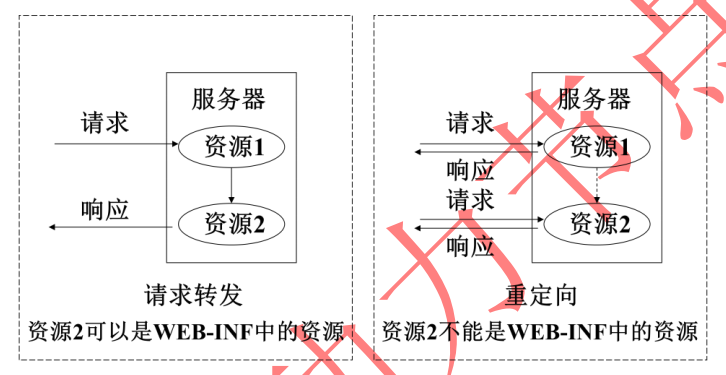
当处理器对请求处理完毕后，向其它资源进行跳转时，有两种跳转方式：***请求转发与重定向***。而根据所要跳转的资源类型，又可分为两类：***跳转到页面与跳转到其它处理器。***

注意，对于请求转发的页面，可以是WEB-INF中页面；而重定向的页面，是不能为WEB-INF中页的。因为***重定向相当于用户再次发出一次请求***，而用户是不能直接访问WEB-INF中资源的。



**服务器内跳转、服务器外跳转**

1. **返回ModelAndView时的请求转发**
2. 请求转发到页面
3. 请求转发到其他Controller
4. **返回ModelAndView时的重定向**

返回ModelAndView时的重定向，***需在setViewName()指定的视图前添加redirect:***，且此时的视图不会再与视图解析器中的前辍与后辍进行拼接。即必须写出相对于项目根的路径。故此时的视图解析器不再需要前辍与后辍。

重定向的目标资源中，将无法访问用户提交请求request中的数据。

1. 重定向到页面

在重定向时，请求参数是无法通过request的属性向下一级资源中传递的。但可以通过以下方式将请求参数向下传递。

1. 通过ModelAndView的Model携带参数

当ModelAndView中的Model存入数据后，视图解析器 InternalResourceViewResolver会将map中的key与value，以请求参数的形式 放到请求的URL后。

不过需要注意以下几点：

* 由于视图解析器会将Map的value放入到URL后作为请求参数传递出去，所以无论什么类型的value，均会变为String。故此，放入到Model中的value，只能是基本数据类型与String，不能是自定义类型的对象数据。
* 重定向的面页中是无法从request中读取数据的。但由于map中的key与value，以请求参数的形式放到了请求的URL后，所以，页面可以通过EL表达式中的请求参数param读取。
* 重定向的页面不能是/WEB-INF中的页面。

1. 使用HttpSession携带参数（自己练习）
2. 重定向到Controller
3. **返回String时的请求转发**
4. **返回String时的重定向**
5. 重定向到页面
6. 重定向到Controller
7. **返回void时的请求转发**
8. **返回void时的重定向**