大厂面试课之Go语言

第19课：前后端分离

同学们好！世界上一切事物的发展规律总是由简单到复杂，再由复杂到更高层次的简单，往复迭代，以否定之否定的形式，螺旋式的上升，正所谓“反者，道之动也”。而推动软件技术持续发展的源动力，就是新业务的不断涌现。新业务为了满足新需求，对技术提出越来越高的要求。原有的技术无法满足新业务的需求，自然会催生新技术的革命。新技术在满足新需求的同时，又使得更新的业务成为可能，倒逼相关行业的发展。业务与技术彼此交感，相互促进，才有了我们今天看到的互联网。这节课我们要为大家介绍的前后端分离，正是这方面的典型范例之一。业务推动技术，技术催生业务。

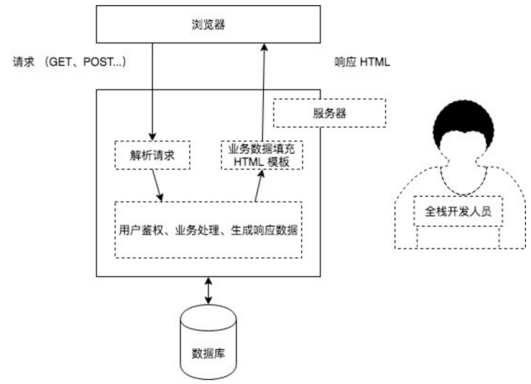
首先我们先来回顾一下传统的，前后端不分离的开发模式，它有什么弊端？为了解决这些弊端，现代的，前后端分离的开发模式，又是怎样的？最后回到《我家租房网》项目，看看在前后端分离方面，我们是怎么做的？

传统（前后端不分离）开发模式

曾几何时，前后端不分离的开发模式，可以说是一切B/S架构应用系统的标准模式。

浏览器——应用服务器——数据库

知识讲解



- 浏览器只负责页面渲染，不负责加载数据
- 应用服务器负责执行业务逻辑，访问数据库，并将数据加载到页面中

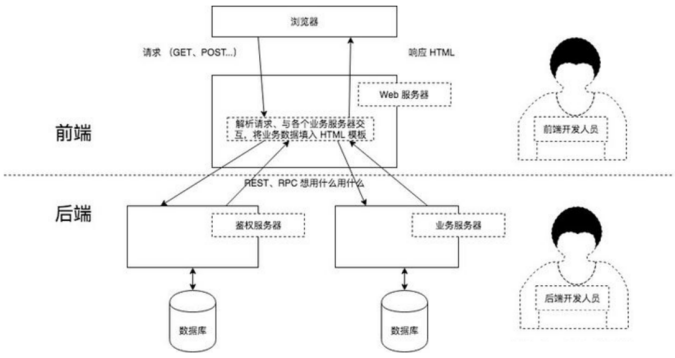
典型的前后端不分离的开发模式，也被称为浏览器——应用服务器——数据库模式。在这样的应用系统中，浏览器向应用服务器发起HTTP请求，应用服务器解析这些请求，提取用户的输入数据，执行业务处理，其间可能还需要操作数据库，最后将处理结果填入类似JSP等形式的HTML模板，生成响应页面并返回给浏览器。在这样的系统中，浏览器只负责页面的渲染，不负责加载数据，而所有的业务逻辑、数据库访问和将数据加载到页面中的工作，全部由应用服务器承担。这种开发模式最大的弊端就是页面和业务逻辑的过度耦合，修改其中的任何一个，都不可避免地会影响另一个。与此同时，这种开发模式对服务器开发人员的要求非常高，他们既要关注业务逻辑，还要做好用户界面，既要精通Java、Go、SQL等后端技术，也要熟悉H5、JS等前端技术，堪称真正意义上的全栈工程师。

现代（前后端分离）开发模式

早期的前端技术还不象今天这般完善且成体系，也没有那么多专业化的编程框架。网页设计甚至不被认为是一件程序员的份内工作。但现在的情况变了，前端页面制作已然发展成为一门自成体系且高度专业化的技艺。后端程序员兼职前端开发的传统模式，已经越来越无法满足新时代的要求。前后端分离的开发模式，应运而生。

前端——后端——数据库

知识讲解



- 前端负责将从后端得到的数据加载到页面中，并渲染之
- 后端负责执行业务逻辑，访问数据库，并将数据返回给前端

前后端分离的开发模式，其实就是将原来的应用服务器一分为二，变成前端服务器和后端服务器，两个物理独立的服务器软件。前端服务器负责解析浏览器的Web请求，调用后端服务器的业务方法，并将处理结果加载到页面中，回传给浏览器。而后端服务器则只专注于业务处理和数据库访问。前后端之间的数据交换其实有很多选择，可以是REST，也可以是Feign或者Dubbo，甚至可以用RPC。这种开发模式从根本上降低了页面和业务逻辑的耦合度，修改其中的任何一个，都不会影响另一个，而且从职责上将前端开发工程师和后端开发工程师做了明确的分工，他们各自只关注各自的领域，在不同的技术栈上愉快地工作。

前后端分离并非浏览器和服务器的分离

知识讲解

- 前后端分离中的前端通常包括浏览器和Web服务器
- 浏览器下载并渲染取自Web服务器的静态页面，同时执行其中包含的JS代码
- JS代码继续向Web服务器发起HTTP请求，并携带必要的参数
- Web服务器将来自浏览器的请求路由到对后端接口的调用
- Web服务器将后端的处理结果返回给浏览器中的JS代码
- 浏览器将处理结果中包含的数据加载到页面元素并显示出来

需要强调的是，所谓前后端分离并非浏览器和服务器的分离。前后端分离中的前端通常包括浏览器和前端服务器。浏览器下载并渲染源自前端服务器的静态页面，同时执行嵌入其中的JavaScript代码。这些JavaScript代码继续向前端服务器发起Web请求，并携带必要的参数。前端服务器将来自浏览器的请求路由到对后端服务器远程方法的调用，并将处理结果返回给浏览器中的JavaScript代码。JavaScript代码再将这些数据加载到页面元素中，并通过浏览器渲染呈现在用户面前。

前后端分离是一种模式而非技术

- 前后端分离只是一种构建互联网应用的架构模式
- 所谓模式就是一种惯用法，大家都觉得这么干不错，你也就这么干了

知识讲解

前后端分离是一种基于解耦理念的最佳实践

- 在接口不变的前提下，后端的任何改变都不会影响到前端，不会影响到用户使用
- 在接口不变的前提下，前端的任何改变也不会影响到后端，不会影响到业务实现

前后端分离让专业的人做专业的事

- 前端开发人员无需关心业务逻辑的实现细节，只需遵照既定的接口做好用户界面
- 后端开发人员无需关心用户界面的显示效果，只需专注于业务逻辑和数据库访问

从本质上看，与其说前后端分离是一项技术，倒不如说它是一种模式，只是为了实现这种模式，确实用到了很多技术。所谓模式其实就是一种惯用法，你用我用大家用，都觉得用着不错。前后端分离的核心理念其实还是解耦。只要接口不变，改了后端不影响前端，用户该怎么用还怎么用。同样，改了前端也不影响后端，业务逻辑该怎么实现还怎么实现，只是看上去跟以前不同了。最重要的是，前后端分离让专业的人专心做专业的事。前端开发人员无需关心业务逻辑的实现细节，只需遵照既定的接口做好用户界面即可。后端开发人员无需关心用户界面的显示效果，只需专注于业务逻辑和数据库访问即可。

前端视图代码结构

9

我们前面花了这么大篇幅从RPC讲起，包括ProtoBuf数据及服务描述语言、gRPC框架、Consul服务注册与发现、go-micro框架、Gin框架、RESTful接口设计风格，一直到MVC代码组织架构，所有这一切，其实都是在为前后端分离的开发模式提供技术支撑。从浏览器到前端服务器，再从前端服务器到微服务形式的后端服务器，这条数据通道已被打通，前后端分离水到渠成。前面我们讲了，在前后端分离的开发模式中，浏览器需要从前端服务器下载并渲染包含JavaScript代码的静态页面。我们的《我家租房网》项目中的静态页面和JavaScript代码又在哪里呢?

项目实战

```
GOPATH\src\iHome\front-end\view
|
|_css/           // 样式表，字体、颜色、阴影等
|_images/        // 图片
|_js/            // 向后端发送请求并接收响应
|_plugins/       // 插件
|
|_favicon.ico    // 首页图标
|_auth.html      // 实名认证页面
|_booking.html   // 预定页面
|_detail.html    // 房屋详情页面
|_index.html     // 默认首页
|_login.html     // 登录页面
|_lorders.html   // 房东订单页面
|_my.html        // 个人信息页面
|_myhouse.html   // 个人房屋列表页面
|_newhouse.html  // 发布新房源页面
|_orders.html    // 租户订单页面
|_profile.html   // 预加载页面
|_register.html  // 注册页面
|_search.html    // 搜索页面
```

10

静态页面、JavaScript代码、层叠式样式单、图标图片等等，共同构成《我家租房网》项目的用户界面，其很重要的一部分的功能是向用户展示数据。这在基于MVC的代码组织架构中显然属于V，即视图层的部分。因此我们将这些内容放置在front-end工程目录的view子目录下。其中包括首页图标和13个静态页面，具体为实名认证页面、预定页面、房屋详情页面、默认首页、登录页面、房东订单页面、个人信息页面、个人房屋列表页面、发布新房源页面、租户订单页面、预加载页面、注册页面和搜索页面。在这些页面中会引用层叠式样式单对字体、颜色、阴影等样式的定义，也会嵌入图片和对JavaScript代码的调用，这些被调用的JavaScript代码又会依赖某些插件。而所有这些层叠式样式单、图片、JavaScript代码和插件，均被分别放置在css、images、js和plugins四个独立的子目录中。

更多精彩，敬请期待

谢谢大家，我们下节课再见！