

## 课前准备

---

- 工具
  - 微信开发者工具
  - 微信小程序账号
- 前置知识
  - html、css基础知识
  - js基础知识

## 课堂主题

---

- 小程序云存储
- 小程序云函数
- mpvue构建项目
- 定位功能实现
- 城市列表选择实现
- 商铺列表实现
- 商铺详细页面
- 上拉加载更多

## 云开发

- 云存储
  - 数据库crud
    - 增加

```
db.collection('news').add({
  data:{
    title: this.data.title,
    content: this.data.content
  }
})
```

- 修改

```
db.collection('news').doc("0ec685215e369f22094d528257476fcb").update({
  data:{content: "修改的内容"}
}).then(res=>{
  console.log(res);
})
```

- 删除

```
db.collection("news").doc("0ec685215e369f22094d528257476fcb").remove().then(res=>{
    console.log(res);
})
```

#### ■ 查询

```
db.collection("news").get().then(res=>{
    console.log(res);
})
```

- 云函数

- 一个云函数的写法与一个在本地定义的 JavaScript 方法无异，代码运行在云端 Node.js 中

## mpvue构建项目

- mpvue简介

- 是一个使用 [Vue.js](#) 开发小程序的前端框架。框架基于 `vue.js` 核心，`mpvue` 修改了 `vue.js` 的 [runtime](#) 和 [compiler](#) 实现，使其可以运行在小程序环境中，从而为小程序开发引入了整套 `vue.js` 开发体验

- mpvue的使用

```
# 全局安装 vue-cli
$ npm install --global vue-cli

# 创建一个基于 mpvue-quickstart 模板的新项目
$ vue init mpvue/mpvue-quickstart my-project

# 安装依赖
$ cd my-project
$ npm install

# 启动构建
$ npm run dev
```

## 实现定位及城市选择功能

- 点击位置跳转城市页面；

```
wx.navigateTo({
  url: "/pages/city/main"
})
```

- 实现定位功能；

- 获取定位信息

```
wx.getLocation({
  type: "wgs84",
  success(res) {
    console.log(res);
  }
})
```

- app.json配置permission字段

```
"pages": ["pages/index/index"],
"permission": {
  "scope.userLocation": {
    "desc": "你的位置信息将用于小程序位置接口的效果展示" // 高速公路行驶持续后台定位
  }
}
```

- 将经纬度转换成地址；

- [腾讯地图小程序JavaScript SDK](#)
- 申请地图申请开发者密钥 ( key )
- 下载微信小程序JavaScriptSDK [微信小程序JavaScriptSDK v1.2](#)
- 引入qqmap-wx-jssdk
  - 解决mpvue不支持 commonjs规范
  - 修改 .babelrc 文件, plugins添加 "transform-es2015-modules-commonjs"

```
"plugins": ["transform-runtime", "transform-es2015-modules-commonjs"]
```

- 逆地址解析reverseGeocoder

```
qqmapsdk.reverseGeocoder({
  location: {
    latitude: res.latitude,
    longitude: res.longitude
  },
  success: res => {
    console.log(res);
  }
})
```

- 渲染城市选择页面

- 引入城市数据cityData.js
- 循环数据

```
<div class="letter" v-for="(item,key) in cityData" :key="key">
</div>
```

- 点击字母定位到对应的城市位置；
  - 通过scroll-view来实现；

- scroll-into-view:值应为某子元素id ( id不能以数字开头)
- 点击选中城市
  - 通过全局变量来带参数

```
let app = getApp();
app.globalData.address = "北京"
```

- 返还上层目录

```
wx.navigateBack({
  delta: 1
})
```

- 主页显示

```
onShow(){
  if( typeof app.globalData.address !== 'undefined'){
    this.loca = app.globalData.address;
  }
},
```

## 实现列表及详细页面

- 自定义导航 "navigationStyle":"custom" [页面配置](#)
- 云储存数据
  - 初始化云开发
    - 在 src/main.js 文件添加以下语句

```
wx.cloud.init({
  traceUser: true
})
```

- 添加数据
  - 导入图片资源
  - 添加列表数据
    - 处理json文件
    - 导入json文件
- 获取云列表数据

- 初始化数据库

```
const db = wx.cloud.database()
```

- 查询数据库

```
db.collection('myremark').get().then(res=>{
  console.log(res);
})
```

- 获取详细内容

- 实现列表上拉加载更多

```
onReachBottom() {}
```

## 总结

---

- mpvue构建项目
- 定位功能实现
- 城市列表选择实现
- 商铺列表实现
- 商铺详细页面
- 上拉加载更多

## 下节预告

---

小程序四