

# 返回值类型

spring mvc 支持如下的返回方式: ModelAndView, Model, ModelMap, Map, View, String, void。

## 一、String

### 1) : 字符串代表逻辑视图名 (能使用视图解析器)

真实的访问路径= “前缀” + 逻辑视图名 + “后缀”

```
@RequestMapping("/testString")
public String testString(Model model){
    System.out.println("testString方法执行了...");
    // 模拟从数据库中查询出User对象
    User user = new User();
    user.setUsername("美美");
    user.setPassword("123");
    user.setAge(30);
    // model对象
    model.addAttribute("user",user);
    return "success";
}
```

### 2) : 代表redirect重定向

redirect的特点和servlet一样, 使用redirect进行重定向那么地址栏中的URL会发生变化, 同时不会携带上一次的request

```
@RequestMapping("/testForwardOrRedirect")
public String testForwardOrRedirect(){
    System.out.println("testForwardOrRedirect方法执行了...");
    // 重定向
    return "redirect:/index.jsp";
}
```

### 3) : 代表forward转发

通过forward进行转发, 地址栏中的URL不会发生改变, 同时会将上一次的request携带到写一次请求中去

```
@RequestMapping("/testForwardOrRedirect")
public String testForwardOrRedirect(){
    // 请求的转发
    return "forward:/WEB-INF/pages/success.jsp";
}
```

## 二、void

**注意：**返回这种结果的时候可以在Controller方法的形参中定义HttpServletRequest和HttpServletResponse对象进行请求的接收和响应

```
@RequestMapping("/testVoid")
public void testVoid(HttpServletRequest request, HttpServletResponse response)
throws Exception {
    System.out.println("testVoid方法执行了...");
    // 1、编写请求转发的程序
    // request.getRequestDispatcher("/WEB-INF/pages/success.jsp").forward(request,response);

    // 2、重定向
    // response.sendRedirect(request.getContextPath()+"/index.jsp");

    // 3、设置中文乱码 直接会进行响应
    response.setCharacterEncoding("UTF-8");
    response.setContentType("text/html;charset=UTF-8");
    response.getWriter().print("你好");

    return;
}
```

## 三、ModelAndView（最常见）

**1、通过ModelAndView构造方法可以指定返回的页面名称，也可以通过setViewName()方法跳转到指定的页面**

### 2、能使用视图解析器

```
@RequestMapping("/testModelAndView")
public ModelAndView testModelAndView(){
    // 创建ModelAndView对象
    ModelAndView mv = new ModelAndView();
    System.out.println("testModelAndView方法执行了...");
    // 模拟从数据库中查询出User对象
    User user = new User();
    user.setUsername("小凤");
    user.setPassword("456");
    user.setAge(30);

    // 把user对象存储到mv对象中，也会把user对象存入到request对象
    mv.addObject("user",user);

    // 跳转到哪个页面
    mv.setViewName("success");
}
```

```
    return mv;  
}
```