

# uniapp03

## 复习

uniapp 基于vue.js的 `vue` 相关生态它基本都可以使用. 微信小程序 的一些语法. 基础组件view....路由系统通过 `pages.json`进行管理的.

封装的一些API方法 以uni开头的

生命周期:

应用级别 `onLaunch` `onShow` `onHide` .....

页面级别 `onLoad` `onPullDownRefresh`[下拉刷新] `onReachBottom` `onPageScroll`

组件级别 创建 挂载 更新 销毁

`mounted` 发送请求

组件库 uni-ui 看文档 对应的参数和标签 写出想要效果即可

徽章 小角标

card组件 更新了 根据文档的参数 进行调整使用 布局

网络请求 客户端[浏览器] 基于JS的发送请求 `ajax` `xmlhttprequest` 对象

```
jquery $.ajax()
vue axios
wx wx.request
uniapp uni.request
```

折叠栏 折叠菜单

## 组件库语法学习

折叠栏

案例:显示一个LOL英雄列表

拿到LOL英雄的数据,发请求

接口地址: 浏览器测试一下是否可以打开,如果不能打开可能是网络问题,中心的网络,可以咨询mis老师

```
https://game.gtimg.cn/images/lol/act/img/js/heroList/hero_list.js?v=43
```

```
<template>
  <!-- 完成一个案例效果 -->
```

```

<!-- 发送请求=>拿数据=>存本地=>做展示 -->
<!-- html地方 -->
<view>
  <!-- 4.做展示 -->
  <!-- ucoll -->
  <!-- accordion 手风琴效果 只打开一个 其它关闭 -->
  <uni-collapse accordion>
    <!-- key 唯一标识 使用尽量短的 一般使用index -->
    <!-- : 动态的属性绑定 vue会进行解析 js代码 v-bind 缩写形式 -->
    <uni-collapse-item
      v-for="(item, index) in data"
      :key="item.heroId"
      :title="item.title"
      :thumb="
        `https://game.gtimg.cn/images/lol/act/img/champion/${
          item.alias
        }.png`
      "
    >
      <!-- 写两个按钮 -->
      <view>
        <!-- item.banAudio 英雄的ban音频地址 -->
        <button
          type="primary"
          size="mini"
          @click="playAudio(item.banAudio)"
        >
          BAN
        </button>
        <button
          type="warn"
          size="mini"
          @click="playAudio(item.selectAudio)"
        >
          PICK
        </button>
      </view>
    </uni-collapse-item>
  </uni-collapse>
</view>
</template>

<script>
// JS代码
export default {
  // option 配置项 属于同等级别
  data() {
    // 这里的数据可以在模板中进行渲染显示
    return {
      // 这里的key定义一般用data result... 数据结果
      // 初始化 声明一下 null 为空 啥都没有
      data: null,
      // 防止多个音频同时播放 使用同一个音频对象

```

```

        audio: uni.createInnerAudioContext()
    };
},
mounted() {
    // 调用方法
    this.getData();
},
methods: {
    // 播放音频
    playAudio(url) {
        // url 形参 看具体调用的方法传入的参数
        // 创建一个音频对象
        // let audio = uni.createInnerAudioContext()
        // 音频地址
        this.audio.src = url;
        // 再播放之前先停掉之前的 避免多个音频同时播放
        this.audio.stop();
        // 手动开启播放
        this.audio.play();
    },
    // 发请求的处理方法
    getData() {
        // uniapp 1.发送请求
        // urequest
        uni.request({
            // 接口地址
            url:
                'https://game.gtimg.cn/images/lol/act/img/js/heroList/hero_list.js?v=43',
            // 请求方法 POST GET
            // Restful接口风格 四种请方式 PUT DELETE GET POST
            method: 'GET',
            // 携带参数 没有可以不传
            data: {},
            // 成功后执行
            success: res => {
                // 2.拿数据
                // console.log(res);
                // 3.存本地
                this.data = res.data.hero;
            },
            // 失败后执行
            fail: () => {},
            // 请求完毕后执行
            complete: () => {}
        });
    }
}
};
</script>

<style scoped>
/* 样式 */
</style>

```

## 日期格式化

从服务器接口拿到的数据是unix时间戳(1970开始到现在经过的时间[JS 单位ms])

```
<template>
  <view>
    <!-- udate 时间处理组件-->
    <uni-dateformat :date="time"/>
    <br/>
    <!-- 自定义格式 -->
    <uni-dateformat :date="time" format="yyyy年MM月dd日"/>
    <br/>
    <uni-dateformat :date="time" format="yyyy年MM月dd日 hh:mm"/>
  </view>
</template>

<script>
export default {
  data() {
    return {
      //获取时间戳 JS代码语法
      time: new Date().getTime()
    };
  }
};
</script>

<style scoped></style>
```

## 通告栏

在应用顶部有一个滚动的通知

```
<template>
  <view>
    <!-- 通告栏 -->
    <!-- unotice -->
    <!-- 属性绑定 : 值为非字符串 都需要绑定 -->
    <uni-notice-bar
      single
      scrollable
      :speed="50"
      showClose
      showIcon
      color="#ff0313"
      backgroundColor="#fd8f6d"
      text="欢迎WEB2107班级同学学习到框架项目阶段,希望能够坚持学习找到好的工作!!!"
    ></uni-notice-bar>
  </view>
</template>
```

```
<script>
export default {};
</script>

<style scoped></style>
```

## 列表

在一些数据展示布局中,经常要展示多个条目,可以使用列表组件进行布局

```
<template>
  <view>
    <!-- ulist 列表 -->
    <uni-list>
      <!-- clickable 可点击的  showArrow 显示箭头 -->
      <uni-list-item title="账号和安全" click showArrow></uni-list-item>
      <!-- rightText 右侧文字 -->
      <uni-list-item
        title="青少年模式"
        rightText="未开启"
      ></uni-list-item>
      <uni-list-item
        title="微信安全中心"
        note="如果遇到微信被盗,无法登陆的情况,可以前往中心申诉"
      ></uni-list-item>
      <uni-list-item
        title="扫一扫"
        thumb="/static/logo.png"
        showArrow
      ></uni-list-item>
      <uni-list-item
        title="雷达添加好友"
        thumb="/static/logo.png"
        thumbSize="lg"
        showArrow
        clickable
      ></uni-list-item>
      <!-- showSwitch 显示开关 -->
      <!-- bluetooth_text 显示文本 已开启 已关闭-->
      <!-- bluetooth_status bool类型 true 开启 false 关闭 -->
      <!-- bluetoothChange 切换开关触发的处理方法 -->
      <!-- @ vue 对于事件的处理 v-on 缩写 -->
      <uni-list-item
        title="蓝牙"
        showSwitch
        :rightText="bluetooth_text"
        :switchChecked="bluetooth_status"
        @switchChange="bluetoothChange"
      ></uni-list-item>
      <!-- 显示扩展图标 -->
      <uni-list-item
        title="10000"
      ></uni-list-item>
```

```

        showExtraIcon
        :extraIcon="{
            color: 'rgb(18, 183, 245)',
            size: '24',
            type: 'qq'
        }"
    ></uni-list-item>
<uni-list-item
    title="10000"
    showExtraIcon
    :extraIcon="{
        color: 'rgb(0, 255, 0)',
        size: '24',
        type: 'weixin'
    }"
></uni-list-item>
<uni-list-item
    title="10000"
    showExtraIcon
    :extraIcon="{
        color: 'rgb(255, 0, 0)',
        size: '24',
        type: 'weibo'
    }"
></uni-list-item>
<uni-list-item
    title="请点击下载"
    showExtraIcon
    :extraIcon="{
        color: 'orangered',
        size: '30',
        type: 'cloud-download'
    }"
></uni-list-item>
<!-- 自定义插槽 插槽 提供更多的订制方式 -->
<!-- direction 布局排列方式 column 竖向 row 横向 -->
<uni-list-item direction>
    <template slot="header">
        <view style="background-color: #007AFF;">
            <text>头部范围</text>
            <text>头部范围</text>
        </view>
    </template>
    <template slot="body">
        <view style="background-color:#18BC37;">
            <text>身体范围</text>
            <text>身体范围</text>
        </view>
    </template>
    <template slot="footer">
        <view style="background-color:#DD524D;">
            <text>脚部范围</text>
            <text>脚部范围</text>
        </view>
    </template>

```

```

        </view>
    </template>
</uni-list-item>
</uni-list>
</view>
</template>

<script>
export default {
  data() {
    return {
      // bluetooth_text: '未开启',
      bluetooth_status: false
    };
  },
  methods: {
    bluetoothChange() {
      // 取反 特别开关操作 true 取反 false false 取反为 true
      this.bluetooth_status = !this.bluetooth_status;
      // if(this.bluetooth_status == true){
      //   this.bluetooth_status = false
      // }else{
      //   this.bluetooth_status = true
      // }
      // 前面表达式 为真 则第一项生效 为假 后面项生效
      // this.bluetooth_text = this.bluetooth_status ? '已开启' : '已关闭';
    }
  },
  // 其中数据 通过其他数据 计算得出 可以将其放在计算属性中 本身具有缓存 数据改变才更新 效率更高一些
  computed: {
    bluetooth_text() {
      return this.bluetooth_status ? '已开启' : '已关闭';
    }
  }
};
</script>

<style scoped></style>

```

## 案例:商品列表

请求接口获取商品列表数据,遍历显示数据,实现触底加载数据,下拉刷新数据,回到顶部按钮.

接口地址:

<http://www.codeboy.com:9999/data/product/list.php>

作业: 根据文档学习官格布局 ugrid 重点看如何布局双列

