# 前后端分离

## 优点

* 前后端解耦，接口复用（前端和客户端公用接口），减少开发量。
* 各司其职，前后端同步开发，提升工作效率。定义好接口规范。
* 更有利于调试（mock），测试和运维部署

## restful

Representational State Transfer

* 表现层状态转移
* 资源，表现层，状态转化
* 是一种以资源为中心的web软件架构风格，可以用ajax和restful web服务构建应用。
* resource:使用URI指定的一个实体
* representation:资源的表现方式，比如图片、HTML文本等
* State Transfer: GET，POST，PUT，DELETE,HTTP动词来操作资源，实现资源状态的改变。
* 所有事物抽象为资源，资源对应唯一的标识（identifier）
* 资源通过接口进行操作实现状态转移，操作本身是无状态的
* 对资源的操作不会改变资源的标识