使用response向客户端浏览器颁发一个Cookie。客户端浏览器会把Cookie保存起来。当浏览器再请求该网站时，浏览器把请求的网址连同该Cookie一同提交给服务器。服务器检查该Cookie，以此来辨认用户状态。

### （2）作用

Cookie的根本作用就是在客户端存储用户访问网站的一些信息。典型的应用有：自动登录。

### （3）缺陷

①Cookie会被附加在每个HTTP请求中，所以无形中增加了流量。

②由于在HTTP请求中的Cookie是明文传递的，所以安全性成问题。（除非用HTTPS）

③Cookie的大小限制在4KB左右。对于复杂的存储需求来说是不够用的。

### （4）常用方法

创建Cookie: `Cookie cookie = new Cookie(name,value)`

向浏览器发送Cookie: `response.addCookie(cookie)`

设置最大时效: `cookie.setMaxAge(秒)`，当设置为0的时候，使用`response.addCookie(cookie)`，表示删除该cookie。

按下图画一个示意图说明一下cookie在浏览器和服务器之间的流转过程



考核点	
学员问题汇总	
作业	
课后 总结分析	

？ ？ ？ ？ ？ ？

章节	第四章 jsp
教学目标 与要求	通过本章的学习，使学生掌握jsp技术，掌握jsp内置对象。
重点 难点	重点：jsp常用标签，jsp内置对象 难点：jsp内置对象

## 【教学进程安排】

- |                |     |
|----------------|-----|
| 1. JSP基本语法     | 2课时 |
| 2.jsp内置对象      | 2课时 |
| 3.jsp与javabean | 2课时 |

## 【课程主要内容】

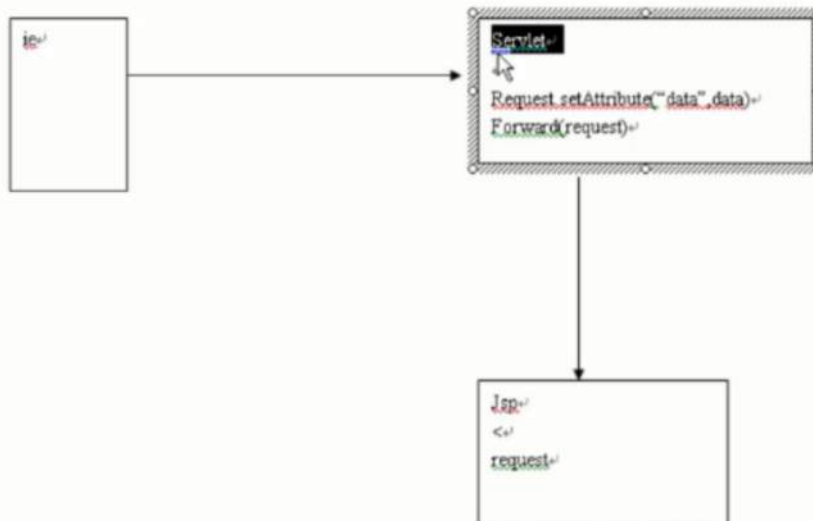
### 10. JSP介绍

通过显示当前时间的jsp例子来讲述jsp被翻译成servlet的原理

jsp最佳实践:

#### Tip: JSP最佳实践

- 不管是JSP还是Servlet，虽然都可以用于开发动态web资源。但由于这2门技术各自的特点，在长期的软件实践中，人们逐渐把servlet作为web应用中的控制器组件来使用，而把JSP技术作为数据显示模板来使用。
- 其原因为，程序的数据通常要美化后再输出：
  - 让jsp既用java代码产生动态数据，又做美化会导致页面难以维护。
  - 让servlet既产生数据，又在里面嵌套html代码美化数据，同样也会导致程序可读性差，难以维护。
  - 因此最好的办法就是根据这两门技术的特点，让它们各自负责各的，servlet只负责响应请求产生数据，并把数据通过转发技术带给jsp，数据的显示jsp来做。



### 11. JSP语法

#### Tip: JSP语法

- JSP模版元素
- JSP表达式
- JSP脚本片断
- JSP注释
- JSP指令
- JSP标签
- JSP内置对象
- 如何查找JSP页面中的错误

考核点	
学员问题汇总	
作业	
课后 总结分析	

？ ？ ？ ？ ？ ？

章节	第五章el&jstl
教学目标 与要求	通过本章学习，使学员掌握了解el与jsp区别，el内置对象，jstl
重点 难点	重点： EL表达式， jstl核心标签库 难点： el内置对象

## 【教学进程安排】

- |           |     |
|-----------|-----|
| 1. el表达式  | 4课时 |
| 2.jstl标签库 | 2课时 |

## 【课程主要内容】

### EL技术:

#### 1.EL 表达式概述:

EL (Express Lanuage) 表达式可以嵌入在jsp页面内部, 减少jsp脚本的编写, EL 出现的目的是要替代jsp页面中脚本的编写。

#### 2.EL从域中取出数据(EL最重要的作用):

jsp脚本: `<%=request.getAttribute(name)%>`

EL表达式替代上面的脚本: `${requestScope.name}`

EL最主要的作用是获得四大域中的数据, 格式`${EL表达式}`

EL获得pageContext域中的值: `${pageScope.key}`;

EL获得request域中的值: `${requestScope.key}`;

EL获得session域中的值: `${sessionScope.key}`;

EL获得application域中的值: `${applicationScope.key}`;

EL从四个域中获得某个值`${key}`;

---同样是依次从pageContext域, request域, session域, application域中 获取属性, 在某个域中获取后将不在向后寻找

### 案例:

- 1) 获得普通字符串
- 2) 获得User对象的值
- 3) 获得List<User>的值

`<%`

`//存储字符串`

`request.setAttribute("company", "baidu");`

`//存储一个对象`

`User user = new User();`

`user.setId(1);`

`user.setName("zhangsan");`

`user.setPassword("123");`

`session.setAttribute("user", user);`

`//存储一个集合`

`List<User> list = new ArrayList<User>();`

`User user1 = new User();`

`user1.setId(2);`

`user1.setName("lisi");`

`user1.setPassword("123");`

考核点	
学员问题汇总	
作业	
课后 总结分析	

？ ？ ？ ？ ？ ？

章节	第六章 监听器与过滤器
教学目标 与要求	通过本章节学习，使学生掌握监听器的使用，及触发监听的情框
重点 难点	重点：监听器与过滤器的使用 难点：监听器与过滤器的使用

## 【教学进程安排】

监听器的生命周期与监听对象      2课时  
过滤器的使用      2课时

## 【课程主要内容】

### 1. 监听器Listener

监听器就是监听某个对象的状态变化的组件。监听器的相关概念事件源：

- 被监听的对象（三个域对象 request, session, servletContext）
- 监听器：监听事件源对象，事件源对象的状态的变化都会触发监听器。
- 注册监听器：将监听器与事件源进行绑定。
- 响应行为：监听器监听到事件源的状态变化时，所涉及的功能代码（程序员编写代码）

按照被监听的对象划分：ServletRequest域；HttpSession域；ServletContext域。按照监听的内容分：监听域对象的创建与销毁的；监听域对象的属性变化的。

	ServletContext域	HttpSession域	ServletRequest域
域对象的创建与销毁	ServletContextListener	HttpSessionListener	ServletRequestListener
域对象内的属性的变化	ServletContextAttributeListener	HttpSessionAttributeListener	ServletRequestAttributeListener

### 三大域对象的创建与销毁的监听器

#### 14.1.1. ServletContextListener

监听ServletContext域的创建与销毁的监听器，Servlet域的生命周期：在服务器启动创建，服务器关闭时销毁；监听器的编写步骤：

- 编写一个监听器类去实现监听器接口
- 覆盖监听器的方法

ServletContextListener监听器的主要作用：

初始化的工作：初始化对象；初始化数据。

例子：MyServletContextListener.java

```
@WebListener()
public class MyServletContextListener implements
ServletContextListener{

    @Override
    //监听context域对象的创建
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("context创建了....");
    }

    //监听context域对象的销毁
    @Override
    public void contextDestroyed(ServletContextEvent sce) {
```

考核点	
学员问题汇总	
作业	
课后 总结分析	

? ? ? ? ? ?

章节	第七章 其它Web开发技术
教学目标 与要求	通过本章节学习，使学生掌握文件上传与下载，数据库连接池，分页查询
重点 难点	重点： 文件上传与下载，数据库连接池，分页查询 难点： 监听器与过滤器的使用

## 【教学进程安排】

监听器的生命周期与监听对象

2课时

文件上传下载

2课时

## 【课程主要内容】

### 14.3. 文件上传

实现 web 开发中的文件上传功能，需完成如下两步操作：

3. 在 jsp 页面中添加上传输入项

4. 在servlet中读取上传文件的数据，并保存到服务器硬盘中

如何在jsp页面中添加上传输入项？

<input type="file">标签用于在 jsp 页面中添加文件上传输入项，设置文件上传输入项时须注意：

必须要设置 input 输入项的 name 属性，否则浏览器将不会发送上传文件的数据。

必须把 form 的 enctype 属性值设为 multipart/form-data 。其实 form 表单在你不写 enctype 属性时，也默认为其添加了 enctype 属性值，默认值是 enctype="application/x-www-form-urlencoded" 设置该值后，浏览器在上传文件时，将把文件数据附带在 http 请求消息体中，并使用 MIME 协议对上传的文件进行描述，以方便接收方对上传数据进行解析和处理。

表单的提交方式必须是post，因为上传文件可能较大。

get：以【明文】方式，通过URL提交数据，数据在URL中可以看到。提交数据最多不超过【2KB】。安全性较低，但效率比post方式高。适合提交数据量不大，且安全要求不高的数据：比如：搜索、查询等功能。

post：将用户提交的信息封装在HTML HEADER内，数据在URL中【不能看到】适合提交数据量大，安全性高的用户信息。如：注册、修改、上传等功能。

区别：

- post隐式提交，get显式提交。
- post安全，get不安全。
- get提交数据的长度有限(255字符之内)，post无限。

示例：

```
<form action="xx.action" method="post" enctype="multipart/form-data">
</form>
```

#### 14.3.1. 第二步

- 如何在 Servlet 中读取文件上传数据，并保存到本地硬盘中？

Request 对象提供了一个 getInputStream 方法，通过这个方法可以读取到客户端提交过来的数据（具体来说是 http 的请求体 entity）。但由于用户可能同时上传多个文件，在 Servlet 端编程直接读取上传数



考核点	
学员问题汇总	
作业	
课后 总结分析	

? ? ? ? ? ?

章节	第8章MVC模式
教学目标 与要求	通过本章学习，使学员掌握同步与异步区别，利用ajax引擎实现异步传输，mvc设计模式。
重点 难点	重点：MVC设计模式， 难点：mvc模式的设计

# 【教学进程安排】

Mvc设计模式

2课时

## 【课程主要内容】

MVC模式:

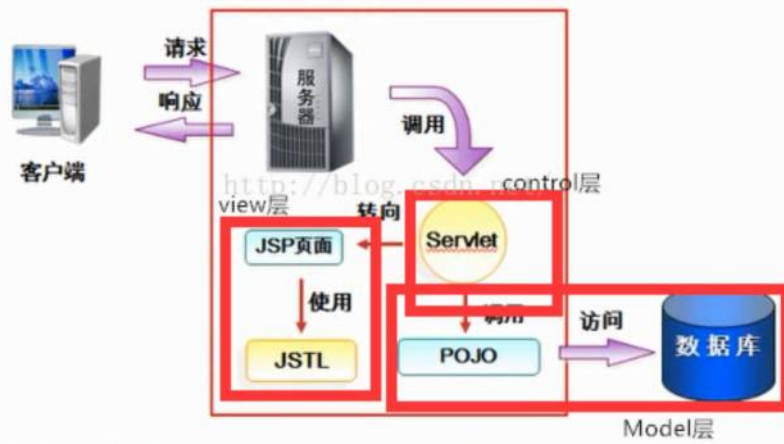
1.MVC: ---- web开发的设计模式

M: Model---模型 javaBean: 封装数据

V: View-----视图 jsp: 单纯进行页面的显示

C: Controller----控制器 Servlet: 获取数据--对数据进行封装--传递数据-- 指派显示的jsp页面

### MVC请求过程



2.javaEE的三层架构:

web层: 与客户端交互

service层: 复杂业务处理

dao层: 与数据库进行交互

开发实践时 三层架构通过包结构体现

三层架构案例:

使用包结构来实现分层:

web层:

ProductListServlet.java代码如下:

```
package com.zl.web;
import java.io.IOException;
import java.sql.SQLException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.zl.domain.Product;
import com.zl.service.ProductService;
```

```
public class ProductListServlet extends HttpServlet {
```

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

```
        //传递请求到service层
```

```
        ProductService service = new ProductService();
```

```
        List<Product> productList = null;
```

```
        try {
```

```
            productList = service.findAllProduct();
```

```
        } catch (SQLException e) {
```

```
            // TODO Auto-generated catch block
```

```
            e.printStackTrace();
```

```
        }
```

考核点	
学员问题汇总	
作业	
课后 总结分析	