

4.8 第69页

第69页最后一行，原文是：

在浏览器中提供/hello2和/hello接口分别进行访问，当访问/hello接口时...

修改后是：

在浏览器中提供/hello2和/hello接口分别进行访问，当访问/hello2接口时...

5.4.3 第103页

第103页第一行，原文是：

DataSourceConfig也和5.4.1小节一致，这里不赘述。

修改为：

DataSourceConfig也和5.4.1小节基本一致，不同的是，这里多了一个@Primary注解，代码如下：

```
@Configuration
public class DataSourceConfig {
    @Bean
    @ConfigurationProperties("spring.datasource.one")
    @Primary
    DataSource dsOne() {
        return DruidDataSourceBuilder.create().build();
    }
    @Bean
    @ConfigurationProperties("spring.datasource.two")
    DataSource dsTwo() {
        return DruidDataSourceBuilder.create().build();
    }
}
```

6.1.4 第120页

修改代码片段，原文是：

```
@Bean
StringRedisTemplate stringRedisTemplate() {
    StringRedisTemplate stringRedisTemplate = new
StringRedisTemplate(jedisConnectionFactory());
    stringRedisTemplate.setKeySerializer(new StringRedisSerializer());
    stringRedisTemplate.setKeySerializer(new StringRedisSerializer());
}
```

```
        return stringRedisTemplate;
    }
```

修改后是：

```
@Bean
StringRedisTemplate stringRedisTemplate() {
    StringRedisTemplate stringRedisTemplate = new
StringRedisTemplate(jedisConnectionFactory());
    stringRedisTemplate.setKeySerializer(new StringRedisSerializer());
    stringRedisTemplate.setValueSerializer(new StringRedisSerializer());
    return stringRedisTemplate;
}
```

13.3.2 第247页

第247页中间的代码片段，原文是：

```
@RestController
public class HelloController {
    @Autowired
    JobLauncher jobLauncher;
    @Autowired
    Job job;
    @GetMapping("/hello")
    public void hello() {
        try {
            jobLauncher.run(job, null);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

修改后是（就是把原文的null换成黄色背景的那一段，修改时不需要背景，这里我只是标记下方便查看）：

```
@RestController
public class HelloController {
    @Autowired
    JobLauncher jobLauncher;
    @Autowired
    Job job;
    @GetMapping("/hello")
    public void hello() {
        try {
            jobLauncher.run(job, new JobParametersBuilder().toJobParameters());
        }
    }
}
```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

16.10 第331页

原文是：

然后创建消息转发Controller，如下：

原文的代码片段

修改后是（这里就是修改原文的代码片段）：

然后创建消息转发Controller，如下：

```
@Controller  
public class WsController {  
    @Autowired  
    SimpMessagingTemplate messagingTemplate;  
  
    @MessageMapping("/ws/chat")  
    public void handleChat(Principal principal, String msg) {  
        String destUser = msg.substring(msg.lastIndexOf(";") + 1, msg.length());  
        String message = msg.substring(0, msg.lastIndexOf(";"));  
        messagingTemplate.convertAndSendToUser(destUser, "/queue/chat", new  
ChatResp(message, principal.getName()));  
    }  
    @MessageMapping("/ws/nf")  
    @SendTo("/topic/nf")  
    public String handleNF() {  
        return "系统消息";  
    }  
}
```

另外，读者可以关注我的公众号，书中的问题可以和我做进一步讨论。

