

4.1 搭建 UI 框架（一）： 如何基于 vant 快速创建小程序页面？



扫码试看/订阅

《微信小程序全栈开发实战》视频课程

<https://youzan.github.io/vant-weapp/#/intro>

vant 组件示例：



取消	标题	确认
北京市		
天津市	请选择	请选择
河北省	太原市	新荣区
山西省	大同市	平城区
内蒙古自治区	阳泉市	云冈区
辽宁省	长治市	云州区

重新截图

npm init

npm i @vant/weapp -S --production

构建 npm

修改项目本地配置

修改 app.json 配置

引用并使用

省市区数据: <https://github.com/youzan/vant/blob/dev/src/area/demo/area.js>



重新截图

4.2 搭建 UI 框架（二）： 使用 art-template 模板引擎

添加 vant 全局样式:

```
@import '/miniprogram_npm/@vant/weapp/common/index.wxss';
```

边框样式:

```
border-top: 1px solid #ddd
```

```
ejs:  
<h1><%= title %></h1>
```

```
art-template:  
<h1>{{title}}</h1>
```

后端使用 art-template:

```
npm install --save art-template
```

```
npm install --save koa-art-template
```

```
npm install html-minifier -S
```

文档:

<https://aui.github.io/art-template/zh-cn/docs/index.html>

4.3 搭建 UI 框架（三）：规整后端 koa 项目结构

```
server/  
- config/  
- controllers/  
- views/  
- lib/  
- models/  
- routers/  
- static/  
  - css/  
  - js/  
  - images/
```

4.4 用户登录（一）：创建与使用数据库

ORM = Object/Relational Mapping

1. 安装 MySQL 及连接工具 MySQLWorkbench
2. 创建 MySQL 数据库 (practice)
3. 安装 sequelize、及 MySQL 中间件
4. 创建 mysql-config 配置
5. 向库中写入 user 对象

<https://dev.mysql.com/downloads/mysql/>

<https://dev.mysql.com/downloads/workbench/>

```
npm i sequelize mysql2 -D
```

sequelize 文档:

<https://sequelize.org/master/manual/model-basics.html>

中文文档:

<https://github.com/demopark/sequelize-docs-Zh-CN/tree/master>

Node.js:

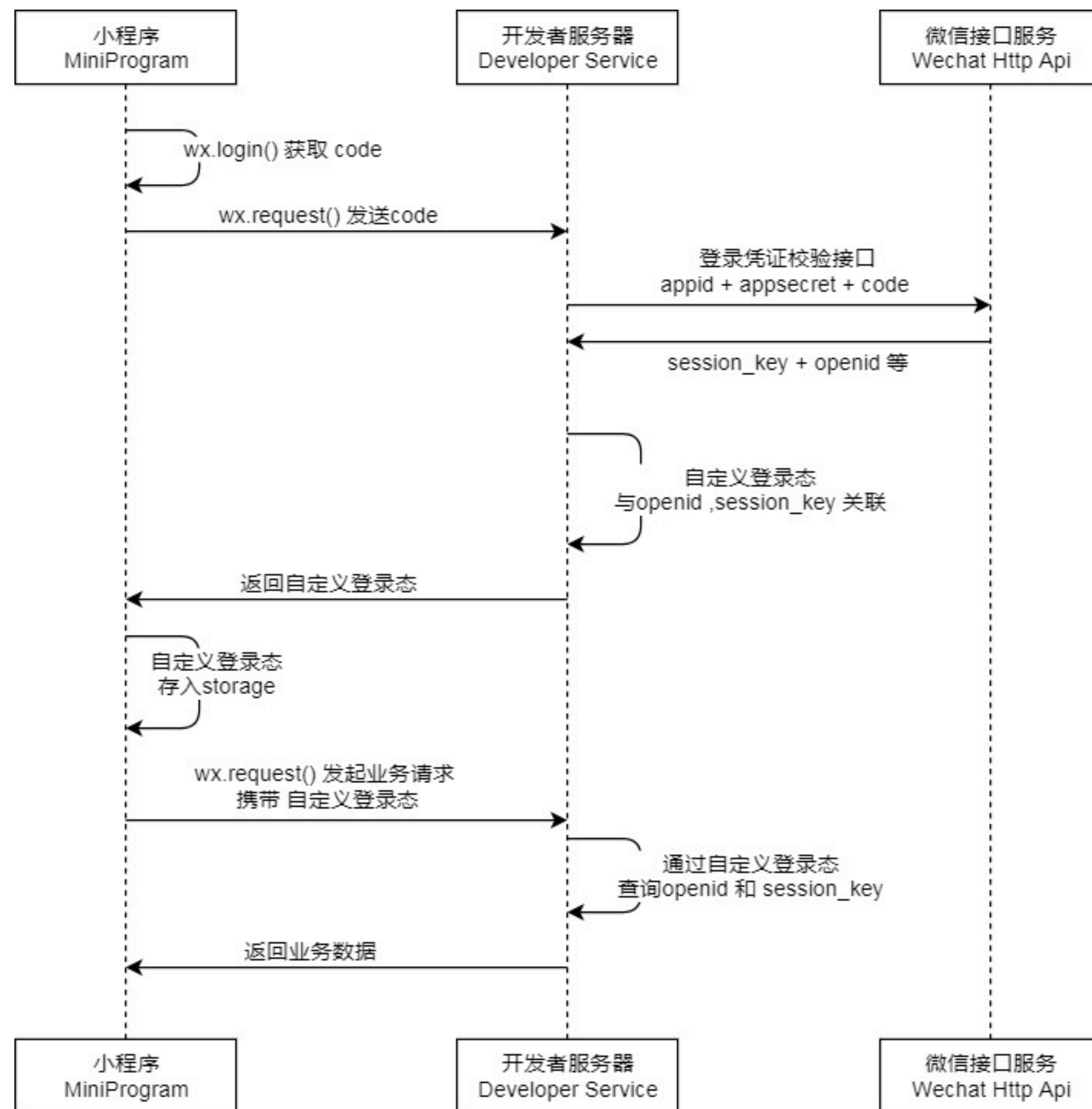
```
module.exports = db.define('user', {  
  id: {  
    type: Sequelize.INTEGER(11),  
    allowNull: false,  
    primaryKey: true,  
    autoIncrement: true,  
  },  
  nickName: {  
    type: Sequelize.STRING(50),  
    allowNull: false  
  },  
  avatarUrl: {  
    type: Sequelize.STRING,  
    allowNull: false  
  },  
  ...  
})
```

Golang:

```
type User struct {  
    Id int64  
    Name string  
    CreatedAt time.Time `xorm:"created"`  
}
```

```
findOne() // 查询单项  
findAll() // 查询所有  
findById() // 依据 id 查询  
findOrCreate() // 查找或保存新实例  
findOrCreateBuild() // 查找或构建新实例  
findAndCountAll() // 返回符合条件的记录，并返回总数  
create() // 创建  
bulkCreate() // 批量创建  
update() // 更新  
upsert() // 是 update&insert 的合称  
destroy() // 销毁记录  
increment() // 递增更新  
decrement() // 递减更新  
count() // 统计数量  
max() // 取某字段最大值记录  
min() // 取某字段最小值记录  
sum() // 统计求和
```

4.5 用户登录（二）： 创建 sessionKey model 并完善用户登录



4.6 商品列表页：创建分类模型



中国联通

13:02

83%

故宫宫廷文化

生活方式

文创好物

宫廷美玉

甄选珠宝

宫廷服饰

创意玩具

微瑕特卖

生活方式

一盖欢喜 夏凉被



¥ 79

海错图 双层流沙冷饮杯 流沙杯垫套装



¥ 49

紫禁御喵 小毛巾 面巾 浴巾



¥ 12.9

鹤鹿同春 餐具套装



¥ 398

御花园 方巾 面巾 毛巾



¥ 12.9

故宫

首页

分类

购物车

我的

分类
商品

4.7 商品列表页：使用存储过程初始化分类表

4.8 商品列表页：创建商品表、品牌表

```
[{"category_name": "\\生活方式\\", "id": 1}, {"category_name": "\\文创好物\\", "id": 2}, {"category_name": "\\创意玩具\\", "id": 3}, {"category_name": "\\微瑕特卖\\", "id": 4}, {"category_name": "\\宫廷服饰\\", "id": 5}, {"category_name": "\\甄选珠宝\\", "id": 6}, {"category_name": "\\宫廷美玉\\", "id": 7}]
```

```

DataTypes.STRING      // VARCHAR(255)

DataTypes.TEXT        // TEXT

DataTypes.TEXT("tiny"), // TINYTEXT

DataTypes.DATE        // DATETIME, Sequelize.NOW

DataTypes.DATEONLY    // 不带时间的 DATE

DataTypes.BOOLEAN     // TINYINT(1)

DataTypes.INTEGER     // INTEGER

DataTypes.BIGINT      // BIGINT

DataTypes.BIGINT(11)   // BIGINT(11)

DataTypes.FLOAT       // FLOAT

DataTypes.FLOAT(11)    // FLOAT(11)

DataTypes.FLOAT(11, 10) // FLOAT(11,10)

DataTypes.DOUBLE      // DOUBLE

DataTypes.DOUBLE(11)   // DOUBLE(11)

DataTypes.DOUBLE(11, 10) // DOUBLE(11,10)

DataTypes.DECIMAL     // DECIMAL

DataTypes.DECIMAL(10, 2) // DECIMAL(10,2)
```

FLOAT、DOUBLE、DECIMAL

品牌表

id

brand_name

商品表

id

spu_no varchar(50) 商品编号，唯一

goods_name varchar(50) 商品名称

start_price decimal(9,2) 起始售价

category_id bigint(20) 分类id

brand_id bigint(20) 品牌id

4.9 商品列表页：创建商品描述表



4.10 商品列表页：创建商品 SKU 表



¥ 79.00 限时折扣

价格¥119.00

剩余 47 件

已选 粉色; 120*200cm*240g

颜色:

蓝色

粉色

尺码:

120*200cm*240g

150*200cm*300g

180*200cm*360g

购买数量:

—

1

+

加入购物车

立即购买

“Any problem in computer science can be solved
by another layer of indirection.”

SPU

SKU

规格表 **goods_attr_key**

id

goods_id 商品id

attr_name varchar(50) 规格名称

规格值表 **goods_attr_value**

id

goods_id 商品id

attr_key_id 规格id

attr_value varchar(50) 规格值

goods_sku 表

id

goods_id

goods_attr_path 规格搭配路径

price 价格

stock 库存



扫码试看/订阅

《微信小程序全栈开发实战》视频课程