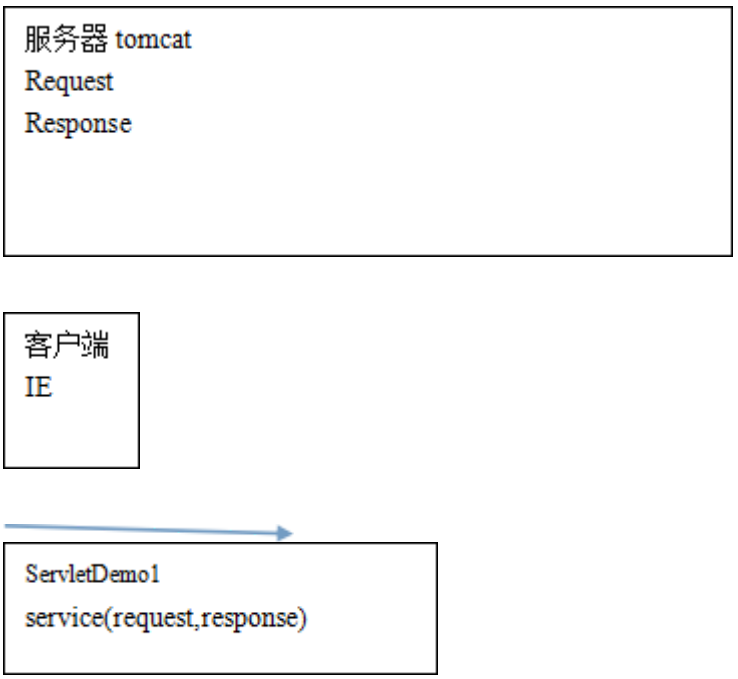
学好的关键：理解HTTP协议



Web服务器收到客户端的http请求，会针对每一次请求，分别创建一个用于代表请求的request对象、和代表响应的response对象。

一、HttpServletResponse

1、响应行　HTTP/1.1　200 OK

1　setStatus(int　sc) 设置响应状态码

2、响应头

1　***** sendRedirect(String　location) 请求重定向

1　setHeader(String　name, String　value) 设置响应头信息

1

```
//告知浏览器使用什么码表
    response.setHeader("content-type", "text/html;charset=UTF-8");

//告知客户端不缓存
```

```
response.setHeader("pragma", "no-cache");
response.setHeader("cache-control", "no-cache");
response.setDateHeader("expires", 0);
```

Referesh刷新

3、响应正文（主体）

l *** getWrite(); 字符输出流

l getOutputStream(); 字节输出流

l setCharacterEncoding(String charset) 告知服务器使用什么编码

l *****setContentType(String type)

二、HttpServletRequest

1、请求行

Get http://localhost:8080/day09/servlet/req1?username=zs http/1.1

getMethod(); 获得请求方式

***getRequestURL();返回客户端发出请求时的完整URL。

***getRequestURI(); 返回请求行中的资源名部分。

*****getContextPath(); 当前应用的虚拟目录 /day09_01_request

getQueryString() ; 返回请求行中的参数部分。

2、请求消息头

* String getHeader(String name) 根据头名称得到头信息值

Enumeration getHeaderNames() 得到所有头信息name

Enumeration getHeaders(String name) 根据头名称得到相同名称头信息值

3、请求正文（重要）

与获取表单数据相关的方法

```
<input type="text" name="username" />
```

*** getParameter(name) 根据表单中name属性的名，获取value属性的值方法

*** getParameterValues（String name）专业为复选框取取提供的方法

getParameterNames() 得到表单提交的所有name的方法

*** getParameterMap 到表单提交的所有值的方法 //做框架用，非常实用

getInputStream 以字节流的方式得到所有表单数据

与操作非表单数据相关的方法(request也是一个域对象)

*** void setAttribute(String name, Object value);

*** Object getAttribute(String name);

Void removeAttribute(String name);

与请求转发相关的方法

//得到请求转发或请求包含的协助对象

RequestDispatcher getRequestDispatcher(String path)

*** forward(ServletRequest request, ServletResponse response) //转发的方法

include(ServletRequest request, ServletResponse response) //请求包含

与请求编码相关的方法：

//解决**post**方式编码

*****request.setCharacterEncoding("UTF-8"); //告诉服务器客户端什么编码,只能处理post请求方式

//解决get方式编码

String name = new String(name.getBytes("iso-8859-1"),"UTF-8");

POST

Servlet:
由于客户端没有告诉服务器，请求正文的编码，
于是服务器默认用ISO-8859-1进行编码
结论：乱码

解决办法：告诉服务器请求正文的数据应该使用的编码是什么？UTF-8
request.setCharacterEncoding("UTF-8");

<%@page pageEncoding="UTF-8"%>
姓名：张三

浏览器当前是什么编码就以什么编码提交

请求正文：
name=%AE%1F%43

Serlvet:
url地址后的参数服务器默认用 ISO-8859-1 进行编码
结论：乱码
解决办法：request.setCharacterEncoding("UTF-8");
结果：无效
String name = reuqest.getParameter("name");
拿到原始的二进制数据，用 UTF-8 进行重新编码
byte b[] = name.getBytes("ISO-8859-1");//1010101
String name = new String(b,"UTF-8");//解决

<%@page pageEncoding="UTF-8"%>
姓名：张三

浏览器当前是什么编码就以什么编码提交

URL？name=%AE%1F%43

GET