# 重定向

## 在一个web应用中如何完成资源的跳转？

在一个web应用中可以通过两种方式完成资源的跳转：

1. **转发**
2. **重定向**

## 转发和重定向的区别

## 代码上的区别

**转发的代码**： ·request.getRequestDispatcher(“/dept/list”).forward(request, response);· （获取转发器调用forward函数）

转发的时候是一次请求，不管转发了多少次，都是一次请求。这是因为调用forward函数的时候，会将当前的request和response转递给下一个Servlet。

**重定向代码**： ·response.sendRedirect(req.getContextPath() + “/servlet/b”);·

重定向式由response对象执行的，实际上是response将程序指定的地址响应给浏览器，然后让浏览器自动向（新的地址）发送了请求。

所以重定向会导致地址栏的内容发生改变。

注意：重定向的请求路径是要添加根目录/项目名的。

## 形式上的区别

使用转发的话，在地址栏上发送的请求是/servlet/a，最终（转发以后）地址栏的地址还是/servlet/a。--》**转发是一次请求**。

使用重定向，在地址栏上发送的请求和最终显示的不是同一个地址 --》**重定向不只有一次请求**。

## 本质区别

转发是由web服务器实现的，a资源到b资源的跳转动作，是服务器内部完成的；而重定向是浏览器完成的，具体跳转到哪个资源，是浏览器说了算。

# 如何选择转发和重定向

什么时候使用转发或者重定向？

如果在上一个Servlet当中的request域中绑定了数据，希望在下一个Servlet中获取这些数据，就使用转发机制。

其余情况全部重定向。

不论是转发还是重定向，跳转的资源只要是服务器内的合法资源就可以。

转发会存在浏览器的刷新问题。

因为转发是一次请求，所以如果一直刷新浏览器，就会一直重复转发，从而导致多次执行后端Servlet程序。

因此建议（如果可以的话）使用重定向取代大部分的转发的情况。