

论前后端分离

在 Web 工程课程项目的开发中，随着对项目的不断分析和开发的持续，对于 Web 项目前后端的耦合与分离产生了一些思考，就此写下了这些想法。

在项目中需要根据用户之前的选择操作动态地生成前端界面，而前端所要生成的界面又需要从数据库中向其中填充数据。对于这个问题，经过考虑，可以使用两种方式进行解决。

一、使用后端技术 `jsp+el+jstl` 根据从数据库中获取的数据生成前端的动态代码；

二、使用前端技术 `js+Ajax` 在页面加载时，通过服务器接口获取数据生成网页；

基于这两方面考虑，最终团队决定对两种方式都做实现（达到学习的目的），但这个与前后端分离的话题值得深入探讨。

1 新旧前后端方式

前面关于项目中的问题，实际牵扯到了新旧前后端耦合与分离的两种实现方式，下面简要介绍两种实现形式，并进行优缺点分析。

1.1 旧方式

此旧方式是以 Java Web 开发使用 `jsp` 技术为例，以下是客户端一次请求和得到响应的过程：

1.1.1 客户端发起一个请求；

1.1.2 服务器端的 `controller`（`servlet`）接受到请求；

1.1.3 `Controller` 调用 `service`、`dao` 层代码完成业务逻辑；

1.1.4 数据经过处理返回到 `jsp`；

1.1.5 `Jsp` 展现一些动态代码到客户端；

1.2 新方式

新方式放弃使用 `jsp` 技术，使用 `js+Ajax` 技术实现，基本流程如下：

1.2.1 客户端发起一个请求；

1.2.2 客户端直接得到 `html` 页面；

1.2.3 `Html` 页面全负责调用数据库接口产生数据（`Ajax`）；

1.2.4 数据填充 `html`，前端展现动态效果；

1.3 优缺点

对于旧方式，使用 `jsp` 技术存在开发维护和服务器压力等方面的痛点。

痛点如下：

1.3.1 无法做到真正的动静分离，因为动态资源和静态资源全部耦合在一起。如果因为用户

量等等原因导致服务器的压力过大，服务器收到的各种请求都无法及时反馈。当服务器出现问题时，将导致用户体验较差；

1.3.2 对于前端工程师使用 html、css、js 方式完成的代码，交到后端工程师的手中，使用 jsp 改写之后使页面错综复杂，当前端有 bug 等需要修改时，降低了开发效率；

1.3.3 Jsp 代码必须在支持 Java 的 Web 服务器容器中运行（如 Tomcat），无法使用 nginx 等，对于大型网站的性能是巨大的挑战；

1.3.4 同时 jsp 也有响应速度上的缺点，如第一次请求速度较慢，当 jsp 页面过大时，动态数据加载与 html+Ajax 方式有较大性能差距；

对于使用 jsp 技术的各种痛点，使用 html+Ajax 技术有较好的改善，新方式优点如下：

1.3.5 可以实现真正的前后端解耦，可以在前端使用 nginx 服务器，在前端服务器放一些 css、js 等一些静态资源，同时前端服务器也负责页面的跳转等，调用后端接口生成动态页面，后端使用 Tomcat 等服务器；

1.3.6 当系统出现了 bug 时，可以精准地定位到问题发生的原因及位置，及时解决问题；

1.3.7 减少后端服务器的压力，让后端服务器专注与数据和业务逻辑的处理，提高服务器的负载能力和并发能力；

1.3.8 Ajax 使用异步形式进行动态数据的加载，即使数据量特别大也不会过多的影响性能，在后端服务器出现问题时，客户端界面仍然可以获取到部分静态数据；

2 标准与发展

2.1 业界标准

前后端分离现今已成为各个互联网项目开发的业界标准了，通过 nginx+Tomcat 的方式进行前后端真正的有效的解耦，这种形式有利于企业设计大型的网站，有更大的负载能力和并发能力，同时会对以后的分布式架构、微服务架构等多种形式打下坚实的基础。

2.2 术业有专攻

在不久之前，大多说 Java Web 项目都是 Java 程序负责各个模块的开发，既负责后端有要负责前端，设计 Java、Mysql、jsp、html、js、css 等多方面的知识，但开发出的项目大多都难以维护，并且对于服务器的压力等方面的考量也不足。

但随着技术的发展，前后端渐渐的界限明确，后端只负责后端，前端只负责前端，正所谓术业有专攻，对于一方面的专注，一定会带来更好的发展，这是对公司和个人都有利的形式。

2.3 对技术发展的意义

当前端与后端真正的实现解耦之后，前后端实现了百花齐放，不断有各种方便开发与提高架构性能的开源项目出现。加快了前后端技术的进步。

对于后端，从 struts2+spring+hibernate 到 spring+springMVC+Mybatis 再到 springBoot+Mybatis 的后端框架发展，越来越便于开发者的开发，集便利的开发与提供给开发者更多的灵活性的形式。

对于前端，从基础的 `html+css+js` 到简单前端框架 `jQuery`、`Bootstrap` 等的使用，再到 `React.js`、`Vue.js`、`Zepto.js` 等的出现，再到 `Node.js` 的全栈开发。发展迅速，也形成了与后端开发对应的整体规模。

3 总结

对于前后端分离解耦的思考并没有至此结束，本次的 **Web** 工程项目是对此问题的小缩影。通过小的项目厘清其中的细微差距，是对于未来大型企业项目的小铺垫，追求新的技术是发展的动力，分离或许能够带来百家争鸣的盛大景象。