前后端分离开发模式设计方案

马少博 2018年8月17日

##### 前端技术选型

1. 对于需支持IE9以下浏览器的项目,使用jQuery的AJAX方法获取数据并填充页面。前端页面的 跳转利用ngnix或其他服务器进行跳转,页面跳转根据服务器相对路径进行跳转(例: login.html **=>** ./../index/index.html?id=1)。
2. 对于高级浏览器,推荐尝试Vue框架进行项目开发。在之前的服务支撑平台项目开发中,部分页面已尝试使用Vue进行开发,从实际开发效果看,Vue在组件复用,开发效率,后期维护方面有很大的优势。

Vue的组件封装,不仅在当前项目中方便开发,基础组件和公司某些特定业务组件也可以在其他项目中直接复用,长期来看对公司的技术积累,项目开发也有很大的好处。

Vue框架如果可以启用Vue-cli进行开发,则项目可以创建为SPA模式,前端的路由管理可以由前端自己处理,不再依赖后端构建controller。现有的前端路由技术可以实现页面跳转,权限控制,异步加载等功能,对于无特殊浏览器要求的项目,可以采用此种方法。

##### 前后端数据交互

1. 前后端在项目初期约定各个接口的地址及相应数据格式,项目一律采用JSON格式进行数据传输。
2. API出参约定

约定后端返回所有接口为固定格式,和之前服务支撑平台的接口格式类似,返回的查询结果统一放置在data.bean中

1. API入参约定

约定request header中携带用户身份token,用于校验用户身份及登录状态。参数一律为JSON格式。

1. 身份权限校验,目前已知解决方案有两种
2. 根据前端token,后端采用切面编程方式进行身份权限验证(之前做过的前后端分离项目采用了此种方式)。
3. 同域/跨域 均可设置携带cookie请求,后台根据cookie确定用户身份及登录状态等内容。
4. 前后端约定HTTP请求方法以进行不同类型的操作,这样根据请求的方法就可以确定接口的作用类型,便于接口的管理及维护。常用请求方法如下:

**GET**:用于查询请求数据对象或列表

**POST**:用于新增数据类型请求

**PATCH**:用于更新数据类型请求

**DELETE**:用于删除数据类型请求

##### 后台人员技术支持

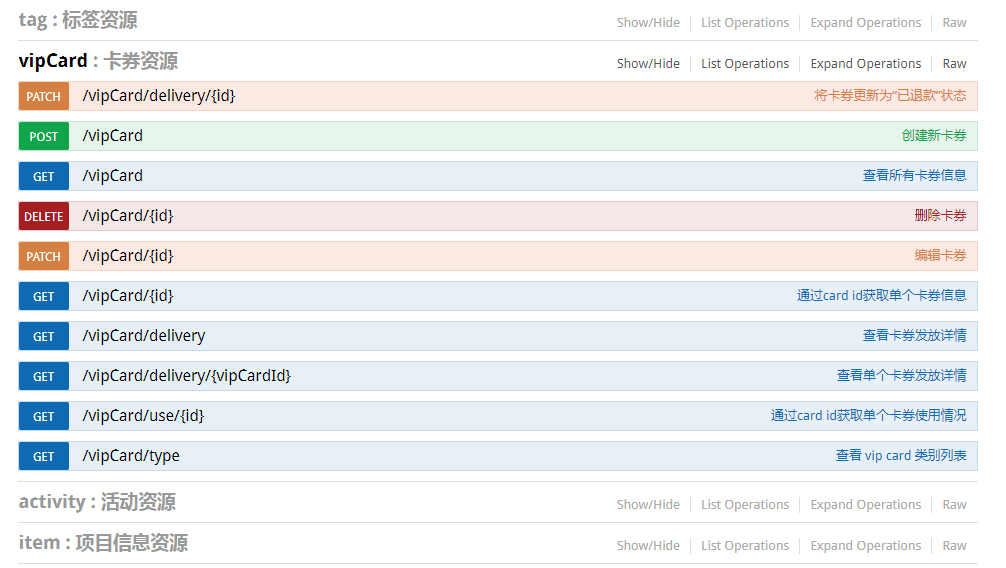
1. 项目初期约定接口名称及参数、响应格式。将每个接口的接口说明及参数说明,存放至API管理工具上。
2. 如果启用API管理工具,则在方法注释及参数说明上需详细标注。某些API工具需要特定的格式标明注释,需要后台人员完成。
3. 在前端发来数据请求时,获取接口中的token信息,需要权限的接口先验证token的可用性及时效性,然后再进行对应的操作逻辑。
4. 后端以开发接口为主，不再参与页面开发，或者混合式jsp页面开发，统一以接口形式返回，前端通过js渲染数据以及处理逻辑。
5. 接口开发尽量考虑到平台复用,对于相同的业务逻辑, PC、移动、小程序共用一套接口, 不在因为平台的变化而去新开发后台项目。

##### Node.js作用

1. 在前端开发时,可以利用Nodejs很方便的启动本地服务器,用于前端开发调试。后期前端项目部署时可以直接用其他服务器代替。
2. 对于某些有SEO需求的项目,可以使用Nodejs服务端渲染方式,由于语言是前端熟悉的javacript,所以可以很快上手应用。
3. 对于部分旧的项目接口,可能会存在多种接口数据格式,如XML,TXT等类型。可以使用Node后台服务统一转化为JSON接口,并将接口名称重新规范,这样既不影响原有后台接口项目,而前端项目中使用规范的接口开发也可以提升效率。
4. 现在流行的前端打包,测试等框架大多基于Nodejs服务生成,前端人员要跟上技术脚步,提升工作效率,Node是不可或缺的一部分。

##### 其他

1. SPA应用模式需要高级浏览器支持,所以对某些特定业务有限制。
2. 前后端分离对前台开发人员的技术素质要求更高,对于一些新的技术选型,要做到稳重选择,团队要及时进行技术交流与培训。
3. 推荐使用API管理工具,前后端分离开发,一个好的API管理工具是必不可少的,这样不仅可以提升前后端接口稳定及沟通效率,并且在后期项目有变动时接口可以及时更新便于维护。如果有新人加入时可以也更快的了解接口的作用和相关业务。目前口碑较好的有swagger(之前项目开发用过),rap等,可根据团队需要进行选择。
4. API管理工具一般要求接口,参数注释比较详细,出参入参有严格的要求,这也就要求后台人员要及时维护并详细编写接口注释。

附swagger工具截图: