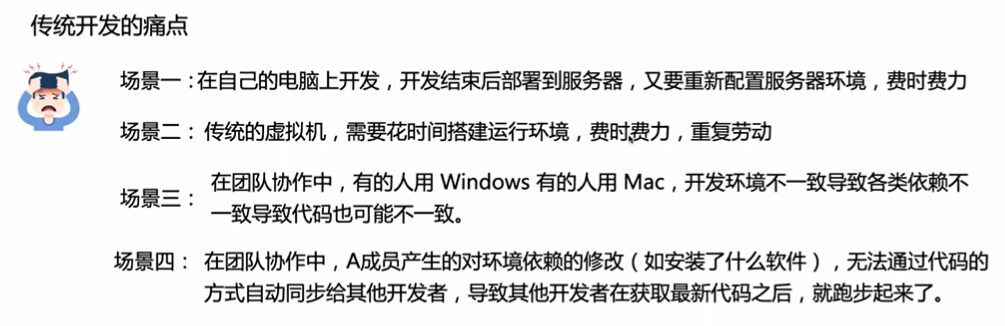
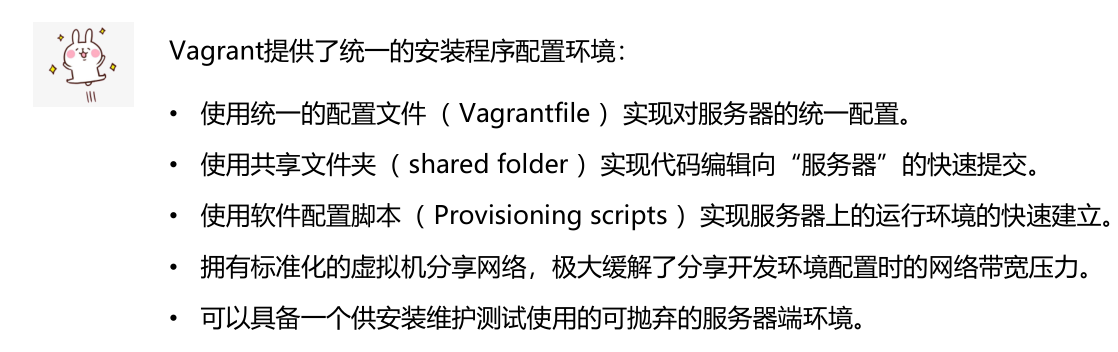
Vagrant的主要作用：同步所有开发者的环境，并将本地和服务器上的环境同步。

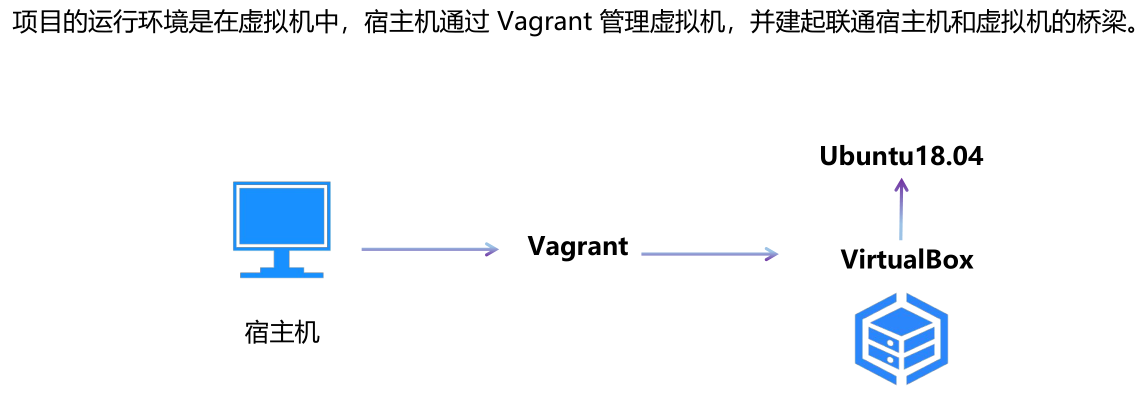
传统开发的缺点：



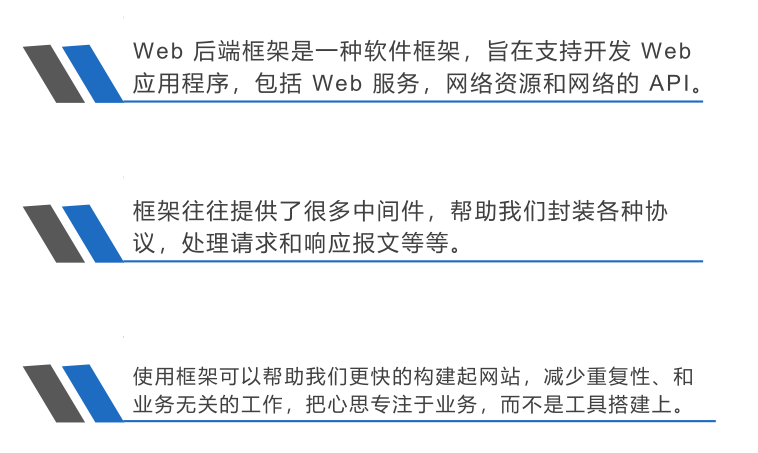
Vagrant的优点：



Vagrant一般只用于开发环境，如果希望部署在服务器上，可以考虑使用Docker。

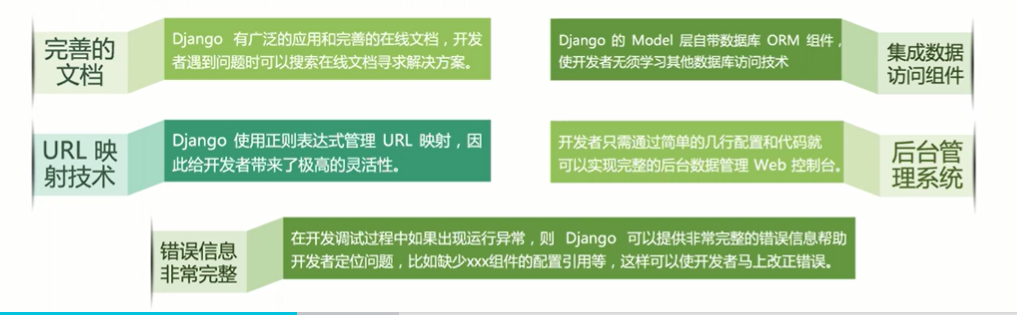


Web Framework：类似一套汽车的生产线，而不仅仅是轮胎，发动机等



Django rest framework自带有admin后台模块，登陆模块，用户模块，不需要关系所有网站都要做的事情。

Django相比于其它Python框架，功能最完整，定义了服务发布、路由映射、模板编程，数据处理的一整套功能。

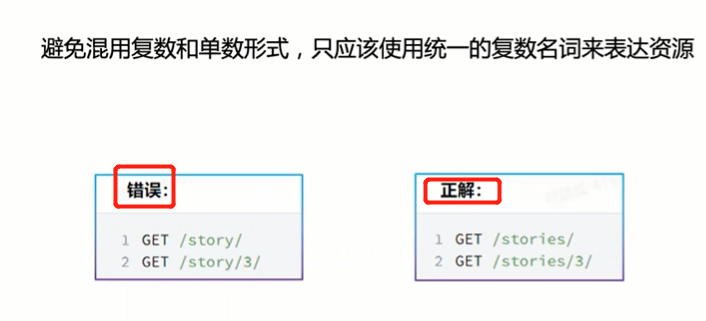


不用考虑javascript的后台框架。

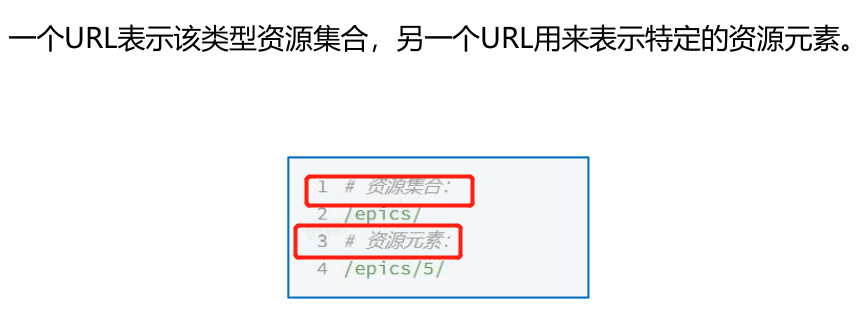
Restful API：表述性状态转移API，对API设计提出了规范(但不是强制的)。

设计规范：

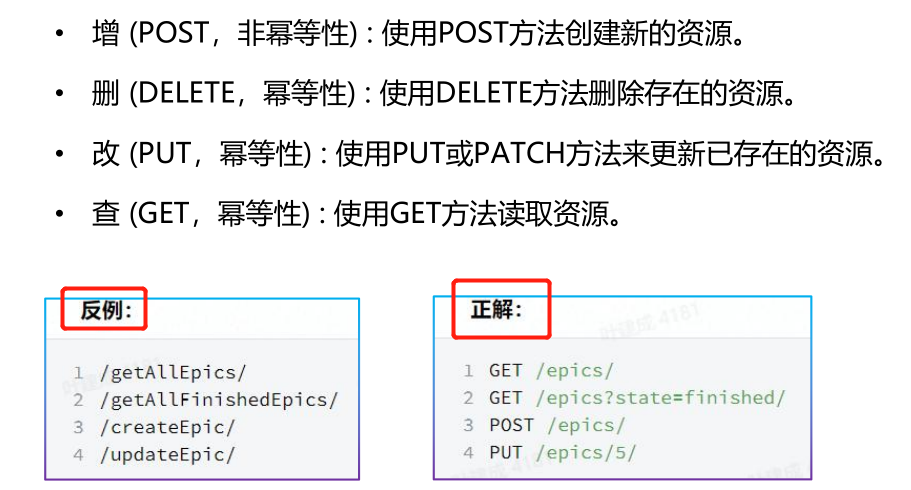
1. 使用一致的复数名词



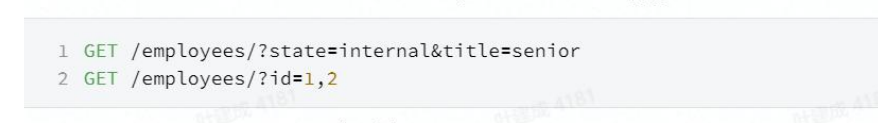
1. 一类资源可以用两类URL



1. 使用名词而不是动词



1. 将实际数据包装在data字段中
2. 对可选及复杂参数使用查询字符串(?)



Django rest framework：Serializer，ViewSet，Restful URL的映射

所有HTTP请求都会把浏览器上的所有Cookie都带上。

login：登陆的状态首先会记录在浏览器端(客户端)，也会记录在服务器上，这是一个安全的验证。

登陆的信息，服务器端记录在session，客户端记录在cookie。

session：django中服务器端记载的登陆信息有以下3个

1. session\_key：Primary key，唯一标识用户登陆新，可以理解为token，会被作为Response中的信息传递给客户端(审校的客户端就会存储token用于验证用户是否处于登陆状态)，客户端为记录在cookie中。两次登陆时，session\_key就会不一样，所以他不能作为Primary key
2. session\_data：至少要包含user\_id，这样才能知道登陆的是谁。
3. expire\_date：超时时间，什么时候登陆过期

部分网站会提示“是否接受cookie”，如果接受，则网站有可能访问一下跨域的cookie信息。

数据库中的Primary Key即主键的列是用户不能改的，像username这些尽管是UNIQUE，但它是可以修改的，所以不能用来做主键，也不能存储在session\_data中(如果存储在session\_data中，一旦修改后，session\_data会发生变化，服务器就会把用户登出)。

如果用户过多，对之前存储的数据库进行拆分，如何保证新建用户时，自动增加的user\_id还是全局唯一的？

将原始的数据库拆分到m个数据库中存储在不同机器上，创建一个manager数据库，不存储用户信息，专门用来计数，记录user\_id已经增加到数，然后获取新的user\_id，但不在manager中创建，而是根据某个规则，到拆分的某个数据库中实际创建。