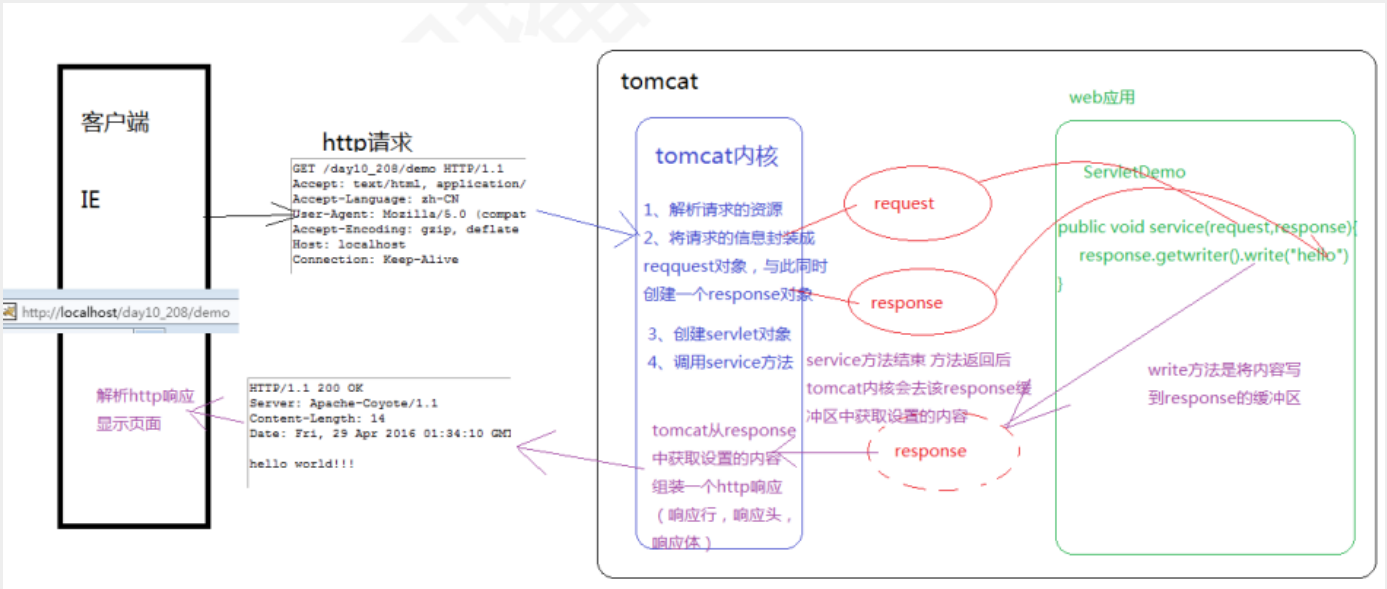
Response响应对象

1 HttpServletResponse概述

在Servlet API中，定义了一个HttpServletResponse接口，它继承自ServletResponse接口，专门用来封装HTTP响应消息。由于HTTP响应消息分为响应行、响应消息头、消息体三部分，因此，在HttpServletResponse接口中定义了向客户端发送响应行、响应消息头、响应消息体的方法，接下来，本节将针对这些方法进行详细的讲解。



2 设置响应行

当Servlet向客户端回送响应消息时，需要在响应消息中设置状态码。为此，在HttpServletResponse接口中，定义了两个发送状态码的方法，具体如下：

1). setStatus (int status)方法

该方法用于设置HTTP响应消息的状态码，并生成响应状态行。由于响应状态行中的状态描述信息直接与状态码相关，而HTTP版本由服务器确定，因此，只要setStatus(int status)方法设置了状态码，即可实现状态行的发送。需要注意的是，正常情况下，Web服务器会默认产生一个状态码为200的状态行。

2). sendError(int sc)方法

该方法用于发送表示错误信息的状态码，例如，404状态码表示找不到客户端请求的资源。在response对象中，提供了两个重载的sendError(int sc)方法，具体如下:

```
        public void sendError(int code) throws Java.io.IOException
        public void sendError(int code, string message) throws Java.io.IOExce
ption
```

在上面重载的两个方法中，第一个方法只是发送错误信息的状态码，而第二个方法除了发送状态码外，还可以增加一条用于提示说明的文本信息，该文本信息将出现在发送给客户端的正文内容中。

3 设置响应头

1). void addHeader(String name,String value)与void setHeader(String name,String value)

前者对响应头的键值对来说可以多对多，后者会覆盖，只能一对一。

2). void addIntHeader(String name,String value)与void setIntHeader(String name,String value)

和上面的两个方法类似，但是此二法专门用于设置包含整数值的响应头。

3). void setContentLength(int len)

该方法专门用于设置响应消息的实体内容大小，单位是字节。对于HTTP协议来说，这个方法就是设置Content-Length响应头字段的值。

其余方法暂时不提

4 重定向

```
9  public class Servlet1 extends HttpServlet {
10
11     protected void doGet(HttpServletRequest request, HttpServletResponse response)
12             throws ServletException, IOException {
13
14
15         //没有响应 告知客户端去重定向到servlet2
16         //1、设置状态码302
17         //response.setStatus(302);
18         //2、设置响应头Location
19         //response.setHeader("Location", "/WEB14/servlet2");
20
21         //封装成一个重定向的方法sendRedirect(url)
22         response.sendRedirect("/WEB14/servlet2");
23     }
24
```

重新定向为：

```
11     protected void doGet(HttpServletRequest request, HttpServletResponse response)
12             throws ServletException, IOException {
13
14         response.getWriter().write("Hello HttpServlet2");
15     }
16
```

此时，我们访问servlet1将会重新定向到servlet2，并在浏览器上打印Hello HttpServlet2。

5 设置响应体

5.1 修改response编码格式以解决中文乱码问题

void setContentType(String type)

该方法用于设置Servlet输出内容的MIME类型，对于HTTP协议来说，就是设置Content-Type响应头字段的值。例如，如果发送给客户端的内容是jpeg格式的图像数据，就需要将响应头字段的类型设置为"image/jpeg"。需要注意的是，如果响应的内容为文本，setContentType()方法还可以设置字符编码，如：text/html;charset=UTF-8。

```
//设置response查询的码表
    response.setCharacterEncoding("UTF-8");
//通过一个头 Content-Type 告知客户端使用何种码表
    response.setHeader("Content-Type", "text/html;charset=UTF-8");
```

上述代码等于response.setContentType("text/html;charset=UTF-8"),由于response默认缓冲区的默认编码格式是iso8859-1，该格式不能显示中文，所以我们可以通过以上代码改变response编码的缓冲区。
如果此时还是不能显示中文，说明此时浏览器默认编码格式与response缓冲区编码格式不一致，可以设置浏览器的编码格式一致即可。

其实response.setContentType("text/html;charset=UTF-8")设置了浏览器与response缓冲区编码格式都为UTF-8。

5.2 响应体设置文本

由于在HTTP响应消息中，大量的数据都是通过响应消息体传递的，因此，ServletResponse遵循以IO流传递大量数据的设计理念。如果需要发送文本，则需要用PrintWriter对象

```
//同时设置浏览器与response对象的编码格式为UTF-8
response.setContentType("text/html;charset=UTF-8");

PrintWriter writer = response.getWriter();
//writer.write("hello response!!!");
writer.write("你好");
```

5.3 响应体设置字节

当然是用字节流对象咯。用response.getOutputStream()获得字节流对象

```
//使用response获得字节输出流
ServletOutputStream out = response.getOutputStream();

//获得服务器上的图片
//用上下文对象获得路径
String realPath = this.getServletContext().getRealPath("a.jpg");
InputStream in = new FileInputStream(realPath);

int len = 0;
byte[] buffer = new byte[1024];
while((len=in.read(buffer))>0){
    out.write(buffer, 0, len);
}

in.close();
out.close();
```