

前后端分离架构设计

以旺财项目为例

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究法律责任。

联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

代码准备

- 使用已有代码

- ✓ 每节课简介里会有初始代码，你可以下载初始代码
- ✓ 下载代码后根据 README 的指引安装依赖
- ✓ 我会在初始代码的基础上加代码
- ✓ 课程结束后，提供完整代码，放在简介里

- 使用上节课代码

- ✓ 如果你上节课跟着做了，那么可以直接使用你的代码
- ✓ 需要给我的初始代码做一下对比，看看我的 commit

- 盗版用户

- ✓ 拿不到代码，因为我会在一段时间后改变初始代码地址
- ✓ 请向盗版者要求退款

如何进行架构设计

- 依据

- ✓ 用户需求
- ✓ 团队配置
- ✓ 技术成熟度

- 用户需求

- ✓ 可登录的增删改查系统

- 团队配置

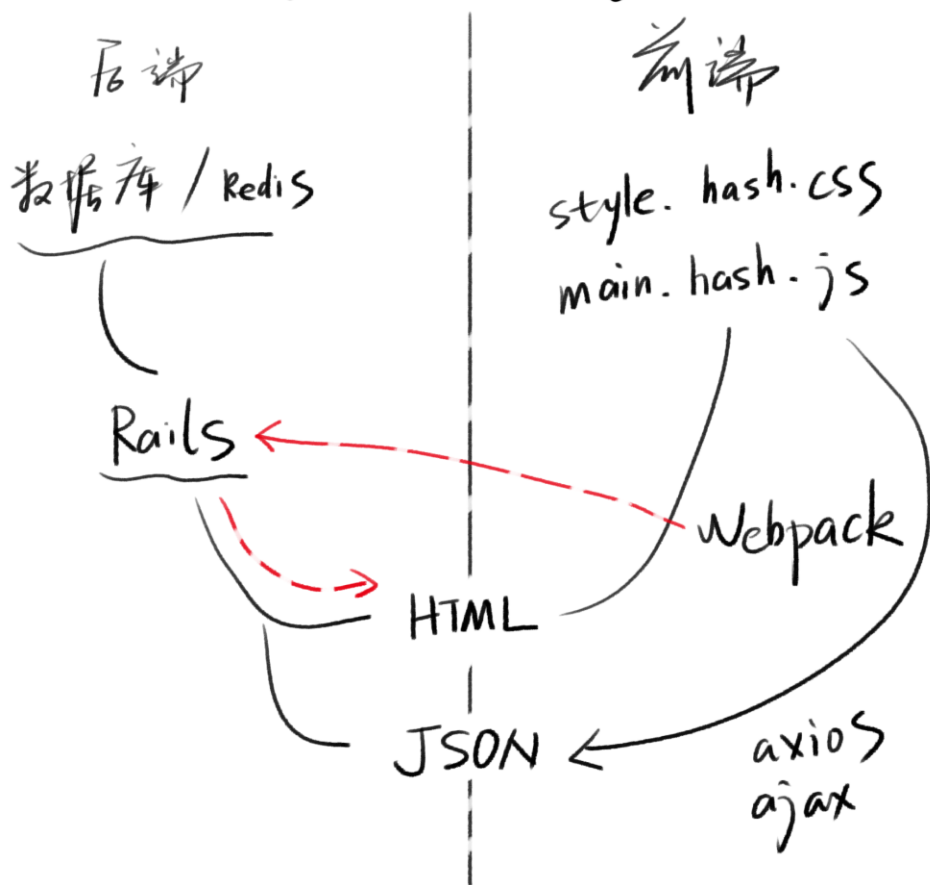
- ✓ 会前后端的全栈工程师

- 技术成熟度

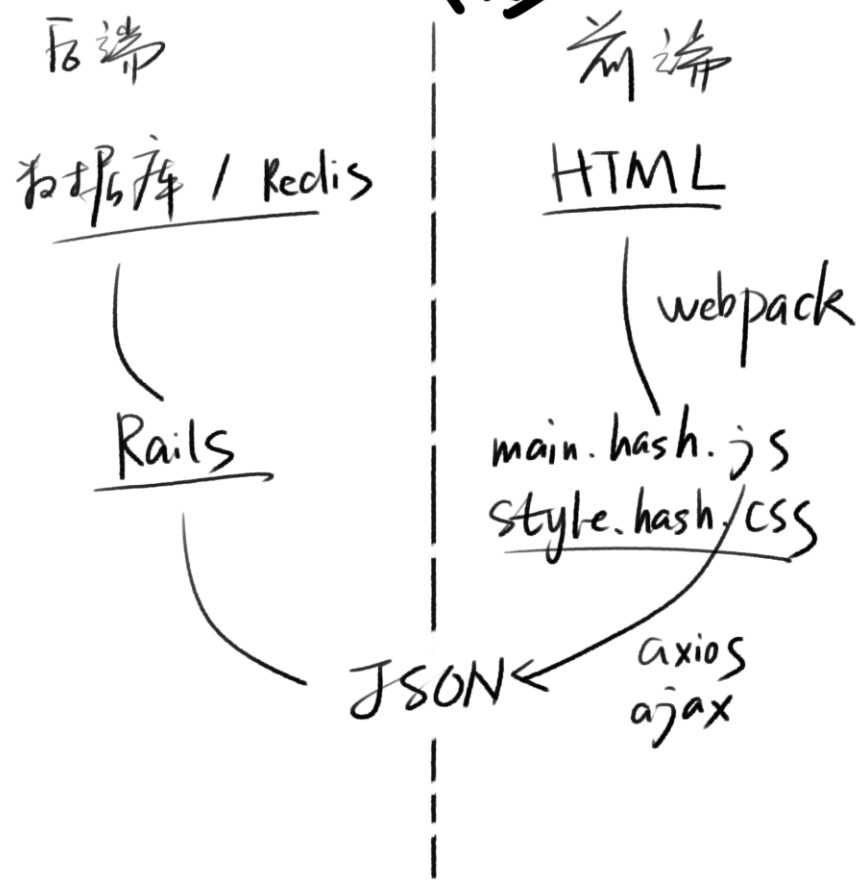
- ✓ 假设只会 Rails + Vue / React

什么是前后端分离

不分离



分离



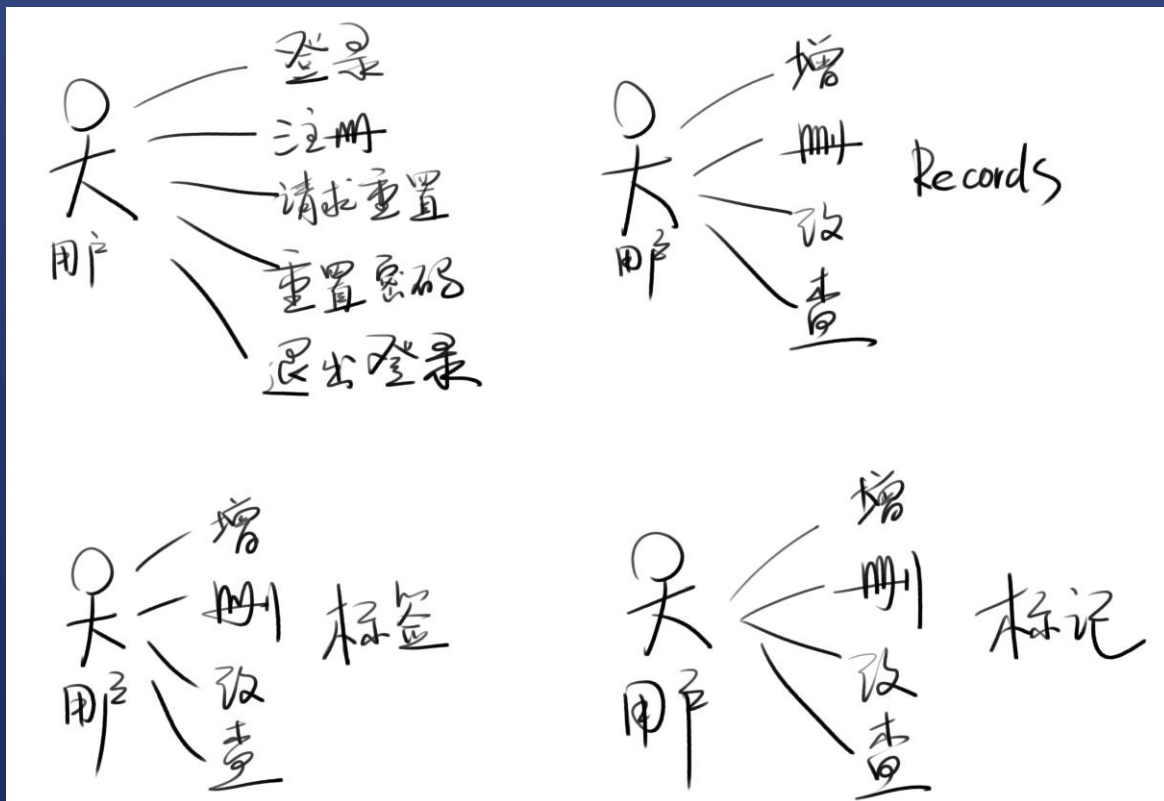
注意：前后端代码不一定要交给两个人写，可以由一个人写

用户需求

✓ 视觉稿

• 用例图

✓ 维基百科释义：把使用的例子全部画出来



表设计

- 有哪些表？

- ✓ users / records / tags
- ✓ 还有 taggings 表，表示 records 和 tags 的关联
- ✓ password_reset_requests 记录所有重置请求

- 每个表有哪些属性

- ✓ 不用一次想好，但命名一定要准确
- ✓ 从简单的需求出发，逐渐迭代

users 表

- 从需求出发

- ✓ 注册：邮箱、密码、确认密码
- ✓ 数据库需要存确认密码吗？不需要
- ✓ 数据库需要存密码吗？不需要，只存密文
- ✓ 如何加密？使用最佳实践 `has_secure_password`
- ✓ 密文叫什么？使用最佳实践 `password_digest`
- ✓ 注册之后需要发欢迎邮件吗？使用 `mailer`
- ✓ 需要强制用户验证邮箱吗？可以不强制，也可以强制

- 结论

- ✓ users 表含有 `email` 和 `password_digest`

开始实现

• 步骤

- ✓ 创建 model
- ✓ console 操作 user
- ✓ 创建 controller
- ✓ 配置 routes
- ✓ 配置 mailer
- ✓ 创建 mailer
- ✓ 使用 HttpClient 测试
- ✓ 创建 rspec 测试用例
- ✓ 改代码，测试
- ✓ 改代码，测试
- ✓ 改代码，测试

• 约定

- ✓ RESTful 接口风格，不需讨论
- ✓ post /users 就是注册接口，不需要讨论，不能不加 s
- ✓ 文件目录不需要讨论