

独立开发者训练营

(全栈)

—— 王皓@宁皓网

目录

介绍

训练方法	9
自主学习法	9
即学即用法	9
系统学习法	9
任务分布	10
项目介绍	11

技术栈

应用	17
Vue.js	17
Flutter	17
Node.js	17
Express.js	18
Nest.js	18
Socket.io	18
语言	18
JavaScript / TypeScript	18
HTML	19
CSS	19
Dart	19
SQL	19
数据	19
MySQL	19
工具	20
VSCode	20
Git	20

终端/Cmder	20
Insomnia	20
TablePlus	20
Github.....	20
运维	21
CentOS / Ubuntu.....	21
Nginx.....	21
阿里云	21
Let's Encrypt	21
服务	21
微信开放平台.....	21
微信支付.....	21
支付宝	21
 Node.js 服务端应用开发训练路线	
Node.js 服务端应用开发实践	23
准备开发.....	23
管理代码.....	23
程序语言.....	24
请求响应.....	24
网络基础.....	25
管理包裹.....	25
应用框架.....	26
交换数据.....	26
工作流程.....	26
设计架构.....	27
数据仓库.....	27
存取数据.....	28
注册用户.....	29
用户登录.....	29
访问控制.....	30
上传文件.....	30
处理图像.....	30
内容标签.....	31
发表评论.....	31
资源关系.....	31
内容列表.....	32

内容分页.....	33
用户头像.....	33
用户点赞.....	33
资源接口.....	34
自动测试.....	34
云服务器.....	35
操作系统.....	35
生产环境.....	35
部署应用.....	36
Node.js 服务端应用开发实例	36
修复改进.....	36
实时服务.....	37
实时应用.....	37
内容搜索.....	37
内容审核.....	38
审核日志.....	38
访问日志.....	38
接口统计.....	39
微信登录.....	39
产品模块.....	40
支付方法.....	40
订单模块.....	40
下载许可.....	41
订阅会员.....	41
下载模块.....	42
支付地址.....	42
准备支付.....	42
微信支付.....	43
支付宝	43
修复改进.....	43
订单列表.....	43

Vue.js 前端应用开发训练路线

Vue.js 前端应用开发实践	45
理解框架.....	45
组件基础.....	45
表单元素.....	45
高级组件.....	46
应用路由.....	46

状态管理.....	46
过渡动画.....	47
反应系统.....	47
网络请求.....	48
身份验证.....	48
上传文件.....	48
应用架构.....	48
Vue.js 前端应用开发实例.....	49
准备项目.....	49
页面布局.....	49
应用主题.....	49
身份验证.....	50
通知提醒.....	50
用户头像.....	50
用户页面.....	51
用户帐户.....	51
用户菜单.....	52
内容列表.....	52
列表布局.....	52
滚动加载.....	53
内容页面.....	53
照片参数.....	53
内容过滤.....	53
内容导航.....	54
页面侧板.....	54
评论列表.....	54
回复列表.....	55
制造工具.....	55
生成模块.....	56
创建评论.....	56
管理评论.....	57
管理回复.....	57
用户评论.....	57
创建内容.....	57
内容标签.....	58
修复改进.....	58
上传文件.....	58
内容点赞.....	59
管理列表.....	59
修复改进.....	59
管理内容.....	60

编辑内容.....	60
实时应用.....	60
修复改进.....	61
内容搜索.....	61
发布状态.....	62
管理模式.....	62
审核内容.....	62
统计仪表.....	63
创建用户.....	63
登录界面.....	63
微信登录.....	64
内容侧板.....	64
照片下载.....	64
支付方法.....	65
准备产品.....	65
准备订单.....	65
会员订阅.....	66
订阅支付.....	66
订阅历史.....	67
许可列表.....	67
订单列表.....	67
订单过滤.....	68
修复改进.....	68

Flutter 客户端应用开发训练路线

Flutter 客户端应用开发实践	69
开发环境.....	69
数据类型.....	69
函数.....	69
控制流程.....	70
类.....	70
异步编程.....	70
包管理.....	70
基础部件.....	71
页面结构.....	71
定义部件.....	71
弹窗对话.....	72
界面布局.....	72
表单元素.....	72
路由导航（一）	73
状态管理（一）	73

路由导航（二）	73
网络请求	74
内容列表	74
内容项目	74
内容页面	74
身份验证	75
状态管理（二）	75
内容点赞	75
列表布局	75
创建内容	76
文件上传	76
主题样式	76
发布应用	77
创建用户	77
Flutter 客户端应用开发实例	77
状态管理（三）	77
内容列表（二）	78
滚动加载	78
内容页面	78
内容导航	79
图像数据	79
内容底板	79
图像视图	79
沉浸模式	80
评论列表	80
回复列表	80
修复改进	81
用户信息	81
创建评论	81
编辑评论	82
创建回复	82
编辑回复	82
实时功能	83
实时评论	83
用户档案	83
用户帐户	84
内容过滤	84
修复改进（二）	84
用户页面	85
用户评论	85
应用搜索	85

创建内容（二）	86
内容标签	86
内容编辑	86
Nest.js 服务端应用开发训练路线	
Nest.js 服务端应用开发实例	88
准备项目	88
应用构架	88
用户注册	88
用户登录	89
身份验证	89
用户头像	89
图像处理	89
用户帐户	90
内容管理	90
权限控制	90
文件上传	91
文件服务	91
内容标签	91
发表评论	92
评论回复	92
内容列表	92
资源接口	93
内容审核	93
资源搜索	93
访问日志	93
访问统计	94
实时服务	94
报名参加	95
报名咨询	95

介绍

这次独立开发者训练营，目的是让大家能够从零基础开始直到成为一名全栈的独立开发者，获得设计与开发全平台应用的能力，包含 Web 端、移动端（iOS 与安卓）与服务端。训练结束以后，你将可以独立做出自己想要的應用，创造属于自己的事业。不得不说，这个真的有点酷。

训练方法

自主学习法

本次训练仍然主要参用我们的传统训练方法，自主学习+即时答疑。我们现在已经准备好了大部分训练内容（以视频内容为主），大家可以自主安排时间跟着训练，在这过程中遇到任何问题或疑问，都能得到即时反馈（通过微信或其它工具），这会为您节省大量的时间，提高学习效率。

使用这种方法学习有两个挑战，一是需要到处收集学习资料，并且要做出判断哪些是好的哪些是不好的，二是遇到问题的时候需要花几个小时，有时甚至要用几天时间解决。我们的训练营解决了自主学习中的这两个主要问题，你不再需要花时间收集学习资料，训练营的学习资料都是精心设计过的，遇到问题的时候能够即时得到反馈并解决。

即学即用法

去年我们发现有些同学参用了一种即学即用的方法，这也是一种很好的学习方法，他们会一边跟着训练内容学习，然后在另一边马上去做自己想要的东西，而且最终真的完成了他们想做的东西。这期间他们会问一些跟自己项目相关的问题，我们也都给出了回应并解决了问题。所以您如果也想使用这种即学即用的训练方法，我们也是支持的，也就是无论是您在学习中遇到的问题，还是在做自己的项目过程中遇到的问题都能够得到解决。

系统学习法

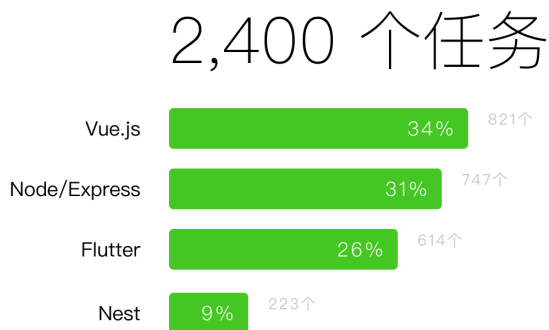
通过多年调查与自身经验，我们发现想要真正学会应用开会，必须要经过系统训练，而且必须要完成一个系统（应用）的开发。系统学习法是我们现在最推崇的一种方法，只有通过系统性的训练完成了一个系统的开发，才有可能真正学成。这次训练营提供的实际上就是一套训练系统，可以帮大家系统地掌握如何开发一个系统的能力。

系统的训练应该包含开发工具，环境，语言，概念，方法，流程以及架构，要通过开发应用系统来锻炼编程思维并解决实际问题。在一个应用系统里面会包含一些小的系统，比如内容管理、权限管理、身份验证、文件处理以及在线支付等等。

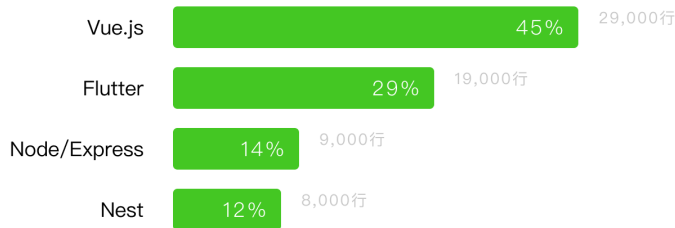
作为开发者，软件工程师，我们的任务就是想办法让不同的零部件组合在一起形成一个系统来按照我们预期那样运行。所以对于初学者来说，单纯的只学某个工具、语言或者框架是不够的，这些只是零部件，在脑子里还没有“开发系统”之前，很难利用这些零部件做出自己真正想要的东西，这样做效率不高，很快也会忘掉。所以我们一定，也必须要强迫自己做系统性的训练，在脑子里安装一套开发系统，这时再去学习各种零部件，就会变得有意义了，遇到具体需求的时候你可以随时想到它们。

任务分布

整个训练我们要完成 2,400 个任务（每个任务都有对应的视频讲解与演示），我们会用几种不同的语言（HTML、CSS、TypeScript/JavaScript 还有 Dart）编写大概 65,000 行代码。



65,000 行代码



项目介绍

在本次训练营中，我们将要构建一个数字资产订阅与交易平台，用户可以选择订阅平台的服务，这样用户在一定期限内使用平台内的所有数字资产，也可以选择单独购买某一个数字资产的使用许可。

创作者创作的任何以数字形式存在的东西都可以称为数字资产，比如音频、视频、设计图、照片、插画、模型、模板、代码等等。我们的任务就是去构建一个数字资产的交易平台，创作者可以上传自己的数字资产，平台用户可以订阅平台服务或者单独购买。

订阅是一种经过验证的商业模式，一般指的就是用户在订阅了某项服务以后，可以在一定期限内使用这项服务，比如最常见的视频网站的会员就是一种订阅服务。在我们要做的这个项目里，需要构建一套订阅系统，用户支付订阅费用以后，可以在订阅期限内任意下载使用平台内所有的数字资产，平台支持用户续订或升级订阅。通过这种订阅服务得到的收入叫 ARR（Annual Recurring Revenue），年度经常性收入。

平台还需要支持销售单个数字资产，也就是用户可以单独购买他需要的在平台里的某一个资产，支付成功以后可以得到这个资产的永久的使用许可。

这种数字资产订阅与交易平台已经有一些比较成功的案例了，比如最早我知道有一个叫 Envato 的平台，一开始这个平台就是一个用 WordPress 做的博客，后来一直发展的很好，以前我在这个网站上还买过一些东西，比如网站模板，配乐，还有一个视频。现在这个平

台也推出了订阅服务。

Envato 这个平台在 20 年到 21 年的销售总收入是 14 亿，利润是 2.3 亿，从平台创立到 20 年底的时候，平台上面的 8 万多个创作者得到的收入是 60 多亿，其中有 88 个人赚到了 600 万以上。

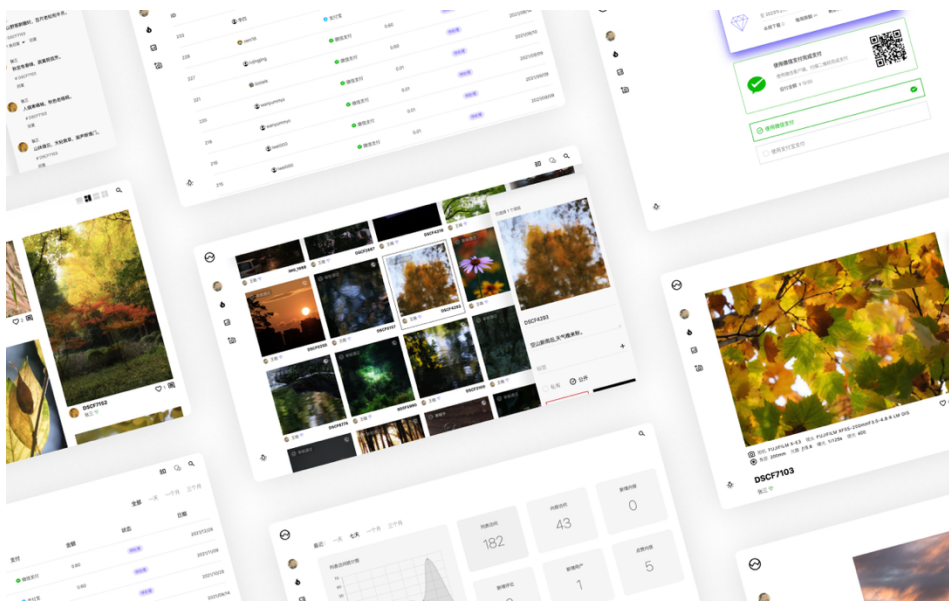
还有一个叫 Artlist 的网站，他们主要做的是视频配乐还有音效这种数字资产，创作者可以上传自己的作品，平台支持订阅服务，用户订阅以后可以获得平台内所有音频的使用许可，根据这些数字资产的具体的使用情况，创作者可以得到相应的分成。这家网站在 20 年的时候获得了 3 亿的融资。

在宁皓网的独立开发者训练营里要做的就是一个数字资产订阅与交易平台，以照片这种数字资产作为实例，创作者可以上传自己的摄影作品，用户订阅以后可以下载使用平台内所有的原版照片，用户也可以单独购买某一张照片的下载与使用许可。

这个项目可以非常复杂，所以在前期我们只需要先做一个 MVP (Minimum viable product)，直接翻译过来就是最小化的可行产品。就是为了快速的验证想法，得到用户的反馈，发布的一个简化版的应用。在训练营里我们会实现一个 MVP，完成这个产品的核心部分，比如订阅与支付系统，购买单个许可，用户注册与登录，资产的发布与管理，用户的权限控制等等。

我们先选择照片这种数字资产，后期如果你对这个项目感兴趣，可以继续去扩展这个平台。完成训练以后，你就已经获得了独立的设计还有开发能力，所以想怎么做都是可以的。比如可以扩展这个平台的功能，可以让它支持一些其它的资产类目，像是音频、视频、模型、模板、文章、电子书等等。

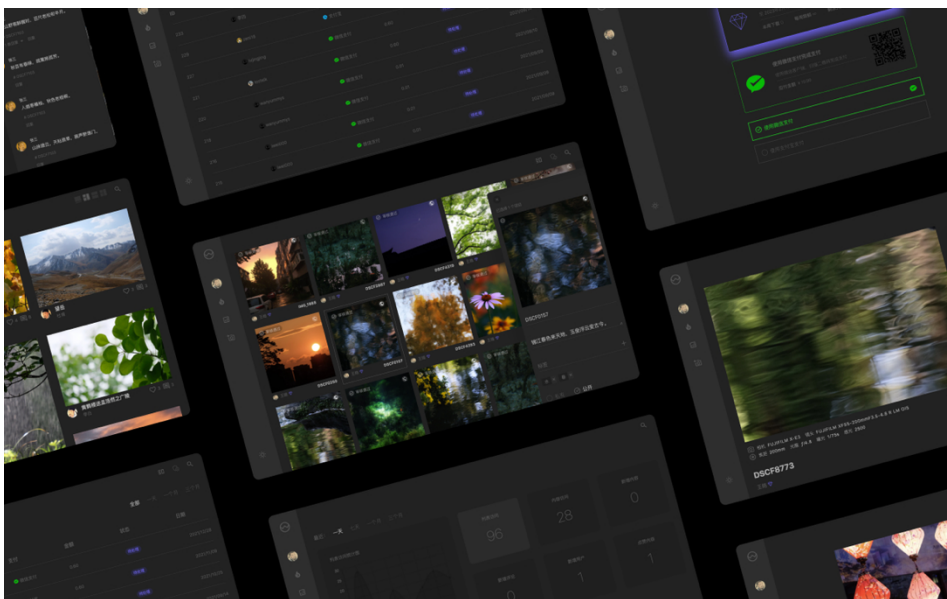
Web 端应用视频演示：<https://www.bilibili.com/video/BV1TZ4y127ZN/>



Web 端应用亮色主题



移动端应用亮色主题



Web 端应用暗色主题



移动端应用暗色主题

技术栈

在本次独立开发者训练营中，所有训练将围绕一个数字资产交易与订阅平台项目展开，我们会用到六个应用框架，五种语言，四个主要开发工具，三个三方服务，一个数据仓库管理系统，一个服务器操作系统。

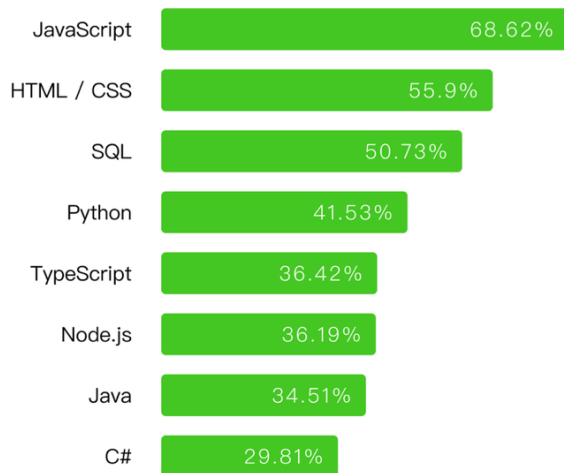
根据著名开发者网站 Stack Overflow 的调查，在最受专业开发者喜欢的排名前十的技术里面，我们将在本次训练营中学会其中五项，分别是排名前三的 JavaScript、HTML/CSS 与 SQL，排名第五的 TypeScript，还有排名第六的 Node.js。调查显示有 50% 的开发者表示 MySQL 是他们最常用的数据库，这也正是我们在这次训练营中选择的数据库系统。

训练营项目所选的技术栈，不但深受专业开发者的喜欢，而且拥有庞大的社区支持与市场需求，大小厂商也都将其用于生产。比如目前估值 200 亿的 ClickUp，它们的服务端技术用的就是 Node.js，数据服务主要用的是关系型数据仓库。微信公众平台、B 站、Upwork 还有 [新宁皓](#) 等这些平台在用 Vue.js 做前端应用。字节、腾讯、美团、阿里等几乎所有大厂都在生产环境中使用 Flutter 框架构建移动端应用。所有这些技术都会在我们这次独立开发者训练营中用到。

Stack Overflow

最受开发者喜欢的技术

来自 58,031 个专业开发者的调查结果



数据来源于 Stack Overflow 网站的调查结果，[查看原文](#)。

下面是在本次独立开发者训练营中所用的技术栈，可以先简单了解一下我们将要使用的各种应用框架、语言、工具、数据仓库、第三方服务还有一些运维技术。



应用

Vue.js, Flutter, Node.js, Express.js, Nest.js, Socket.io。

Vue.js

前端应用框架。在训练营中我们将会用到这个构架构建应用的 Web 端界面，也就是通过浏览器使用的客户端应用，应用需要的界面组件全部手写，包括组件的逻辑与样式，不依赖任何第三方组件库与样式库，这样做能够帮我们更好地理解基于 Vue.js 的前端应用开发，也能学会使用 HTML 与 CSS。

学会 Vue.js 框架还有个隐藏的技能，就是开发各个平台的小程序，通过一些技术我们可以使用 Vue.js 的方式开发各个平台的小程序。也就是如果你能用 Vue.js 开发前端应用，也就可以用它开发小程序。

Flutter

跨平台客户端应用构架。Flutter 支持使用同一种方法为不同平台开发客户端应用，在训练营中我们会用基于 Flutter 构建应用的移动端，也就是可以安装在安卓与 iOS 平台上的 App。

Node.js

服务端应用技术。Node.js 支持我们用 JavaScript / TypeScript 语言开发服务端应用，也就是在服务器上运行的应用，它可以给应用的客户端提供各种服务，比如存储数据，处理文件，调用第三方服务等等。

Express.js

基于 Node.js 的轻型服务端应用框架。Express.js 是一个轻型的应用框架，提供了最基本的中间件与路由等组件，在 npm 网站每周下载量达到两千多万，应该是最受欢迎的 Node.js 应用框架。在本次训练营中，我们将基于这个应用框架构建应用的服务端，为客户端应用提供身份验证，处理数据，内容管理，订阅管理与处理支付等服务。

Express.js 能做的事情其实也可以直接使用 Node.js 做到，不过它简化了很多工作，比如让定义路由，接口与中间件等工作变得更方便了。我们可以把它想成是 Node.js 里的一些功能包（Package）。

Nest.js

基于 Node.js 的高级服务端应用构架。Nest.js 与 Express.js 都属于 Node.js 服务端应用框架，相比 Express.js，Nest.js 会给我们提供更多的零部件与更好的开发方法，它能让我们开发的应用更规矩也更稳定，适用于开发中大型或企业级的服务端应用。

在训练营中我们将使用 Nest.js 重构部分服务端，应用架构将采用扩展性极强的 CQRS 模式，更方便的数据处理，更规矩的数据验证方式与更灵活的策略型权限检查。

Socket.io

客户端与服务端实时通信构架。客户端与服务端一般的交流方式是请求与响应，客户端请求一次，服务端做出一次对应的响应，不过有时候我们的应用需要双向实时的沟通，也就是客户端有什么事情可以实时告诉给服务端，服务端发生的事情也可以实时的通知给客户端，Socket.io 这个框架可以帮我们实现这种沟通。

我们在训练营中开发的应用会在有用户赞了某个内容或者发表了新的评论时，实时地更新所有用户正在使用的客户端界面，比如内容的点赞状态与数量，或者实时显示其它用户新发表的评论内容等等。

语言

JavaScript / TypeScript, HTML, CSS, Dart, SQL。

JavaScript / TypeScript

程序语言。JavaScript 是最受开发者欢迎的程序语言，能够运行使用这种语言的环境非常多，不过主要还是浏览器，还有安装了 Node.js 的服务器。我们要开发的 Web 应用（前端应用）与基于 Node.js 的服务端应用都会用到这种程序语言。如果你是想学习应用开发的新手，强烈建议用 JavaScript 作为你的第一门程序语言。

TypeScript 可以想成是增强版的 JavaScript，它在 JavaScript 语言的基础之上，又给我们提供了一套类型系统，能帮我们减少开发中的错误。我们在训练营中开发的前端应用与基于 Node.js 的服务端应用，都会选择使用 TypeScript。

HTML

标记语言。在训练营中开发前端应用时，会用到 HTML 这种标记语言，构建界面的内容与骨架。

CSS

样式语言。在训练营中开发前端应用时，会使用 CSS 这种样式语言，设计界面的布局与样式。

Dart

程序语言。在训练营中基于 Flutter 应用构架构建移动端应用时，我们会用到 Dart 这种程序语言。

SQL

数据查询语言。在处理关系型数据仓库里的数据的时候，会用到 SQL 这种语言，比如将数据放到数据仓库里，或者从数据仓库里提取需要的数据，都需要通过 SQL 这种语言来表达。在训练营中，基于 Express 构架构建服务端应用时，我们会手写一些 SQL，应用会将它们交给数据仓库那里去处理。

虽然在开发服务端应用时，有很多方法可以不用与 SQL，直接用一些现成的方法就可以处理关系型数据仓库里的数据，不过还是强烈建议大家首先要理解使用 SQL，这样能够更好地解决后续遇到的各种复杂的问题。在基于 Nest.js 框架开发应用的训练中，我们就不再直接使用 SQL 了，而是通过 ORM 提供的方法来处理应用的数据，比如用 Repository 或者查询构建器等方法。

数据

MySQL 是此次训练营主要使用的数据库管理系统。

MySQL

关系型数据库管理系统。MySQL 的历史悠久，也是专业开发者最广泛使用的数据库管理系统。我们在训练营中开发的应用会选择使用这种数据库管理应用的数据，比如用户数据，用户发布的内容与评论，订单，还有访问日志等等。

工具

VSCode, Git, 终端 / Cmdr, Insomnia, TablePlus, Github。

VSCode

代码编辑器。VSCode 是微软公司做的一款开源的代码编辑器，可自由定制与扩展，是目前非常流行的编辑器。我们在训练营中会使用这款编辑器编写应用的所有的代码。

Git

版本控制（源代码管理）工具。学会使用源代码管理工具，对开发的项目做源代码管理是开发者必须要做的事情。Git 是目前最常用的源代码管理工具，它是在我们在训练营中要学会使用的开发工具之一，它也是无论如何都要掌握的重要的开发工具。

终端/Cmdr

命令行界面。学会在命令行界面下使用命令工具，是我们这次训练营里的重要任务之一，我们需要在命令行界面下去创建项目，给项目安装功能包，生成项目需要的各种组件，做源代码管理，远程管理服务器等等。

Insomnia

接口调试与测试工具。开发服务端应用时需要定义一些功能接口，客户端应用可以使用这些功能接口去做一些事情，在开发这些接口或者使用这些接口的时候，我们需要测试接口或者调试这些接口，这就需要用到接口测试与调试工具，在本次训练营中我们会使用 Insomnia 来测试与调试应用的接口。

TablePlus

数据库客户端。在本次训练营中我们会使用 TablePlus 这个数据库客户端软件，它可以用图形化界面展示与处理数据仓库里的数据，比如我们可以通过 TablePlus，直接查看在应用的数据仓库里都有哪些数据，查看数据表的结构，也可以直接通过图形化界面添加、更新或者删除数据表里的数据记录。

Github

远程代码仓库提供商。我们平时在本地电脑上开发项目，使用 Git 对项目做源代码管理，另外我们还可以找一个远程代码仓库提供商，在他们那里给项目创建一个远程仓库，这样团队的其它成员可以通过这个远程仓库获得我们对项目做的最近的修改，成功也可以贡献他们自己的修改。Github 就是一家远程代码仓库提供商，在训练营中我们在 Github 那里给项目创建一个远程仓库，然后把我们在本地对项目做的修改推送到这个远程仓库里。

运维

CentOS / Ubuntu, Nginx, 阿里云, Let's Encrypt。

CentOS / Ubuntu

Linux 操作系统。在训练营中开发的服务端应用会部署到一种安装了 Linux 操作系统的服务器上，在这种类型的操作系统里面，我们要搭建应用的生产环境。CentOS 与 Ubuntu 都属于 Linux 这种类型的操作系统，由于 CentOS 在未来将不再更新，所以现在推荐用 Ubuntu 替代。

Nginx

Web 服务器。Nginx 是一款成熟的 Web 服务器，在训练营我们会使用 Nginx 提供的功能，给我们的前端应用创建一个 Web 服务器，给服务端应用创建一个反向代理服务器，用它接待来自客户端的请求。

阿里云

云服务提供商。在训练营我们会选择使用阿里云提供的服务器，给我们的服务端应用提供存储与计算能力。

Let's Encrypt

加密证书签发商。如果你想让应用支持使用 HTTPS 协议，那就需要申请加密证书，并且配置好应用的 Web 服务。在训练营中我们会使用 Let's Encrypt 签发的免费的加密证书。

服务

微信开放平台，微信支付，支付宝。

微信开放平台

在本次训练营中开发的前端应用里，我们会使用微信开放平台提供的微信登录功能，用户可以通过微信客户端扫码完成登录。

微信支付

第三方支付平台。在训练营中开发的应用里，我们会集成微信支付功能，允许用户使用微信支付完成订阅与数字资产交易。

支付宝

第三方支付平台，在训练营中开发的应用里，我们会集成支付宝的支付功能，允许用户使用支付宝完成订阅与数字资产交易。

Node.js 服务端应用开发训练路线

Node.js 服务端应用开发实践

准备开发

Windows: 准备命令行界面<Cmder>
Windows: 准备命令行界面<Cmder>.resources
Windows: 配置 Cmder
Windows: 熟悉基本的命令
Windows: 理解 cmd 与 bash 的区别
Windows: 知道命令来自哪里
macOS: 准备命令行界面<终端>
macOS: 配置终端
macOS: 熟悉基本的命令
macOS: 知道命令来自哪里
macOS: 知道命令来自哪里.resources
macOS: 命令行界面的配置文件
Windows: 安装 VSCode
Windows: 安装简体中文语言包
打开项目<VSCode>
定制编辑器<VSCode>
Windows: 安装 Node.js
macOS: 安装 Node.js
创建一个服务端应用项目<Node.js>
运行应用<Node.js>

管理代码

管理代码
配置 Git
初始化仓库<Repository>
做一次提交<Commit>
带图形界面的 Git 工具<Source Tree>
把文件恢复到上一次提交时的样子
创建与切换分支
创建分支<greeting>
再做一次提交
重做最后一次提交

理解分支
合并分支
配置 SSH 密钥
创建远程仓库<Github>
添加远程仓库
把远程仓库复制到本地
推送（上传）与拉取（下载）
添加 README.md 说明文档<Windows>
下载远程仓库里的新东西<macOS>
忽略不需要做源代码管理的东西<.gitignore>

程序语言

语言训练场地<JavaScript>
数据: Data
常量: Constant
变量: Variable
字符串: String
数字: Number
数组: Array
对象: Object
布尔值: Boolean
字符模板: Template strings
函数: Function
箭头函数
函数参数
多个参数
返回结果
方法: Method
解构<对象>: Destructuring
解构<数组>: Destructuring
展开<数组>: Spread
展开<对象>: Spread
if
switch
throw
try...catch
类: Class
构造方法
this
继承

请求响应

准备项目<web-server>
创建 Web 服务器<http 模块>
完成任务以后做一次提交 (Commit)
使用编辑器的源代码管理<VSCode>
请求与响应
请求: Request
响应: Response
请求的地址
JSON 数据格式
响应 JSON 格式的数据

网络基础

传输协议<HTTP>
网络地址<IP>
macOS: 知道电脑的 IP 地址
! Windows: 知道电脑的 IP 地址
知道终端设备上网用的公网 IP 地址
测试内网互联
访问在内网设备上运行的服务
主机名, 域名, DNS
知道主机名指向的 IP 地址
localhost, 127.0.0.1, hosts 文件
理解主机名配置文件<hosts>
自定义本地主机名
交流通道<Port>
理解端口的作用

管理包裹

包
依赖关系
包管理
包在哪里?
版本号
npm: 修改安装源
npm: 安装包
包的版本号里的 ^ 与 ~
npm: 安装包 (开发依赖)
带命令行工具的包
带命令行工具的包<Windows Cmder>
清理与总结

应用框架

应用框架

准备项目<Express>

filter(): 过滤数据项目

Express.js: 安装应用框架

创建 Web 服务器<Express.js>

应用接口_路由

定义应用接口(路由)

准备演示数据

定义响应 JSON 数据的应用接口

定义带参数的应用接口

中间件

理解中间件

总结与整理

整理项目

交换数据

交换数据

准备项目<http>

HTTP 客户端

Insomnia: 下载安装<macOS>

Insomnia: 下载安装<Windows>

Insomnia 的基本用法

工作区: Workspace

环境: Environment

HTTP 方法

应用资源接口

创建内容用的应用接口

在应用全局使用 JSON 中间件

定义创建内容资源用的接口<POST>

测试创建内容资源用的接口

状态码: Status Code

设置服务端响应的状态码

头部数据: Headers

在客户端的请求里设置头部数据

在服务端设置响应的头部数据

总结与整理

工作流程

工作流程

准备项目<workflow>

TypeScript

在项目中 使用 TypeScript 开发

编译应用 (tsc)

理解类型定义 (.d.ts)

@types_node

安装

使用 JavaScript 标准模块系统

安装 Express 框架的类型定义

设置处理器参数的类型

处理 TypeScript 检查出来的问题

智能提示: IntelliSense

自动编译与重启

自动排版代码

总结与整理

设计架构

设计架构

准备项目<structure>

JavaScript 模块

JavaScript 模块: 默认导出

创建应用与 Web 服务器

配置

环境变量: Environment Variables

设置环境变量的方法

把应用的配置放在环境变量里

控制器: Controller

路由: Router

服务: Service

中间件: Middleware

理解中间件

处理异常

创建默认的异常处理器

测试使用异常处理器

总结与整理

数据仓库

数据仓库

MySQL 数据仓库服务

macOS: 安装 MySQL 数据仓库系统

Windows: 安装 MySQL 数据仓库系统

数据库客户端

macOS: 配置使用 mysql 命令行工具

用 mysql 命令行工具管理数据仓库

Windows: 配置使用 mysql 命令行工具

准备一个图形界面的数据库客户端软件<TablePlus>

数据仓库

创建数据仓库

使用数据仓库

数据表格

创建数据表格<user>

创建数据表格<post>

处理数据

插入数据: INSERT

调取（查询）数据记录: SELECT

更新数据: UPDATE

删除数据: DELETE

数据关系

定义数据关系<一对多 | 多对一>

利用数据关系

总结

存取数据

存取数据

准备项目<database>

异步动作

Callback

Promise

异步函数: async, await

连接数据仓库服务

安装驱动连接数据仓库

测试连接数据仓库

调取内容

利用数据之间的关系

处理使用数据服务时遇到的异常情况

存储内容

测试创建内容接口

更新内容

测试更新内容接口

使用帮手<Lodash>

删除内容

测试删除内容接口

总结与整理

注册用户

注册用户
准备项目<register>
用户模块
注册用户
定义创建用户接口
测试创建用户接口
验证用户数据
处理创建用户的错误
定义按用户名查找用户的服务
注册时检查用户名是否存在
存储密码
Hash 用户密码<Bcrypt>
Hash 用户密码<Bcrypt>_1
总结与整理

用户登录

用户登录
准备项目<login>
身份验证模块
定义用户登录接口
测试用户登录接口
验证用户登录数据
重构按用户名查找用户
验证用户密码是否匹配
总结与整理
验证身份
准备项目<authentication>
TypeScript: interface
基于令牌的身份验证<JWT>
签发令牌时用的加密方法
制作密钥与公钥
准备密钥与公钥配置
签发令牌
定义签发 JWT 令牌的服务
签发 JWT 令牌
理解 JWT 令牌
验证令牌
定义验证用户身份中间件
测试验证用户登录状态
当前用户

在请求里添加当前用户
创建内容时记录内容的作者
总结与整理

访问控制

访问控制
准备项目<access-control>
定义检查用户是否拥有指定资源的功能
访问控制：定义访问控制中间件
配置接口的访问控制
测试访问控制
总结与整理

上传文件

上传文件
准备项目<files>
JavaScript 语法：some()
概念：地址查询符
VSCode：项目代码片断
文件模块
-上传文件
定义上传文件的功能
定义上传文件接口
测试上传文件接口
-存储文件信息
创建存储文件信息数据表
定义存储文件信息功能
存储上传文件信息
-文件服务
定义文件服务接口
测试文件服务接口
总结与整理

处理图像

处理文件
准备项目<file-process>
图像信息：读取图像文件附加信息
存储图像的附加信息
定义调取图像信息的接口
测试图像信息的接口
处理文件：调整图像文件的大小

提供不同尺寸的图像文件
总结与整理

内容标签

内容标签
准备项目<tags>
标签模块
创建标签数据表
定义存储标签数据的功能
定义按标签名称查找标签的功能
定义创建标签用的接口
测试创建标签接口
内容标签：创建保存内容与标签关系用的数据表
定义保存内容标签服务
定义检查内容标签服务
定义添加内容标签接口
测试使用添加内容标签接口
删除内容标签：定义移除内容标签服务
定义移除内容标签接口
测试使用移除内容标签接口
总结与整理

发表评论

发表评论
准备项目<comments>
评论模块
发表评论：创建评论数据表
定义存储评论数据的功能
定义发表评论用的接口
测试发表评论接口
回复评论：定义检查评论是否为回复评论的功能
定义回复评论接口
测试使用回复评论接口
修改评论：定义修改评论内容用的功能
定义修改评论接口
测试使用修改评论接口
删除评论：定义删除评论用的功能
定义删除评论接口
测试使用删除评论接口
总结与整理

资源关系

资源关系
准备工具<TablePlus>
内容，用户
找出内容的作者
找出某个用户发布的内容列表
在查询结果中组织 JSON 数据 (JSON_OBJECT)
内容，文件，用户
在内容列表里包含相关的文件
解决重复出现的内容项目
在查询中判断条件 (IF)
在内容列表里包含内容作者还有相关文件
内容，评论，用户
找出某个内容的评论列表
找出某个用户的评论列表
内容，标签
找出包含某个标签的内容列表
找出某个内容与这个内容的标签

内容列表

内容列表
准备项目<lists>
SQL 查询片段
评论总数：手工为内容添加一条评论
SQL：拼接评论数据表后统计评论总数
SQL：直接用子查询统计内容评论总数
在内容列表中包含内容的评论数量
SQL：拼接文件数据表获取内容相关文件列表
SQL：解决多表拼接时出现的重复问题
SQL：拼接文件数据表时限制内容相关文件数量
SQL：用 CAST 转换数据的类型
SQL：用 CONCAT 组织一个数组
在内容列表中包含内容的相关文件
在内容项目里添加标签列表
-SQL：设置内容列表的排序方式
改造调用内容列表用的功能
创建排序方式中间件
SQL：按用户过滤内容列表
SQL：按标签名过滤内容列表
改造调用内容列表用的功能
创建过滤列表中间件
总结与整理

内容分页

内容分页

准备项目<pagination>

SQL: 理解分页功能的基础原理

改造调取内容列表功能（分页）

添加分页配置

创建内容分页中间件

定义统计内容数量的功能

总结与整理

用户头像

用户头像

准备项目<avatar>

创建头像模块<avatar>

定义文件过滤器

创建上传头像中间件

定义上传头像接口

测试上传头像接口

调整头像的尺寸

头像数据: 创建头像数据表<avatar>

定义保存头像数据的功能

保存头像数据

定义按用户 ID 查找头像的功能

定义头像服务接口

测试头像服务接口

在内容中标记用户是否上传过头像

总结与整理

用户点赞

用户点赞

准备项目<likes>

创建点赞模块<like>

创建用户点赞内容的数据表

定义保存点赞内容的功能

定义点赞内容接口

测试点赞内容接口

定义取消点赞内容功能

定义取消点赞内容接口

测试使用取消点赞内容接口

在内容项目中包含点赞数量

SQL: 查询出某个用户点过赞的内容列表
用户赞过的内容列表
总结与整理

资源接口

资源接口
准备项目<resources>
定义按 ID 调取内容的功能
定义单个内容接口
测试单个内容接口
准备评论数据
SQL: 调取评论列表
SQL: 统计评论的回复数量
定义评论相关的查询片断
定义获取评论列表功能
定义评论列表接口
创建评论列表过滤器
获取内容的评论列表
获取用户的评论列表
获取用户的回复评论列表
在回复中包含被回复的评论数据
在评论列表中包含评论的回复数量
改造内容分页功能
评论列表分页功能
统计评论数量
定义获取评论回复列表功能
定义评论回复列表接口
测试评论回复列表接口
定义获取用户数据功能
定义用户帐户接口
测试用户帐户接口
更新用户
创建验证更新用户数据中间件
定义更新用户数据功能
定义更新用户接口
测试更新用户接口
整理项目

自动测试

自动测试
准备项目<testing>

测试框架
准备测试框架<Jest>
创建测试
创建与运行测试
准备接口测试
创建接口测试
准备测试用户接口
测试创建用户接口
测试用户帐户接口
测试更新用户接口
总结与整理

云服务器

云服务器
阿里云
准备一台云服务器
云服务器的 IP 地址与安全组
域名
申请一个域名
远程连接云服务器
添加新用户
为用户生成密钥对
用密钥的方式验证登录
禁用密码与 root 用户登录
配置使用数据盘

操作系统

操作系统
系统包管理工具
使用系统的包管理工具
熟悉管理系统里的服务
理解目录与文件权限
理解目录与文件权限.resources
观察目录与文件的权限
管理用户与用户组
理解目录与文件的权限
修改资源的拥有权
修改资源的权限

生产环境

生产环境

- 安装 Node.js
- 安装与启动 MySQL
- MySQL 安全配置
- 创建新用户与数据仓库
- 创建数据库客户端连接
- 测试使用数据仓库
- 安装与启动 Nginx
- 理解 Nginx 的配置文件
- 理解用户与工作进程
- 理解错误日志
- 理解 http 配置区块
- 理解访问日志配置
- 理解文件类型配置
- 理解自定义配置文件
- 理解服务器配置
- 配置一个 Nginx 服务器
- Docker：应用容器
- 安装 Docker
- HTTPS：加密传输
- 用 Certbot 申请 Let's Encrypt 签发的 SSL 证书
- 配置 Nginx 服务器支持 HTTPS
- 给应用创建反向代理服务
- 用 PM2 管理运行 Node.js 应用

部署应用

- 准备项目 <deploy>
- 定义编译命令
- 迁移数据仓库到生产环境
- 把本地的 master 分支推送到远程仓库
- 在远程仓库添加部署密钥
- 在生产环境上部署与运行应用
- 测试在生产环境上运行的应用
- 测试文件相关接口

Node.js 服务端应用开发实例

修复改进

- 准备项目（fix-and-improve）
- 只返回包含相关文件的内容列表

在请求中识别当前用户
在内容列表项目中标记当前用户是否赞过内容
在单个内容中标记当前用户是否赞过内容
数据库：理解数据表的外键限制
数据库：把外键删除动作修改成串联（CASCADE）
删除内容同时删除内容相关文件
修复：用户发布作品数量统计有误
整理项目

实时服务

准备项目（realtime）
添加跨域资源共享配置（ALLOW_ORIGIN）
改进创建 HTTP 服务器的方法
创建 Socket.io 实时服务器
客户端：准备项目（realtime-client）
安装配置 Socket.io 客户端
测试实时服务器与客户端之间的连接
整理项目

实时应用

准备项目（realtime-app）
实时触发点赞事件（userLikePostCreated）
实时触发取消点赞事件（userLikePostDeleted）
在事件数据里包含 Socket Id
定义按 ID 获取评论与回复的方法
实时触发评论已创建事件（commentCreated）
实时触发回复已创建事件
实时触发评论与回复已更新事件
实时触发评论与回复已删除事件
实时触发评论已删除事件（commentDeleted）
实时触发评论回复已创建事件
整理项目

内容搜索

准备项目（search）
创建搜索模块
定义搜索标签服务
定义搜索标签接口
定义搜索用户服务
定义搜索用户接口
定义搜索相机服务

定义搜索相机接口
定义搜索镜头接口
过滤出相机与镜头内容
整理项目

内容审核

准备项目 (publish-audit)
修改 post 数据表添加发布状态字段
获取指定状态的内容列表
定义验证内容发布状态的中间件
切换内容列表管理模式与普通模式
创建与更新内容时设置内容发布状态
获取内容时检查内容发布状态
检查文件权限 (fileAccessControl)
整理项目

审核日志

准备项目 (audit-log)
创建审核日志数据表 (audit_log)
创建审核日志模块
定义检查审核日志的中间件 (auditLogGuard)
定义创建审核日志接口
定义按资源获取审核日志的服务方法
限制创建同样状态的审核日志
定义删除审核日志服务方法 (deleteAuditLog)
定义取消审核接口
定义拼接审核日志与审核字段查询片断
内容列表项目审核
单个内容审核状态
文件的审核状态
整理项目

访问日志

准备项目 (access-log)
创建访问日志数据表 (access_log)
定义访问日志数据模型 (AccessLogModel)
定义创建访问日志服务方法 (createAccessLog)
定义访问日志中间件 (accessLog)
测试使用记录访问日志中间件
配置访问日志中间件记录 Payload
配置访问日志中间件记录资源 ID

记录删除内容接口的使用情况
记录添加内容标签与取消内容标签接口的使用情况
记录上传文件与获取文件信息接口的使用情况
记录评论相关接口的使用情况
记录内容点赞接口的使用情况
记录用户相关接口的使用情况
记录搜索相关接口的使用情况
记录标签相关接口的使用情况
整理项目
改进：记录单个内容接口的使用情况

接口统计

准备项目 (dashboard)
创建仪表盘模块 (Dashboard)
定义被允许的访问次数统计
SQL: 统计访问日志次数
SQL: 按动作与日期时间范围统计访问日志次数
定义访问次数过滤器中间件 (accessCountsFilter)
定义访问次数列表方法 (getAccessCounts)
定义获取访问次数列表接口
定义访问次数守卫中间件 (accessCountsGuard)
定义获取按动作分时段的访问次数的方法
定义按动作分时段访问次数接口
整理项目
修正: accessCountShow 方法里的错别字
修正: 7-day 时间范围参数
修正: createUser 动作标题错误

微信登录

准备项目 (weixin-login)
准备网站应用微信登录相关配置
准备 HTTP 客户端
创建用户 Meta 数据表 (user_meta)
创建用户 Meta 数据模型 (UserMetaModel)
定义创建用户 Meta 方法 (createUserMeta)
定义更新用户 Meta 方法 (updateUserMeta)
定义按 Info 字段值查询用户 Meta 的方法
测试用户 Meta 相关方法
定义获取微信登录访问令牌的方法
定义获取微信用户信息的方法
改进: 设置获取用户以后提供的用户数据的类型

定义微信登录后处理方法下载微信用户头像
创建微信登录模块
定义微信登录守卫中间件（weixinLoginGuard）
定义微信登录关联器中间件
定义微信登录用户授权重定向接口
定义微信登录关联本地帐户接口
定义微信登录创建并关联本地帐户接口
整理项目
修复：错别字、接口地址、处理器响应

产品模块

准备项目（product）
创建产品数据表
添加产品数据（许可与订阅产品）
定义产品模型（ProductModel）
定义按类型调取产品的服务方法
定义获取许可产品的接口
定义获取订阅产品的接口
整理项目

支付方法

准备项目（payment-options）
创建支付方法数据表（payment）
添加支付方法数据
定义支付方法数据模型（PaymentModel）
定义获取支付方法用的服务方法（getPayments）
定义获取支付方法接口
整理项目

订单模块

准备项目（order）
创建订单数据表（order）
定义订单数据模型（OrderModel）
创建订单模块
定义按产品 ID 调取单个产品的方法
创建订单守卫中间件（orderGuard）
定义创建订单方法（createOrder）
定义创建订单接口
定义按 ID 调取订单的方法（getOrderById）
定义更新订单方法（updateOrder）
定义更新订单守卫中间件（updateOrderGuard）

定义更新订单接口
创建订单日志数据表 (order_log)
定义订单日志数据模型 (OrderLogModel)
定义创建订单日志用的方法 (createOrderLog)
创建与更新订单时记录订单日志
整理项目

下载许可

准备项目 (license)
创建许可数据表 (license)
定义许可数据模型 (LicenseModel)
定义创建许可方法 (createLicense)
创建订单以后判断产品类型创建许可
定义更新许可方法 (updateLicense)
定义按订单 ID 调取许可方法
定义获取用户有效许可方法
定义获取用户有效许可接口
准备许可列表配置
定义许可相关查询片断 (licenseSqlFragment)
定义调取许可列表方法 (getLicenses)
定义统计许可数量方法 (getLicensesTotalCount)
定义许可列表接口
整理项目

订阅会员

准备项目 (subscription)
创建订阅数据表 (subscription)
定义订阅数据类型 (SubscriptionModel)
定义创建订阅方法 (createSubscription)
定义更新订阅方法 (updateSubscription)
定义调取用户有效订阅的方法
创建订单以后判断产品类型创建订阅
创建订阅日志数据表 (subscription_log)
定义订阅日志数据模型 (SubscriptionLogModel)
定义创建订阅日志方法 (createSubscriptionLog)
定义按订单 ID 调取订阅日志方法
增加订阅日志动作
创建订阅以后创建订阅日志
创建订单时处理订阅的续订与重订
创建订单时处理升级订阅
定义按 ID 调取订阅的方法 (getSubscriptionById)

定义订阅完成后处理方法
测试订阅的后处理动作
定义获取用户有效订阅接口
定义调取订阅历史的方法
定义订阅历史接口
整理项目

下载模块

准备项目 (download)
创建下载数据表 (download)
定义下载相关服务方法
定义统计下载次数的方法 (countDownloads)
标准订阅下载限制配置与错误信息处理
定义下载守卫中间件 (downloadGuard)
定义生成唯一 ID 的方法
定义创建下载接口
测试创建下载接口
定义文件下载守卫中间件 (fileDownloadGuard)
定义文件下载接口
测试文件下载接口
在有效订阅接口里添加周下载情况
整理项目

支付地址

准备项目 (payment-url)
创建支付地址数据表
定义支付地址相关服务方法
添加日期时间格式配置 (DATE_TIME_FORMAT)
定义支付地址守卫中间件 (paymentUrlGuard)
定义支付地址接口
测试支付地址接口
整理项目

准备支付

准备项目 (payment-prepare)
创建日志记录器 (logger)
XML 构建器与解析器 (xmlBuilder, xmlParser)
添加应用名与应用别名配置
定义订单支付守卫中间件 (payOrderGuard)
定义订单支付接口
定义已收到付款的处理方法 (paymentRecived)

测试已收到付款的处理方法 (paymentRecived)
定义支付结果通知接口 (微信支付、支付宝)
配置 SSH 通道访问本地应用开发服务
整理项目

微信支付

准备项目 (payment-wxpay)
准备微信支付相关配置
定义微信支付相关类型
定义微信支付签名方法 (wxpaySign)
定义微信支付验证支付结果通知方法
定义微信支付方法 (wxpay)
请求使用微信支付
处理微信支付结果通知 (wxpayNotify)
测试使用微信支付
整理项目

支付宝

准备项目 (payment-alipay)
准备支付宝相关配置
定义支付宝相关类型
定义获取支付宝请求参数方法
定义签名预处理与签名方法
定义制作支付宝支付地址的方法
定义支付宝支付方法 (alipay)
定义支付宝验证支付结果的方法
处理支付宝支付结果通知
使用支付宝支付
整理项目

修复改进

准备项目
改进: 创建订单以后响应订单数据
改进: 调取用户数据时包含用户订阅相关数据
改进: 使用了下载以后触发实时事件
改进: 许可与订阅发生变化时实时通知客户端
改进: 在内容列表项目里包含用户订阅数据
整理项目

订单列表

准备项目 (order-list)
准备每页订单数量配置
订单列表查询片段
定义调取订单列表方法 (getOrders)
订单列表过滤器中间件 (orderIndexFilter)
定义订单列表接口
定义统计订单数量与金额的方法 (countOrders)
产品 Meta 与按产品类型过滤订单列表
定义获取订单许可项目方法
改进: 按订单 ID 调取许可
定义订单许可项目守卫中间件
定义订单许可项目接口
定义调取订单订阅项目方法
改进: 让权限控制中间件支付检查是否为管理员
定义订单订阅项目接口

Vue.js 前端应用开发训练路线

Vue.js 前端应用开发实践

理解框架

准备 Vue 项目的开发环境与工具
Vue CLI: 项目的命令行工具
用 Vue CLI 创建一个 Vue 项目
运行 Vue 项目
了解 Vue 项目的结构
了解 Vue 应用的结构
了解 Vue 应用的组件 (Components)
了解 Vue 应用的路由 (Vue Router)
了解 Vue 应用的 Store (Vuex)
VSCode: 配置开发 Vue 应用
了解如何部署 Vue 应用

组件基础

准备一个空白的 Vue 项目
创建 Vue 应用实例
data: Vue 组件里的数据
created: 组件的生命周期方法
methods: 组件里的方法
computed: 组件里的计算属性
watch: Vue 组件里的数据监视器
v-on: 处理 Vue 组件里发生的事情 (事件处理)
v-for: 在 Vue 组件里显示内容列表
v-if: 按条件显示内容
绑定 Class
在 Vue 组件里导入样式表文件

表单元素

准备项目 (forms)
文本框 (Text)
多行文本 (Textarea)
复选框 (Checkbox)

多值复选框 (Checkbox)

单选按钮 (Radio)

选择列表 (Select)

input 事件: 元素值变化时发生的事件

change 事件: 用户提交修改元素值时发生的事件

keyup 事件: 按下并抬起键盘按键时发生的事件

理解在表单元素上用的 v-model 指令

v-model 的修饰符: .lazy

v-model 的修饰符: .number

v-model 的修饰符: .trim

高级组件

准备项目 (components)

定义与使用 Vue 组件

Vue 组件的样式

scoped: 限定组件样式的应用范围

props: Vue 组件的属性

监听子组件的事件

slot: 在 Vue 组件里用插槽分发内容

动态组件 (Dynamic Components)

keep-alive: 保持动态组件的活动状态

\$refs: 引用模板

在自定义组件上使用 v-model

Vue 组件的生命周期方法

应用路由

准备项目 (router)

VSCode: 选择代码格式化工具

创建 Vue 应用的路由器

定义 Vue 应用的路由

router-link: 路由链接

用代码切换路由地址

给路由起个名字

重定向 (Redirect)

路由模块

动态路由 (带地址参数的路由)

嵌套路由 (Nested Routes)

给路由组件传递属性

导航守卫 (Navigation Guards)

路由元数据字段 (meta)

状态管理

准备项目 (vuex)
在 Vue 应用里创建与使用 Store
State: 数据
mapState
Getters: 获取器
Mutations: 修改器
mapMutations
Actions: 动作
mapActions
动作的 Context 参数
实践: 加载状态
modules: Vuex Store 模块
namespaced: Vuex Store 模块的命名空间
创建与使用 Vuex Store 插件

过渡动画

准备项目 (transition_animation)
准备一个演示组件
基于 Class 的过渡
基于 Class 的动画
transition: 进入与离开过渡
transition: 指定过渡效果的名字
transition: 进入与离开动画
transition: 自定义过渡类
组件之间的过渡
列表的过渡与动画
状态的过渡

反应系统

准备项目 (reactivity_composition)
JavaScript: Proxy 与 Reflect
理解 Vue 的反应系统
Composition API: 组合式接口
声明具有反应特性的数据 (ref)
声明具有反应特性的数据 (reactive)
解构反应数据 (toRefs)
组合式接口: 提供组件方法
组合式接口: 生命周期方法
组合式接口: 监视数据 (watch)
组合式接口: 计算属性 (computed)

setup: 组合式接口的入口

网络请求

- 准备项目 (http)
- 准备服务端应用接口
- 解决服务端跨域资源共享问题 (CORS)
- 安装与使用 HTTP 客户端请求服务端接口 (axios)
- 使用 axios 发送请求时发生的错误
- 使用 async await 处理 axios 请求
- axios 请求的配置
- 修改 axios 默认的请求配置
- axios 的请求与响应的拦截器 (Interceptors)
- 创建与使用 axios 实例
- 发送 POST 类型的 HTTP 请求
- 在请求里配置头部数据
- 发送 PATCH PUT 类型的 HTTP 请求
- 发送 DELETE 类型的 HTTP 请求

身份验证

- 准备项目 (auth)
- 理解用户登录与验证用户身份
- 创建用户登录组件
- 请求用户登录接口
- 处理用户登录
- 获取当前用户数据
- 在 Vue 应用里记住用户的登录状态
- 退出登录

上传文件

- 准备项目 (upload-files)
- 选择文件用的文件字段
- 使用 FileReader 生成预览图像
- 请求上传文件
- 显示文件上传进度
- 使用拖拽的方式选择要上传的文件
- 切换拖放文件时的激活状态

应用架构

- 准备项目 (structure)
- 创建与使用 Store

定义使用 Store 模块
解决 Store 模块的类型问题
应用的路由：Vue Router
HTTP 客户端（axios）：请求服务端应用接口
Vue 应用的环境变量配置
在 Vue 应用里定义与使用组件
在 Vue 组件内部请求服务接口获取组件需要的数据
在 Vue 组件里使用 Store 里的动作与数据
内容页面：定义 Store 模块
内容页面：定义路由与组件
Mixin：制作一个设置页面标题的混合

Vue.js 前端应用开发实例

准备项目

准备前端项目（Vue）
准备服务端项目（Node）
创建前端项目仓库
创建服务端项目仓库
启动本地开发服务
了解开发流程
VSCode：Vue 项目的编辑器配置
准备项目资源

页面布局

准备项目（layout）
用 normalize 样式表重置默认样式
提交（Commit）对项目做的修改
定义应用标志组件（AppLogo）
定义应用图标组件（AppIcon）
定义页面布局组件（AppLayout）
页面布局基本样式
定义应用边栏组件（AppSidebar）
应用边栏组件样式
定义应用工具栏组件（AppToolbar）
应用工具栏组件样式
整理项目

应用主题

准备项目 (theme)

Bug Fix: Safari 浏览器下无法正常显示应用图标

添加基本样式表 (base)

添加主题样式表 (theme)

定义布局 Store 模块 (layoutStoreModule)

定义主题切换器组件 (AppSidebarItemTheme)

根据主题切换标志与图标颜色

定义本地存储服务方法

定义本地存储 Store 插件

创建应用布局组件时设置主题

整理项目

身份验证

准备项目 (auth)

定义登录路由

定义登录组件 (AuthLogin)

设计表单元素样式 (form.css, button.css)

定义文本框表单字段组件 (TextField)

定义按钮表单字段组件 (ButtonField)

定义用户登录 Store 模块

使用用户登录 Store 模块

在本地存储里存储用户令牌

定义表示当前用户是否登录的获取器 (isLoggedIn)

设置 API HTTP 客户端 Authorization 头部

整理项目

通知提醒

准备项目 (notification)

定义通知提示组件 (AppNotification)

定义通知提示信息组件

通知提示信息组件样式

定义通知提示 Store 模块

推送通知提示信息 (pushMessage)

收回通知提示信息 (dismissMessage)

通知提示信息动画与过渡

用户登录时显示通知提示信息

整理项目

用户头像

准备项目 (user-avatar)

了解服务端应用的头像接口
创建用户头像组件 (UserAvatar)
定义用户头像组件
用户头像组件的样式
用户头像的链接
定义用户档案 Store 模块
定义获取当前用户用的动作 (getCurrentUser)
应用创建时与登录成功后获取当前用户
未登录时淡化用户默认头像
登录成功以后返回到上一级的页面
整理项目

用户页面

准备项目 (user-show)
定义用户路由 (userRoutes)
让用户头像链接到用户页面
定义用户页面组件 (UserShow)
用户页面组件样式
定义用户名组件 (UserName)
定义用户资源组件与路由
定义用户页面菜单组件 (UserShowMenu)
用户页面菜单组件样式
整理项目

用户帐户

准备项目 (user-account)
修正：切换用户页面时不加载用户数据
创建用户帐户组件 (UserAccount)
用户帐户菜单项目
创建设置用户头像的组件
定义文件表单字段组件 (FileField)
文件表单字段组件样式
选择头像文件后预览用户头像
定义用户帐户 Store 模块
定义与使用上传用户头像动作 (createAvatar)
在用户头像组件里显示选择的头像
设置用户头像组件的选择文件与取消按钮
定义修改用户名组件 (UserAccountChangeName)
定义与使用更新用户帐户动作
修正：更新用户帐户后显示多个帐户菜单项目
定义与使用修改用户密码组件

整理项目

用户菜单

准备项目 (user-menu)

改进: 添加 body 与 notification 背景

滚动页面时设置页面头部阴影

处理边栏用户菜单项目的链接与点击

定义用户菜单组件 (UserMenu)

过渡用户菜单组件的显示与隐藏

定义关闭按钮组件 (CloseButton)

点击关闭按钮或按 ESC 键关闭用户菜单

用户菜单头部

用户菜单项目

用户菜单里的退出登录按钮

定义退出登录动作 (logout)

整理项目

内容列表

准备项目 (post-list)

定义内容列表项目类型 (interface PostListItem)

在 posts 获取器里准备好图像文件地址

设置应用首页

定义内容列表项目媒体 (PostListItemMedia)

内容列表样式

根据内容图像方向设计内容列表项目样式

定义内容列表项目内容组件

设计内容列表项目内容组件样式

整理项目

列表布局

准备项目 (post-list-layout)

定义内容列表布局图标组件 (PostListLayoutIcon)

定义内容列表布局切换器组件

准备与使用内容列表布局数据

存储当前选择的内容列表布局

内容列表布局小图标的激活状态

设计内容列表布局样式

定义工具栏 Store 模块 (appToolbarStoreModule)

定义工具栏项目路由守卫 (appToolbarItemGuard)

整理项目

滚动加载

- 准备项目 (infinite-scrolling)
- 定义与使用内容列表项目骨架组件
- 设置每页内容项目数量
- 定义获取内容列表后期处理动作
- 滚动页面到底部时加载内容列表
- 热门内容列表页面
- 定义处理地址查询符用的方法
- 定义获取内容列表前期处理动作
- 设置内容列表页面的标题
- 整理项目

内容页面

- 准备项目 (post-show)
- 定义内容数据的类型 (Post)
- 定义处理内容图像文件服务方法
- 定义内容页面媒体组件 (PostShowMedia)
- 内容页面布局变化
- 定义内容页面头部组件 (PostShowHeader)
- 定义内容页面内容组件 (PostShowContent)
- 定义内容页面动作组件 (PostShowActions)
- 整理项目

照片参数

- 准备项目 (file-meta)
- 定义文件 Store 模块
- 定义文件原数据类型 (FileMetadata)
- 定义拍摄设备获取器 (kit)
- 定义拍摄参数获取器 (specs)
- 定义图像文件元数据组件 (PostShowFileMeta)
- 定义拍摄设备组件 (PostShowFileMetaKit)
- 设计拍摄设备组件样式
- 定义拍摄参数组件 (PostShowFileMetaSpecs)
- 设计拍摄参数组件样式
- 整理项目

内容过滤

- 准备项目 (post-list-filter)
- 定义内容标签组件 (PostTag)
- 定义内容标签列表组件 (PostShowTags)

用路由给内容列表传递 sort 与 filter 属性
定义处理内容列表过滤器服务方法 (filterProcess)
内容列表过滤
准备过滤器数据 (filterItems)
定义过滤器组件 (PostListFilters)
用户发布与喜欢的作品页面
内容列表布局切换器过渡效果
整理项目

内容导航

准备项目 (post-show-navigator)
定义与使用内容导航组件 (PostShowNavigator)
控制内容导航的显示
内容导航的过渡效果
准备与使用内容导航数据与动作
定义与使用是否允许前进与后退的获取器
在内容导航中加载更多列表数据
使用键盘左右方向键控制内容导航
定义与使用内容页面骨架 (PostShowSkeleton)
内容页面图像加载样式
整理项目

页面侧板

准备项目 (page-side-sheet)
创建页面侧板组件 (PageSideSheet)
显示页面侧板
关闭页面侧板
页面侧板过渡
显示或隐藏页面侧板时改变内容页面布局
在内容页面上按 B 返回上一页
改进: 内容列表项目骨架媒体元素比例
改进: 用户头像图像加载
改进: 内容列表项目媒体组件图像的显示比例
修正: 内容导航路由地址错误
修正: 快速内容导航时内容页面闪动
整理项目

评论列表

准备项目 (comment-list)
定义评论索引组件 (CommentIndex)
定义评论列表组件 (CommentList)

定义评论 Store 模块 (commentStoreModule)
定义评论索引 Store 模块
准备获取评论列表需要的过滤器数据
点击内容列表项目评论时打开页面侧板
设置每页评论数量 (COMMENTS_PER_PAGE)
完善评论索引 Store 模块
请求获取内容的评论列表数据
定义评论列表项目组件 (CommentListItem)
定义评论列表项目元数据组件
定义评论列表项目内容组件
定义评论列表项目动作组件
用回复数量按钮切换显示回复列表
整理项目

回复列表

准备项目 (reply-list)
定义回复 Store 模块 (replyStoreModule)
定义回复索引 Store 模块
创建回复索引组件 (ReplyIndex)
创建回复列表组件 (ReplyList)
同时显示多条评论的回复列表
创建回复列表项目组件 (ReplyListItem)
创建回复项目元数据组件 (ReplyListItemMeta)
创建回复项目内容组件 (ReplyListItemContent)
整理项目

制造工具

准备项目 (cli)
创建一个 Vue CLI 插件项目
在项目里安装本地 Vue CLI 插件
理解 Vue CLI 插件里的 Generator
用 Generator 扩展项目自定义命令
用 Generator 在项目里添加新组件 (render)
定义获取生成文件完整路径的方法
获取生成的组件样式表的存放位置
定义获取组件与样式模板文件的方法
定义获取生成的组件名的方法
在模板文件里使用传递的值
修正: PascalCase 组件名多余空格
定义获取父辈文件路径方法 (getParentFilePath)
定义获取父辈名字方法 (getParentName)

- 定义获取导入路径方法
- 定义获取组件导入声明方法
- 在父组件中导入新生成的组件（injectImports）
- 利用 Generator 钩子做后期处理
- 定义获取项目文件内容方法
- 定义插入内容用的方法（insertFileContent）
- 在父组件那里注册与使用生成的组件

生成模块

- 创建 Store Generator 与 Store Generator Hook
- 准备 Store 模块模板文件
- 定义获取 Store 数据类型方法
- 定义获取 Store 数据模块名方法
- 用命令在项目里生成 Store 模块
- 在父 Store 模块里导入新创建的 Store 模块
- 在父 Store 模块里注册子模块
- 定义转换为 PascalCase 命名方式的方法
- 定义获取 Store 选项的方法（getStoreOptions）
- 根据命令选项使用不同的 Store 模板
- 定义与使用带动作的 Store 模块模板
- 准备创建带资源的 Store 选项
- 定义与使用带资源的 Store 模块模板
- 添加 Store 模块动作里的参数
- 添加 Store 模块动作的预处理与后处理
- 生成组件时支持使用 Vuex 的帮手方法
- 把 Vue CLI 插件发布到 NPM
- 在 Vue 项目里安装使用 Vue CLI 插件
- 整理项目

创建评论

- 使用项目的命令行工具
- 准备项目（comment-create）
- 新建创建评论组件与 Store 模块
- 定义文本区域字段组件（TextareaField）
- 定义创建评论组件（CommentCreate）
- 设计创建评论组件样式
- 定义创建评论动作（createComment）
- 发表评论
- 用按键组合发表评论
- 处理创建评论组件里的注册与注册按钮
- 整理项目

管理评论

- 准备项目 (comments)
- 页面侧板滚动触地
- 页面侧板滚动触地时加载评论列表数据
- 定义评论列表项目骨架组件
- 点按评论内容显示删除按钮
- 定义删除评论 Store 模块
- 删除评论
- 准备更新评论按钮
- 设计 Pill 按钮的样式
- 定义评论编辑组件 (CommentEdit)
- 定义更新评论 Store 模块
- 更新评论
- 防止提交空白评论并整理项目

管理回复

- 准备项目 (reply)
- 添加评论回复按钮
- 定义回复组件 (ReplyCreate)
- 定义创建回复 Store 模块
- 创建回复
- 回复成功以后增加评论的回复数量
- 回复成功以后加载回复列表
- 定义回复列表项目动作组件
- 删除回复
- 删除回复后减少评论回复数量
- 编辑回复
- 整理项目

用户评论

- 准备项目 (user-comments-replies)
- 用户评论列表
- 在用户评论列表项目里包含评论内容
- 用户评论列表分页
- 用户回复列表
- 在用户回复列表里包含回复的评论
- 整理项目

创建内容

- 准备项目 (post-create)

定义创建内容组件 (PostCreate)
添加标题与描述字段
在创建内容组件里添加提交按钮
定义创建内容 Store 模块
在内容列表中过滤掉没有相关文件的项目
内容发布以后
定义与使用更新内容 Store 模块
去掉文本框字段与文本区域字段值的前后空格
整理项目

内容标签

准备项目 (post-tags)
定义内容标签字段组件 (PostTagField)
定义与使用添加内容标签动作 (createPostTag)
在创建与编辑内容时显示内容标签列表
在内容标签组件里添加删除按钮
定义与使用删除内容标签动作 (deletePostTag)
添加与删除内容标签时的过渡
整理项目

修复改进

准备项目 (post-create-refactor)
修正: 获取内容时使用的错误的参数类型
把创建内容组件里的数据放在 Store 模块里
定义内容标题字段组件 (PostTitleField)
定义内容正文字段组件 (PostContentField)
定义内容动作组件 (PostActions)
在提交按钮上显示未保存状态
改进创建内容页面上的标签字段
改进: 用 input 事件触发 dirty 事件
准备删除内容按钮
删除按钮自动恢复状态
定义删除内容 Store 模块
定义内容元数据组件 (PostMeta)
创建内容编辑按钮组件 (PostShowEdit)
整理项目

上传文件

准备项目 (file-create)
定义文件上传组件 (FileCreate)
定义文件拖放组件 (FileCreateDragZone)

拖放选择文件
定义创建文件 Store 模块
定义创建内容媒体组件 (FileCreateMedia)
生成与使用预览图像
准备上传文件动作 (createFile)
定义文件上传进度组件 (FileCreateStatus)
上传与创建文件
整理项目
修复: FileCreateMedia 组件少写了事件处理器
改进: 更新内容时的闪动

内容点赞

准备项目 (likes)
定义点赞与取消点赞 Store 模块
定义点赞内容动作组件 (PostLikeAction)
点赞与取消点赞时更新内容项目的点赞状态与数量
单个内容的点赞与取消点赞
整理项目

管理列表

修复分支问题并准备项目 (manage)
定义管理内容组件与路由
定义管理内容列表组件 (ManagePostList)
定义管理内容列表项目组件
定义管理内容列表项目媒体组件
定义管理内容列表项目内容组件
定义管理内容列表项目骨架组件
内容管理页面滚动加载
内容高度不够时自动加载列表数据
整理项目

修复改进

准备项目 (side-sheet-refactor)
改进页面侧板
修改内容页面触发页面侧板
创建内容评论动作组件 (PostCommentAction)
修改内容列表项目触发页面侧板
页面侧板工具栏项目组件
定义页面侧板工具栏项目组件
整理项目

管理内容

准备项目 (manage-post)
定义管理内容侧板组件 (ManagePostSideSheet)
定义管理 Store 模块 (manageStoreModule)
定义管理选择 Store 模块
定义获取选择内容动作 (getSelectedPosts)
定义管理选择项目动作 (manageSelectedItems)
选择管理内容项目
按 esc 按键取消选择所有项目
整理项目

编辑内容

准备项目 (manage-post-edit)
修复：侧板工具栏项目错误的方法名称
定义管理内容编辑组件 (ManagePostEdit)
创建管理内容编辑选择相关组件
定义页面侧板状态组件 (PageSideSheetStatus)
定义管理内容选择状态组件
定义管理内容选择列表项目组件
在管理内容选择列表项目里移除选择的项目
建管理内容编辑表单组件 (ManagePostEditForm)
在管理内容编辑表单里添加内容标题与描述字段
定义提交按钮组件 (SubmitButton)
在管理内容编辑表单提交更新内容
内容更新以后更新内容列表项目
定义删除按钮组件 (DeleteButton)
在管理内容编辑表单删除选中的内容项目
改进：管理内容页面自动加载数据
在管理内容页面用快捷键删除选择的内容项目
在管理内容页面编辑表单上添加打标签字段
获取选中内容的标签列表
在管理编辑表单上给选中的内容打标签
在管理编辑表单上移除选中内容里的标签
整理项目

实时应用

准备项目 (realtime-app-client)
请求服务端接口时在请求里包含 Socket Id
在内容列表页面监听服务端点赞事件
在内容页面监听服务端点赞事件

监听服务端取消点赞事件 (userLikePostDeleted)
监听评论已创建事件添加新的评论项目
监听评论已更新事件更新评论内容
监听评论已删除事件删除评论项目
实时更新内容与内容项目的评论数量
实时更新评论回复数量
实时更新评论的回复列表
实时更新回复内容
整理项目

修复改进

准备项目 (fix-and-improve)
将网页语言从英文改为简体中文
在内容页面撰写评论或回复时停止传播按键事件
:after 全部替换成 ::after
防止没有选择照片提交创建内容
在客户端限制上传文件的类型
当照片没有元数据时在内容页面隐藏元数据
在客户端限制要上传的图像文件的尺寸
退出登录以后还可以用之前的用户身份操作
退出登录以后防止点赞行为
应用通知被页面侧板覆盖
改进创建内容页面提示用户登录
用路由守卫检查用户登录状态
整理项目

内容搜索

准备项目 (search)
过滤出用特定相机与镜头拍摄的内容列表
创建应用搜索组件与处理搜索工具栏项目组件
定义应用搜索 Store 模块
定义搜索选项组件
处理应用搜索组件的样式
定义搜索字段组件 (AppSearchField)
搜索字段焦点状态
处理搜索结果数据
定义搜索结果组件
搜索结果组件样式与搜索测试
关闭与清理搜索结果
整理项目
修复: 搜索字段的位置 (AppSearchField)

发布状态

准备项目 (publish-audit)
配置普通管理与管理员模式
在内容管理页面使用管理模式获取内容列表
理解图片权限控制
定义与使用获取图片数据的方法
创建管理内容状态动作组件
定义管理内容状态动作组件
创建复选框字段组件 (CheckBoxField)
复选框字段组件的模板与样式 (CheckBoxField)
定义复选框字段组件 (CheckBoxField)
在管理内容编辑表单更新与批量内容发布状态
创建与编辑内容时的内容发布状态
定义内容状态字段组件 (PostStatusField)
整理项目

管理模式

准备项目 (admin)
修复: 移动 AppToolBarItemSideSheet 组件的位置
创建管理工具栏项目组件 (AppToolBarItemAdmin)
定义表示是否使用管理员模式的相关数据
控制管理工具栏项目组件的显示
切换管理员模式
修复: 从管理页面切换到首页时出现 403 警告
整理项目

审核内容

准备项目 (audit-log)
修复: backdrop-filter 引起的页面性能问题
定义圆形按钮组件 (CircleButton)
改进: 使用圆形按钮组件
定义审核日志相关 Store 模块
准备编辑内容时需要的 audit 数据
创建内容审核动作组件 (PostAuditAction)
内容审核动作组件的模板与样式
定义内容审核动作组件 (PostAuditAction)
内容审核中或审核通过后隐藏上传文件用的组件
在管理内容页面的内容项目上使用内容审核动作
创建管理内容审核动作组件
定义管理内容审核动作组件

修复：在内容页面打开首页时不显示内容列表
整理项目

统计仪表

准备项目（dashboard）
创建仪表盘与仪表盘访问统计组件
定义仪表盘路由
定义应用边栏仪表盘项目组件
创建仪表盘与仪表盘访问统计 Store 模块
定义获取访问次数列表动作（getAccessCounts）
定义按动作分时段获取访问次数动作
创建访问次数面板组件
设计访问次数面板组件样式
实时更新访问次数
创建访问次数的日期时间范围过滤器组件
定义访问次数的日期时间范围过滤器组件
按时间范围过滤访问次数
设置当前活动的访问次数面板
访问次数活动面板变更时更新地址查询符
创建访问次数图表面板组件
制作访问次数统计图表（Chart.js）
调整统计图表样式
实时更新统计图表的显示
整理项目

创建用户

准备项目（user-create）
定义创建用户组件与路由（UserCreate）
切换登录与注册页面用的链接
动态设置登录与注册按钮的状态
定义创建用户 Store 模块
注册用户
注册用户成功以后的页面提示
整理项目

登录界面

准备项目（weixin-login-ui）
创建微信登录组件、Store 模块与路由
定义微信登录按钮组件（WeixinLoginButton）
定义微信图片组件（WeixinIcon）
准备微信登录步骤数据

定义微信登录页头组件 (WeixinLoginHeader)
创建微信登录嵌入组件 (WeixinLoginIframe)
用动态组件展示微信登录步骤组件
创建带图标文字的按钮组件 (IconTextButton)
准备微信登录关联帐户选项数据
定义微信登录关联帐户选择组件
定义微信登录绑定帐户组件
创建微信登录组件时重置登录步骤
整理项目

微信登录

准备项目 (weixin-login)
准备微信登录相关配置
在应用里记录客户端的 socketId
在 Iframe 元素里展示微信登录二维码页面
微信登录时提示用户绑定本地应用帐户
记录微信帐户信息
定义微信登录动作 (weixinLogin)
定义微信登录关联帐户动作
微信登录绑定本地应用帐户
微信登录成功
整理项目

内容侧板

准备项目 (post-side-sheet)
创建内容页面侧板 (PostSideSheet)
控制页面侧板关闭按钮的显示
在内容页面添加下载动作组件
创建内容页面侧板 Store 模块
定义内容页面侧板内容组件
定义内容页面侧板元数据组件
定义内容页面侧板头部组件
创建内容页面侧板动作组件
创建内容页面侧板页脚组件
创建页面侧板头部下载组件
定义二维码图像组件 (AppQrcode)
整理项目

照片下载

准备项目 (download)
下载与创建下载 Store 模块

定义创建下载动作 (createDownload)
内容页面侧板头部下载组件
点击下载图标下载原版照片
处理服务端的文件下载已使用事件
生成二维码扫码下载
定义有效订阅获取器 (hasValidSubscription)
创建许可 Store 模块定义获取有效许可动作
定义获取下载权限动作 (getDownloadPermission)
测试用户对当前资源的下载权限
整理项目

支付方法

准备项目 (payment)
定义支付列表 Store 模块
定义支付方式选择 Store 模块
定义支付方式小图标组件 (PaymentIcon)
定义初始化动作检查下载权限加载支付方式列表
定义内容页面侧板头部支付组件
改进: 复选框组件 (CheckboxField)
定义支付方式选择组件 (PaymentSelect)
整理项目

准备产品

准备项目 (product)
定义产品 Store 模块获取产品
初始化内容页面侧板时获取产品数据
定义产品选择 Store 模块
定义内容页面侧板标签组件 (PostSideSheetTab)
定义内容页面侧板标签组件 (PostSideSheetTab)
创建内容页面侧板许可与订阅内容组件
定义内容页面侧板许可内容组件
设置内容页面侧板支付头部标题
创建产品价格组件 (ProductPrice)
定义产品价格组件
设计产品价格组件样式
定义订阅图标组件
创建订阅选择与订阅选择选项组件
定义订阅选择选项组件
设计订阅选择选项样式
整理项目

准备订单

准备项目 (order)
订单与创建订单 Store 模块
处理创建订单动作 (createOrder)
在创建订单 Store 模块里定义一些 Getters
定义创建许可与订阅订单动作
定义创建订单分解器动作 (createOrderResolver)
定义订单支付 Store 模块 (orderPayStoreModule)
支付订单
定义更新订单 Store 模块
定义更新订单分解器动作 (UpdateOrderResolver)
更新订单后修改已创建的订单
切换产品时根据订单设置支付方法
站外支付链接
监听服务端许可与订阅变更事件
测试购买许可与订阅
整理项目

会员订阅

准备项目 (subscription)
定义订阅颜色配置
创建订阅组件与订阅路由
修改用户名组件添加订阅图标
创建处理订阅用的 Store 模块
创建订阅注册与注册头部组件
定义步骤组件 (AppSteps, AppStep)
定义订阅注册内容组件
创建订阅注册卡片组件 (SubscriptionSignupCard)
创建订阅卡片组件 (SubscriptionSignupCard)
定义订阅注册卡片组件 (SubscriptionSignupCard)
设计订阅卡片组件样式
选择与重新选择订阅类型
设计 Stack 订阅卡片样式
整理项目

订阅支付

准备项目 (subscription-payment)
创建订阅支付组件 (SubscriptionPayment)
定义订阅支付组件 (SubscriptionPayment)
定义当前订单获取器 (currentOrder)
创建与更新订阅订单

订阅卡片带边框的 Meta 样式
完成订阅
创建订阅信息 Store 模块定义调取有效订阅动作
创建订阅信息组件并切换显示订阅组件
订阅注册完成以后点击查看订阅信息
定义订阅信息卡片组件 (SubscriptionInfoCard)
定义日期时间组件 (AppDatetime)
订阅信息卡片组件动作
续订、重订、升级支付
处理订阅变更事件
整理项目

订阅历史

准备项目 (subscription-history)
创建订阅历史组件 (SubscriptionHistory)
准备订阅历史数据与调取订阅历史动作
定义订阅历史项目组件 (SubscriptionHistoryItem)
设计订阅历史项目组件样式
改进：订阅支付里的应付金额
修复：重置订阅步骤
修复：订阅完成以后清除订阅订单数据
修复：订阅完成以后调取当前用户
整理项目

许可列表

准备项目 (license-list)
准备每页许可数量配置
创建许可列表 Store 模块
创建许可列表组件与路由
创建许可列表与列表项目组件
定义支付方法名称组件 (PaymentName)
定义内容图像组件 (PostImage)
定义许可列表项目组件 (LicenseListItem)
设计许可列表项目组件样式
点击许可项目内容图像下载原版照片
切换显示内容图片与下载二维码
在许可列表扫码下载
准备许可列表分页 Store
准备许可列表分页组件
整理项目

订单列表

准备项目 (order-list)
创建订单列表页面组件与路由
创建订单列表页面侧板组件
在订单列表页面切换使用管理模式
定义每页订单数量配置
创建订单列表 Store 模块
定义订单列表 Store 模块
定义订单列表组件 (OrderList)
定义订单状态组件 (OrderStatus)
定义订单列表项目与头部组件
设计订单列表项目与头部组件样式
修复：每页订单数量环境变量
定义订单列表统计组件 (OrderCounter)
整理项目

订单过滤

准备项目 (order-list-filter)
定义获取日期时间范围方法 (getDateTimeRange)
定义日期时间范围过滤器
定义订单日期时间过滤器 (OrderDateTimeFilter)
定义订单状态过滤器 (OrderStatusFilter)
定义订单支付过滤器 (OrderPaymentFilter)
定义获取订单项目数据动作
定义订单项目销售组件 (OrderItemSales)
定义订单许可项目组件 (OrderLicenseItem)
定义订单订阅项目组件 (OrderSubscriptionItem)
定义订单列表项目 Extra 组件
整理项目

修复改进

准备项目 (fix-and-improve-2)
修复：内容页面过渡在 Safari 浏览器卡顿
改进：添加页面头部透明效果

Flutter 客户端应用开发训练路线

Flutter 客户端应用开发实践

开发环境

介绍（Flutter 准备开发）

安装 Xcode（macOS）

安装 Homebrew（macOS）

安装配置 Flutter（macOS）

安装配置 Flutter（Windows）

配置国内 Flutter 镜像

配置开发 iOS 与 macOS 平台应用的环境（用 Flutter Doctor 检查问题）

配置开发 Android 平台应用的环境

管理配置运行 Android 模拟器

配置 VSCode 编辑器开发 Flutter 项目

创建并运行 Flutter 项目

创建项目远程仓库

数据类型

介绍与准备（Dart 语言基础：数据类型）

创建与运行 Dart 项目

Dart 语言：声明变量（var）

Dart 语言：声明变量（final）

Dart 语言：声明变量（const）

对比 Dart 语言里的 var，final，const

Dart 语言：类型

Dart 语言的内置类型：字符串（String）

Dart 语言的内置类型：数字（int, double）

Dart 语言的内置类型：布尔值（bool）

Dart 语言的内置类型：列表（List）

Dart 语言的内置类型：集合（Set）

Dart 语言的内置类型：映射（Map）

函数

Dart 语言：函数

Dart 语言：函数参数

Dart 语言: 多个参数
Dart 语言: 有名字的参数
Dart 语言: 参数的默认值
Dart 语言: 函数返回值

控制流程

Dart 语言: 条件判断 (if)
Dart 语言: 检查不同的情况 (switch)
Dart 语言: 抛出异常 (throw)
Dart 语言: 自定义异常
Dart 语言: 处理异常 (try...catch)
Dart 语言: For 循环

类

Dart 语言: 类与实例
Dart 语言: 属性
Dart 语言: 构造方法
Dart 语言: this
Dart 语言: 带名字的构造方法
Dart 语言: 设置属性值的简单写法
Dart 语言: 方法
Dart 语言: 继承
Dart 语言: super
Dart 语言: 类属性与类方法
Dart 语言: 泛型
Dart 语言: 导入使用库

异步编程

Dart 异步编程: 定义异步函数
Dart 异步编程: 使用异步函数提供的值
Dart 异步编程: 处理异步函数遇到的错误
Dart 异步编程: Stream
Dart 异步编程: 使用流水线上的数据
Dart 异步编程: 处理流水线上的错误
Dart 异步编程: 订阅流水线
Dart 异步编程: 流水线控制器
Dart 异步编程: 流水线下水槽
Dart 异步编程: 多次订阅流水线 (Broadcast)

包管理

Dart 管理包裹
Dart 管理包裹：创建 pubspec.yaml
Dart 管理包裹：安装包
Dart 管理包裹：使用包
Dart 管理包裹：使用自己项目里的库
Dart 管理包裹：升级包

基础部件

准备项目 (widget)
准备应用入口
使用小部件
使用 Dart 开发者工具
自定义小部件 (StatelessWidget)
使用 Text 小部件显示文本
使用 RichText 小部件显示不同样式的文本
显示资源包里的图像
显示来自网络的图像
调整图像的显示
使用 Container 小部件
装饰容器
整理项目

页面结构

准备项目 (page-structure)
创建 Material 应用
使用 Material 小图标 (Icon)
使用按钮 (ElevatedButton) 小部件
定制应用的主题样式
构建 Material 风格的页面 (Scaffold)
设置页面顶部应用栏 (AppBar)
使用标签栏 (TabBar) 与标签视图 (TabBarView) 小部件
设置底部导航栏 (BottomNavigationBar)
把 App 转换成有状态小部件 (StatefulWidget)
点按底部导航栏项目切换当前活动项目
点按底部导航栏项目切换显示小部件
点按底部导航栏项目时动态显示或隐藏 AppBar
使用漂浮动作按钮 (FloatingActionButton)
整理项目

定义部件

准备项目 (define-widget)
准备 VSCode 编辑器代码片断
定义应用页面头部小部件 (AppPageHeader)
定义应用标志小部件 (AppLogo)
定义应用页面主体小部件 (AppPageMain)
定义标签视图小部件 (PostIndex)
定义应用页面底部小部件 (AppPageBottom)
定义应用漂浮动按钮小部件 (AppFloatingActionButton)
整理项目

弹窗对话

准备项目 (modal-dialog)
显示页面底部面板 (BottomSheet)
用漂浮动按钮显示与关闭底部面板
显示警告对话框 (AlertDialog)
显示操作提示栏 (SnackBar)
使用边栏抽屉 (Drawer)
设置边栏抽屉上显示的内容 (ListView 与 ListTile)
使用弹出菜单按钮小部件 (PopupMenuButton)
整理项目

界面布局

准备项目 (layout)
准备练习页面小部件
理解小部件的约束
准备布局项目小部件 (PlaygroundLayoutItem)
安全区域 (SafeArea), 尺寸盒子 (SizedBox)
Align: 对齐
Column: 栏 列
Row: 行 排
Expanded: 扩展空间
Stack: 堆
Positioned: 定位
整理项目

表单元素

准备项目 (input)
准备练习演示用户输入的小部件
按钮: ElevatedButton
文本字段: TextField

获取文本字段里的数据
文本表单字段: TextFormField
验证文本表单字段数据: validate()
文本编辑控制器: TextEditingController
表单: Form
整理项目

路由导航 (一)

准备项目 (routing)
准备导航与路由演示小部件
导航与路由 (Navigator & Route)
用 Navigator 命令式管理路由 (Navigator.push 与 Navigator.pop)
理解 MaterialApp 的 home 属性
定义路由表使用带名字的路由
onGenerateRoute
从路由名字中提取参数
整理项目

状态管理 (一)

准备项目 (state-management)
准备状态管理演示小部件
安装 Provider
定义状态类 (Model)
用 Provider 提供数据与方法
在小部件里使用来自 Provider 提供的数据与方法 (Provider.of)
用 ChangeNotifierProvider 提供数据与方法
在小部件里使用来自 Provider 提供的数据与方法 (Consumer)
整理项目

路由导航 (二)

准备项目 (routing_2)
使用 Navigator 声明式接口 (Pages API)
使用 MultiProvider 提供多种东西
动态添加与移除页面
创建路由器代表 (RouterDelegate)
使用路由器管理路由 (Router)
应用状态变化时通知 Router 重建 Navigator
定义路由配置类型
把路由信息转换成自定义的路由配置 (parseRouteInformation)
根据路由配置数据修改应用状态 (setNewRoutePath)
把路由配置转换成路由信息 (restoreRouteInformation)

整理项目

网络请求

准备项目 (http)

准备网络请求演示小部件

安装 http

请求服务端接口获取数据

将 JSON 数据转换成自定义的类型

请求服务端接口创建内容 (用户)

发送用户登录请求

请求服务端接口更新内容 (用户)

自动生成序列化 JSON 数据用的代码 (json_serializable)

整理项目

内容列表

准备项目 (list)

使用环境变量定义应用配置 (flutter_dotenv)

创建内容列表小部件 (PostList)

定义并提供获取内容列表数据用的方法

请求内容列表数据

定义内容数据类型

将内容列表 JSON 数据转换成一组内容 (Post) 类型的数据

使用 ListView 构建内容列表视图

整理项目

内容项目

准备项目 (list-item)

创建内容列表项目小部件 (PostListItem)

定义内容媒体小部件 (PostMedia)

定义内容头部小部件 (PostHeader)

定义用户头像小部件 (UserAvatar)

定义内容动作小部件 (PostActions)

整理项目

内容页面

准备项目 (post)

处理点按内容列表项目图像

定义单个内容数据模型 (PostShowModel)

点按内容项目图像时显示内容页面

定义内容页面主体小部件 (PageShowMain)

定义内容正文小部件 (PostContent)
定义内容标签小部件 (PostTags)
配置路由器处理内容页面
请求内容页面需要的数据
整理项目

身份验证

准备项目 (auth)
添加用户登录页面
准备登录表单小部件 (AuthLoginForm)
定义用户登录相关类型 (LoginData, Auth)
自定义网络请求异常 (HttpException)
定义身份验证模型 (AuthModel)
请求用户登录
整理项目

状态管理 (二)

准备项目 (state-management_2)
使用 Provider 的 value 构造方法提供值
在单独的文件里定义要提供的 Provider
用 Shared preferences 插件记住登录状态
应用启动以后恢复登录状态
定义应用服务与接口客户端 (AppService, ApiHttpClient)
用 ChangeNotifierProxyProvider 解决依赖
改造 PostIndexModel 用 apiHttpClient 发送请求
整理项目

内容点赞

准备项目 (like)
使用 GestureDetector 处理手势动作
定义点赞内容模型
定义取消点赞模型
定义提供点赞 Provider
处理用户点赞动作
处理用户取消点赞动作
整理项目

列表布局

准备项目 (post-list-layout)
准备热门内容列表

准备内容列表布局相关数据与方法
设置与存储内容列表布局
准备网格内容列表
准备多种布局的内容列表项目
恢复内容列表布局
整理项目

创建内容

准备项目 (post-create)
定义创建内容数据模型 (PostCreateModel)
改进 AppTextField 自定义小部件
重新定义异常
准备创建内容页面 (PostCreate)
定义创建内容表单小部件 (PostCreateForm)
发布内容
整理项目

文件上传

准备项目 (file-upload)
安装文件选择器插件 (file_picker)
选择照片应用里的图像文件
准备上传文件用的请求
定义上传文件需要的相关数据与方法
定义选择图像文件小部件 (PostCreateMedia)
使用进度指示器 (CircularProgressIndicator)
用选择的文件的名字作为内容的默认标题
是否保留未发布的内容?
创建内容后上传文件
整理项目

主题样式

准备项目 (theme)
主题颜色
图标主题 (IconThemeData)
文本主题 (TextTheme)
应用栏主题 (AppBarTheme)
根据平台当前选择的主题来设置小部件的样式
标签栏主题 (TabBarTheme)
底部导航栏主题 (BottomNavigationBarThemeData)
消息提示栏主题 (SnackBarTheme)
按钮主题 (ElevatedButtonThemeData)

改进并整理项目

发布应用

准备项目 (release)
加入苹果开发者计划
用真实设备调试应用 (iOS)
应用小图标 (iOS)
应用启动屏幕 (iOS)
注册应用 ID (iOS)
在 App Store Connect 创建应用 (iOS)
构建应用 (iOS)
修正问题后重新构建
提交发布应用 (App Store Connect)
通过 TestFlight 安装测试应用
整理项目

创建用户

准备项目 (user-create)
准备创建用户模型 (UserCreateModel)
用 Provider 提供创建用户模型
准备创建用户表单小部件 (UserCreateForm)
修改用户页面未登录时的样子
修改应用重新编译并提交审核
整理项目

Flutter 客户端应用开发实例

状态管理 (三)

准备一个 Flutter 项目
定义存储本地数据用的方法 (AppStorage)
添加应用标志新变化
安装 MobX 相关依赖
使用 MobX 管理应用状态: 创建 Store
使用 MobX 管理应用状态: 添加数据 (Observable) 与动作 (Action)
使用 MobX 管理应用状态: 添加反应 (Reaction)
使用 MobX 管理应用状态: 取消反应 (Reaction)
修改内容列表相关的几个地方
定义内容列表 Store (PostIndexStore)

添加设置内容布局时的反应
整理项目 (improve)

内容列表 (二)

准备项目 (post-list)
在内容列表页面使用 CustomScrollView
重构网格布局内容列表 (flutter_staggered_grid_view)
重构首页头部 (SliverAppBar)
创建首页头部动作 (AppHeaderSearch, AppHeaderUser)
改造获取内容列表的方法显示加载状态 (getPosts)
创建应用工具栏 (AppBar)
定义文字菜单项目小部件 (AppTextMenuItem)
定义内容列表页面菜单 (PostIndexMenu)
当排序方式变化时请求内容列表
内容列表布局切换器 (PostListLayoutSwitcher)
整理项目

滚动加载

准备项目 (infinite-scroll)
创建与使用滚动控制器 (ScrollController)
点按标题时回到页面顶部
定义加载新页面数据的滚动位置
定义几个新的环境变量与相关配置
滚动加载新页面数据
定义回到顶部小部件 (AppMoveToTop)
重置请求内容列表数据
下拉刷新
整理项目

内容页面

准备项目 (post-show)
创建内容视图 (PostContentView)
内容头部 (PostShowHeader)
内容头部标题 (PostShowHeaderTitle)
内容头部动作 (PostShowHeaderMeta, PostShowHeaderAspectRatio)
定义内容页面 Store (PostShowStore)
内容媒体比例 (PostMediaAspectRatioBottomSheet)
内容媒体 (PostShowMedia)
设置内容媒体使用 Hero 动画
定义内容页面相关小部件
整理项目

内容导航

- 准备项目 (post-show-navigation)
- 修改内容 Store (PostShowStore)
- 用内容 Store 替换内容 Model
- 使用 PageView 改造内容页面
- 使用 PageController 设置 PageView 初始页面
- 浏览内容时如果需要加载新的列表数据
- 修复: 更新 PostShowStore 以后无法显示新页面数据
- 改进: 设置内容媒体加载状态
- 向下拉拽内容页面时返回
- 整理项目

图像数据

- 准备项目 (file-metadata)
- 定义 File 与 FileMetadata 数据类型
- 定义与提供文件信息 Store (FileShowStore)
- 准备拍摄参数信息数据 (Specs)
- 文件信息数据缓存 (fileMetadataCache)
- 内容页面文件信息小部件 (PostShowFileMetadata)
- 加载文件数据时显示骨架
- 升级 Flutter 框架
- 整理项目

内容底板

- 准备项目 (post-bottom-sheet)
- 内容底板小部件 (PostBottomSheet)
- 在内容底板使用标签栏 (TabBar 与 TabBarView)
- 内容底板文件信息小部件 (PostBottomSheetFileMetadata)
- 文件信息列表小部件 (FileMetadataList)
- 显示拍摄设备数据 (相机、镜头)
- 显示拍摄相关数据 (处理软件、拍摄日期...)
- 格式化日期时间 (Intl DateFormat)
- 在内容页面显示拍摄相机
- 整理项目

图像视图

- 准备项目 (post-image-view)
- 改进: 决定要使用的内容视图
- 改进: 添加媒体全屏幕显示比例
- 创建内容图像视图小部件 (PostImageView)

隐藏系统顶部状态栏与底部导航栏（setEnabledSystemUIMode）
用 photo_view 定义可缩放的内容图像小部件（PostShowMediaZoomable）
使用 PhotoViewGallery 构建可左右导航的内容图像视图
切换页面时的页面间隔（FractionallySizedBox）
修复：内容页面视图与内容图像页面视图不同步
整理项目

沉浸模式

准备项目（immersive-mode）
改进：用内容图像视图用 Stack 代替 CustomScrollView
添加表示使用沉浸式模式的数据（immersiveMode）
内容全屏顶部小部件（PostShowFullscreenTop）
创建内容全屏底部小部件（PostShowFullscreenBottom）
内容全屏底部动作小部件（PostShowFullscreenBottomActions）
内容全屏底部小部件的内容
媒体比例变化时设置系统界面
整理项目

评论列表

准备项目（comment-list）
点按评论动作显示底板评论
创建评论底板与评论列表首页小部件（PostBottomSheetComment, CommentIndex）
定义评论数据类型（Comment）
定义评论列表 Store（CommentIndexStore）
用 Provider 提供 CommentIndexStore
加载与显示评论列表
评论列表小部件（CommentList）
评论列表项目小部件（CommentListItem）
评论列表项目骨架小部件（CommentListItemSkeleton）
改进：内容底板的边距
整理项目

回复列表

准备项目（reply-list）
创建评论列表项目动作小部件（CommentListItemActions）
定义回复数据类型（Reply）
定义评论列表 Store（ReplyIndexStore）
用 Provider 提供 ReplyIndexStore
切换显示回复列表
加载回复列表数据
创建进度指示器小部件（AppIndicator）

回复列表小部件 (ReplyList)
回复列表项目小部件 (ReplyListItem)
整理项目

修复改进

准备项目 (fix-improve)
定义用户数据类型 (User)
改进: 使用 User 类型替换内容作者相关类型
定义订阅图标小部件 (SubscriptionIcon)
定义用户名小部件 (UserName)
改进: 使用用户名小部件
改进: 定义应用导航器小部件 (AppNavigator)
改进与修复: 定义应用路由器小部件 (AppRouter)
改进: 把 AuthModel 放在 appProviders 里提供
添加应用初始化页面 (AppInitialize)
改进: 新建 authModelProvider 提供 AuthModel
改进: 将应用初始化放在 AppRouter 里执行
定义身份验证 Store (AuthStore)
用 AuthStore 替换 AuthModel
整理项目

用户信息

准备项目 (user-info)
定义用户页 Store (UserShowStore)
定义用户页小部件 (UserShow)
在页面列表里添加用户页 (userShowPage)
处理用户页路由信息
处理用户登录与注册路由信息
改进: 处理用户未设置头像的情况 (UserAvatar)
定义用户 Store (UserStore)
获取当前登录用户数据
整理项目

创建评论

准备项目 (comment-create)
定义与提供创建评论 Store (CommentCreateStore)
创建评论相关小部件
创建评论的作者头像 (CommentCreateThumbnail)
定义创建评论内容字段小部件 (CommentCreateContentField)
定义创建评论动作小部件 (CommentCreateActions)
改进: 应用按钮小部件 (AppButton)

提交创建评论
验证评论内容
根据字段焦点状态显示创建评论动作
用户未登录时显示注册与登录动作
创建内容底板小部件时设置评论所属内容 ID
整理项目

编辑评论

准备项目 (comment-edit)
定义与提供编辑评论 Store (CommentEditStore)
创建评论编辑相关小部件
定义评论项目的编辑动作
编辑与取消编辑评论
定义编辑评论内容字段小部件 (CommentEditContentField)
定义编辑评论动作小部件 (CommentEditActions)
定义删除指定评论项目的动作 (removeComment)
定义与提供删除评论 Store (CommentDestroyStore)
删除评论
删除评论后的清理 (onDelete)
整理项目

创建回复

准备项目 (reply-create)
定义增加与减少回复数量的动作 (increaseTotalReplies、decreaseTotalReplies)
定义并提供创建回复 Store (ReplyCreateStore)
切换显示创建回复
定义创建回复小部件 (ReplyCreate)
创建回复相关小部件
定义创建回复字段小部件 (ReplyCreateContentField)
定义创建回复动作小部件 (ReplyCreateActions)
整理项目

编辑回复

准备项目 (reply-edit)
创建回复编辑小部件 (ReplyEdit)
切换显示回复编辑
定义回复编辑小部件 (ReplyEdit)
定义回复编辑内容字段小部件 (ReplyEditContentField)
定义回复编辑动作小部件 (ReplyEditActions)
删除回复
删除回复后更新界面

修复：设置评论列表项目的 key
整理项目

实时功能

准备项目 (realtime)
修复：初次加载内容列表数据时没有使用用户令牌
安装 Socket.io 客户端
创建 Socket
测试 Socket 客户端与服务端的连接
在应用接口客户端添加 X-Socket-Id 头部
定义更新点赞数量与状态的动作
监听服务端应用的实时事件
定义检查是否为当前用户的方法 (isCurrentUser)
实时更新点赞数量与状态
实时更新点赞数量与状态：内容页
整理项目

实时评论

准备项目 (realtime-comment)
定义设置评论数量用的动作 (setTotalComments)
实时更新评论数量
定义在评论列表里添加新评论用的动作 (addComment)
实时添加或删除评论项目
有回复时实时更新内容评论数量
实时更新评论的回复数量
定义添加回复用的动作 (addReplyCache)
实时添加与删除回复
整理项目

用户档案

创建用户展示小部件 (UserShow)
准备项目 (user-profile)
创建用户档案相关小部件 (UserProfile)
用户档案内容小部件 (UserProfileContent)
准备用户设置页面 (UserProfileSetting)
用户设置：退出登录
在用户 Store 增加订阅相关数据
创建用户档案头部相关小部件 (UserProfileHeader)
用户档案头部 (UserProfileHeaderThumbnail)
用户档案头部 (UserProfileHeaderContent)
设置应用主题的禁用颜色 (disabledColor)

设置应用主题的溅墨与高亮颜色（ splashColor, highlightColor）
整理项目

用户帐户

准备项目（user-profile-account）
准备用户帐户页面（UserProfileAccount）
准备上传头像方法（UploadAvatar）
定义用户档案帐户 Store（UserProfileAccountStore）
改进：让用户头像使用选择的头像
创建用户档案帐户相关底板小部件
定义用户档案帐户小部件（UserProfileAccount）
修改用户头像（ChangeAvatarBottomSheet）
修改用户名（ChangeNameBottomSheet）
修改密码（ChangePasswordBottomSheet）
修复：在底板文本字段无法显示设备键盘
改进：用媒体查询改进底板边距与最小高度
整理项目

内容过滤

准备项目（post-list-filter）
定义内容列表过滤器类型（PostListFilter）
在内容列表 Store 里添加内容过滤
根据过滤器设置应用头部标题（AppHeaderTitle）
点击内容标签设置内容列表过滤器
创建内容列表过滤器小部件（PostListFilter）
点按过滤器内容重置过滤器
过滤器有变化时设置内容列表页大标题
设置相机与镜头过滤器
整理项目

修复改进（二）

准备项目（fix-improve-2）
在 appModel 里添加 setPostShow
问题：内容页显示错误的内容
解决：内容页显示错误的内容
修复：未恢复内容图像显示比例
改进：设置内容列表媒体比例
创建内容列表项目骨架小部件（PostListItemSkeleton）
在内容列表页面使用内容骨架（网格布局）
在内容列表页面使用内容骨架（堆叠布局）
定义与使用内容视图骨架（PostContentViewSkeleton）

整理项目

用户页面

准备项目 (user-show)

创建用户页相关小部件 (UserShowHeader, UserShowContent)

定义用户页头部骨架小部件 (UserShowHeaderSkeleton)

定义用户页头部小部件 (UserShowHeader)

创建用户页标签视图相关小部件

定义用户页内容小部件 (UserShowContent)

显示用户页的指定标签内容

准备作品与点赞列表数据

改造内容网格列表小部件 (PostGridList)

定义用户作品与点赞小部件 (UserPosts, UserLiked)

重置内容列表过滤器

重置用户页作品与点赞列表数据

准备用户页主体的滚动控制器

用户页滚动无限加载作品与点赞数据

整理项目

用户评论

准备项目 (user-comments)

准备用户评论列表与回复列表数据

定义用户评论页小部件 (UserComments)

滚动无限加载用户评论列表数据

在评论列表项目里添加内容标题

改进用户页评论与回复的相关操作

监听用户页标签变化设置 tabIndex

实时更新用户页评论

用户页回复列表 (UserReplies)

整理项目

应用搜索

准备项目 (app-search)

创建应用搜索 Store (AppSearchStore)

显示搜索字段小部件 (AppSearchField)

显示搜索菜单小部件 (AppSearchMenu)

隐藏搜索字段与搜索菜单小部件

创建搜索相关小部件 (AppSearch)

显示与关闭搜索底板

搜索底板的限制

搜索底板的样式

搜索时设置页面标题
准备搜索选项相关数据与动作
搜索选项菜单小部件 (AppSearchMenu)
准备搜索结果相关数据
定义搜索方法
定义搜索字段小部件 (AppSearchField)
执行搜索
搜索页头部小部件 (AppSearchHeader)
搜索页主体小部件 (AppSearchContent)
搜索结果
点按搜索结果
整理项目

创建内容（二）

准备项目 (post-create)
准备创建内容相关小部件
定义创建内容头部标题小部件 (PostCreateHeaderTitle)
定义创建内容头部提交小部件 (PostCreateHeaderSubmit)
定义创建内容主体小部件 (PostCreateContent)
定义创建内容 Store (PostCreateStore)
提交创建内容
内容发布状态
显示内容状态 (PostCreateHeaderStatus)
创建内容页脚小部件 (PostCreateFooter)
查看内容图像时验证用户身份
整理项目

内容标签

准备项目 (post-tags)
创建内容标签字段 (PostTagField)
创建并提供标签 Store (TagStore)
定义在标签列表中添加与移除标签用的动作
添加并显示标签列表
从标签列表里移除标签
定义给内容打标签与删除标签用的方法
给内容打上标签
删除内容标签
整理项目

内容编辑

准备项目 (post-edit)

内容编辑模式
更新内容
在编辑模式添加内容标签
是否保留正在编辑的内容
创建内容头部更多小部件 (PostCreateHeaderMore)
处理弹出菜单里的新建与删除内容
控制文本字段的验证
清理并整理项目

Nest.js 服务端应用开发训练路线

Nest.js 服务端应用开发实例

准备项目

- 创建并运行 Nest.js 项目
- 准备 Http 客户端测试应用接口
- 创建并添加远程仓库
- 修改应用目录结构
- 准备应用配置（基于 @nestjsconfig）
- 使用单独配置文件
- 创建应用数据仓库
- 准备数据库相关配置
- 安装配置 TypeOrm
- 配置 TypeOrm 同步与自动加载实体

应用构架

- 准备项目（architecture）
- 使用 CQRS 应用架构
- 准备项目命令行工具（@ninghaoschematics）
- 模块（Module）
- 控制器（Controller）
- 实体（Entity）
- 命令（Command）
- 事件（Event）
- 连串事件（Saga）
- 定义带参数的接口
- 查询（Query）
- 使用 Repository 处理实体数据
- 使用 QueryBuilder 处理实体数据
- 整理项目

用户注册

- 准备项目（register）
- 准备代码片断，调整应用目录结构
- 定义注册用户接口

存储用户数据
Hash 用户密码
在数据表内限制用户名必须唯一
验证数据 (class-validator)
自定义验证器
自定义验证装饰器
验证用户不存在
整理项目

用户登录

准备项目 (login)
创建身份验证模块 (AuthModule)
创建本地身份验证策略 (用户名与密码)
使用本地身份验证策略 (AuthLocalStrategy)
执行验证用户登录凭证命令
定义获取当前用户的参数装饰器
整理项目

身份验证

准备项目 (auth)
准备签发验证令牌用的密钥对
安装配置 JWT 模块 (@nestjs/jwt)
定义与执行签发令牌 (JWT) 的命令
定义 Jwt 身份验证策略
测试使用 Jwt 身份验证策略
整理项目

用户头像

准备项目 (avatar)
创建文件上传模块 (AppUploadModule)
准备文件过滤器
创建头像模块 (AvatarModule)
定义上传头像接口
将上传模块改为动态模块
定义头像实体 (AvatarEntity)
存储头像文件相关数据
定义头像服务接口
整理项目

图像处理

准备项目 (image-process)
创建应用图像模块 (AppImageModule)
定义缩放 (调整大小) 图像命令 (ResizeImageCommand)
定义缩放图像命令处理器
创建并发布头像已创建事件 (AvatarCreatedEvent)
处理头像已创建事件
测试调整头像大小
提供不同尺寸的头像
整理项目

用户帐户

准备项目 (user-account)
修改单个用户接口 (添加头像)
创建更新用户模块与接口
验证更新用户数据
创建更新用户命令 (UpdateUserCommand)
更新用户时检查用户与密码
更新用户名
更新用户密码
整理项目

内容管理

准备项目 (manage-post)
创建内容管理相关模块
定义内容实体 (PostEntity)
创建发布内容接口
验证创建内容数据
创建内容
更新内容
删除内容
整理项目

权限控制

准备项目 (access-control)
创建权限控制模块 (AppAccessControlModule)
用 CASL 描述用户权限: 定义动作与主题类型
用 CASL 描述用户权限: 定义用户能力
定义测试用户权限接口
测试用户权限
定义权限策略类型与装饰器 (UsePolicies)

定义检查权限策略守卫 (PolicyGuard)
测试检查权限策略
设置删除内容接口的权限
整理项目

文件上传

准备项目 (file)
创建文件相关模块与实体
定义文件实体 (FileEntity)
文件上传
准备提取图像文件元数据
定义提取图像文件元数据命令 (ExtractImageMetadataCommand)
定义创建文件命令 (CreateFileCommand)
验证创建文件接口的查询符
检查用户对文件关联的内容的权限
保存文件数据
整理项目

文件服务

准备项目 (file-serve)
调整图像文件大小
定义文件服务 (FileService)
准备 ServeFileQueryDto 与 GetFileQuery
定义文件服务接口
定义文件元数据接口
定义使用可选 JWT 身份验证守卫
定义使用文件访问守卫 (FileAccessGuard)
整理项目

内容标签

准备项目 (tags)
创建标签相关模块与实体
定义创建标签命令与获取标签查询
创建内容标签模块与实体
定义创建内容标签接口
定义删除内容标签接口
整理项目
准备项目 (likes)
创建用户点赞内容模块
定义用户点赞内容实体 (UserLikePost)
定义创建与删除用户点赞内容命令

定义创建与删除用户点赞内容接口
整理项目

发表评论

准备项目 (comments)
创建评论相关模块与实体
定义评论实体 (CommentEntity)
改进权限策略守卫 (PolicyGuard)
改进当前用户装饰器 (CurrentUser)
定义创建评论接口
描述用户评论权限
定义更新评论接口
定义删除评论接口
整理项目

评论回复

准备项目 (replies)
回复与评论
创建评论回复模块 (CommentReplyModule)
描述用户的评论回复权限
定义评论回复接口
定义获取评论回复列表查询
定义获取评论回复列表接口
整理项目

内容列表

准备项目 (lists)
准备项目演示数据
创建内容列表模块与接口
内容列表分页
提供 SQL 片断 (PostProvider)
内容列表排序
在内容项目里添加相关数据 (作者、文件、标签)
当前用户是否赞过内容
过滤内容列表
按标签过滤内容列表
用户发布与点赞的内容列表
过滤出指定拍摄设备的内容列表
内容列表的普通管理模式与管理员模式
统计内容列表项目数量
整理项目

资源接口

准备项目 (resources)
定义单个内容接口
删除内容后删除相关文件
创建删除用户模块
定义删除用户命令 (DeleteUserCommand)
定义删除用户接口
删除用户以后删除相关文件与头像
定义评论列表模块 (CommentIndexModule)
获取评论列表查询预处理
处理获取评论列表查询
定义评论列表接口
内容评论列表的访问控制
整理项目

内容审核

准备项目 (audit)
用户能力拦截器 (UserAbilityInterceptor)
使用 nn 命令快速创建零部件
创建审核相关模块
定义审核日志实体 (AuditLogEntity)
定义创建审核命令 (CreateAuditCommand)
用户的提交审核权限
定义提交审核接口
内容列表接口的审核状态
用户的取消审核权限
定义取消审核命令
定义取消审核接口
整理项目

资源搜索

准备项目 (search)
搜索标签
搜索用户
搜索相机
搜索镜头
整理项目

访问日志

准备项目 (access)

改进：转换响应里的某些属性的数据类型

准备客户端应用

创建访问日志模块

定义访问日志实体（AccessLogEntity）

定义访问日志装饰器（AccessLog）

创建访问拦截器（AccessInterceptor）

定义与使用访问拦截器

记录接口的访问情况

整理项目

访问统计

准备项目（dashboard）

创建仪表模块（DashboardModule）

创建仪表访问模块（DashboardAccessModule）

定义获取统计列表查询（GetAccessCountsQuery）

定义按动作统计访问次数接口

定义按动作获取统计次数查询

整理项目

实时服务

准备项目（realtime）

准备 WebSocket 实时服务（Socket.io）

实时点赞与取消点赞

定义按 ID 获取评论查询（GetCommentByIdQuery）

实时评论

实时回复

实时更新评论或回复

实时删除评论或回复

实时访问统计

整理项目

报名参加

报名参加宁皓网独立开发者训练营（全栈），[立即报名](#) →

报名咨询

登录 [宁皓网](#) 或关注 [宁皓网](#) 微信公众号可以获得训练营最新情况。

咨询详情可以添加微信好友，搜索用户：wanghao_8088，或用微信扫描下面的二维码。



王皓

wanghao_8088 ✓



添加王皓为微信好友