今天我们简单认识了vue

Vue (读音 /vjuː/，类似于 **view**) 是一套用于构建用户界面的**渐进式框架**。

然后我们学习了

1. 搭建后台管理前端

1.1导入已有资源，

解压放到工作目录中，然后在idea中导入新的工程。

1.2安装依赖

package.json中依然定义了我们所需的一切依赖。

打开终端，进入项目目录，输入npm install命令，即可安装这些依赖。

可以输入命令：npm run dev（重新运行）或npm start（开始运行）

发现默认的端口是9001。访问：<http://localhost:9001>。

1.3域名解析

本地域名解析：

Windows下的hosts文件地址：C:/Windows/System32/drivers/etc/hosts

1.4解决域名解析问题

修改本地的host为：  
127.0.0.1 api.leyou.com：我们的网关Zuul  
127.0.0.1 manage.leyou.com：我们的后台系统地址

然后，Ping一下域名试试是否通畅。

1.5nginx解决端口问题

域名问题解决了，但是现在要访问后台页面，还得自己加上端口：<http://manage.taotao.com:9001>。

我们希望直接域名访问没有端口号显示，要用到反代理工具：Nginx。

1.6安装和使用

资料里的压缩包直接解压拿出来用。

目录结构：

1. conf：配置目录
2. contrib：第三方依赖
3. html：默认的静态资源目录，类似于tomcat的webapps
4. logs：日志目录
5. nginx.exe：启动程序。可双击运行，但不建议这么做。

Nginx中每个server就是一个反向代理配置（可以有多个server）

使用：

Neginx可以通过命令启动：

启动：start nginx.exe

停止：nginx.exe –s stop

重新加载：nginx.exe –s reload

启动过程会闪烁，成功后，任务管理器会有俩个nginx进程。

最后用域名访问后台管理系统。

调用关系

1.首先index.html（项目唯一的Html，是整个单页应用的用户入口）:定义一个空的div，它的id为app，等待被vue渲染。

2.然后main.js：新建一个Vue对象，el关联id为app的元素，就是index.html中定义的div，

router关联了vue-router进行组件导航。

导入App组件，并在模板中使用，将来会引入到div#app的内部。

3.index.js：router目录下的index.js，定义了路由配置，有router的path、组件位置、当我们访问到/index/dashboard，会指向dashboard组件，这个组件会在App.vue定义锚点位置渲染（<router-view>）。

4.App.vue（这个组件，是一切其他自定义组件的祖宗（意思是最高级的））：App.vue虽然是空的，没有任何数据，但是却定义了vue-router的锚点（<router-view>），显然：以后所有组件都会渲染在这个位置。

最终总结：一切路由后的内容都将通过App.vue在index.html中显示。