

小程序实战开发

小程序实战开发

1. 课前准备
2. 回顾
3. 课堂主题
4. 课堂目标
5. 知识点
 - 通用小程序能力
 - 自定义taro主题
 - tabbar
 - 网络请求
 - 轮播图
 - 图片mode
 - rpx
 - 商品列表
 - flex布局
 - 下拉刷新
 - 触底加载更多
 - 条件编译解决H5不支持的问题
 - 无更多数据
 - 购物车
 - 购物车页面
 - 持久化
 - 转发
 - 模板消息
 - 登录
 - 支付
 - 分包
 - 小程序MINA框架
 - 小程序运行机制
 - 小游戏
6. 扩展点
7. 总结
8. 作业 && 答疑
9. 下节课预告

1. 课前准备

无

2. 回顾

1. taro使用s
2. 多端框架意义

3. 课堂主题

1. 实战 小程序全栈

4. 课堂目标

1. 实战小程序
2. taro条件编译

5. 知识点

通用小程序能力

有些能力是小程序端自带的，比如扫码，微信登录openid，云开发，微信支付这种，别的端很难模拟，比如H5上扫码就不太现实，所以可以用一些条件判断，小程序端使用原生的能力，不支持的端，可以调web开发接口

自定义taro主题

官方组件主题选择器 <https://nervjs.github.io/taro-ui-theme-preview/>

拷贝custom-theme.scss到app.scss上面

```
/* Custom Theme */
$color-brand: #e93b3d;
$color-brand-light: #ef6c6e;
$color-brand-dark: #ba2f31;

@import "~taro-ui/dist/style/index.scss"; // 引入组件样式，仅需引入一次即可
```

tabbar

```
// 新建三个页面 home,user,card
import Taro, { Component } from '@tarojs/taro'
import { View, Text, Button, Input } from '@tarojs/components'

export default class Index extends Component {

  constructor(props) {
    super(props)
    this.state = {
    }
  }
  config = {
    navigationBarTitleText: 'Home'
  }
  render() {
    return (
      <View>Home</View>
    )
  }
}
```

```
// src/app.js
config = {
  pages: [
    'pages/home/home',
    'pages/cart/cart',
    'pages/user/user',
  ],
  window: {
    navigationBarTitleText: '开课吧',
  },
  tabBar: {
    selectedColor: "#b4282d",
    list: [{
      pagePath: "pages/home/home",
      iconPath: "./assets/home.png",
      selectedIconPath: "./assets/home-active.png",
      text: "首页"
    },
    {
      pagePath: "pages/cart/cart",
```

```

    iconPath: "./assets/cart.png",
    selectedIconPath: "./assets/cart-active.png",
    text: "购物车"
  },

  {
    pagePath: "pages/user/user",
    iconPath: "./assets/user.png",
    selectedIconPath: "./assets/user-active.png",
    text: "个人"
  }
]
}
}

```

网络请求

Taro.request, h5端使用ajax, 小程序端使用wx.request 封装个函数,挂载再global

```

// app.js

global.url = 'http://localhost:3000/'
global.getData = (url)=>{
  return Taro.request({
    url: global.url+url,
  }).then(res=>{
    if(res.statusCode===200 && res.data.code===0){
      return res.data.data
    }else{
      Taro.showToast({
        title: '小老弟 出错了',
        duration: 2000
      })
    }
  })
}
}

```

后台使用express模拟

```

// 简单express

```

```

const express = require('express')
const fs = require('fs')
const path = require('path')
const app = express()
app.use(express.static('static'))
app.get('/', (req, res) => res.send('Hello world!'))
app.listen(3000, () => console.log('Example app listening on port 3000!'))

app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header('Access-Control-Allow-Methods', 'PUT, GET, POST, DELETE,
OPTIONS');
  res.header("Access-Control-Allow-Headers", "X-Requested-With");
  res.header('Access-Control-Allow-Headers', 'Content-Type');
  next();
});

```

```

const course =
JSON.parse(fs.readFileSync(path.resolve(__dirname, './goods.json')).toString())

const allData = []

course.tags.forEach(key=>{
  course.data[key].forEach(cor=>{
    allData.push(cor)
  })
})

app.get('/api/top', (req, res)=>{
  allData.sort((a,b)=>{
    return b.solded-a.solded
  })
  res.json({
    code:0,
    data:allData.slice(0,5)
  })
})

```

轮播图

<https://nervjs.github.io/taro/docs/components/viewContainer/swiper.html>

```
/pages/home/home.js
import { View, Text, Button, Input, Swiper, SwiperItem, Image } from
 '@tarojs/components'

componentDidMount() {
  // 获取轮播图

  global.getData('api/top').then(top => {
    this.setState({ top })
  })
}

<Swiper
  className='swiper-container'
  indicatorActiveColor='#e93b3d'
  interval={5000}
  circular
  indicatorDots
  autoplay>
  {this.state.top.map(t => {
    return <SwiperItem>
      <Image className='swiper-img' mode='aspectFit' src=
        `${global.url}course/${t.img}` ></Image>
      </SwiperItem>
    })}
</Swiper>
```

图片mode

Taro image <https://nervjs.github.io/taro/docs/components/media/image.html>

小程序 Image <https://developers.weixin.qq.com/miniprogram/dev/component/image.html>

Taro 设计稿尺寸描述 <https://nervjs.github.io/taro/docs/size.html>

rpx

rpx单位是微信小程序中css的尺寸单位，rpx可以根据屏幕宽度进行自适应。规定屏幕宽为750rpx。如在 iPhone6 上，屏幕宽度为375px，共有750个物理像素，则750rpx = 375px = 750物理像素，1rpx = 0.5px = 1物理像素。

<https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxss.html>

在 Taro 中尺寸单位建议使用 px 或 百分比 %，Taro 默认会对所有单位进行转换。在 Taro 中书写尺寸按照 1:1 的关系来进行书写，即从设计稿上量的长度 100px，那么尺寸书写就是 100px，当转成微信小程序的时候，尺寸将默认转换为 100rpx，当转成 H5 时将默认转换为以 rem 为单位的值。

```
<Swiper
  className='swiper-container'
  indicatorActiveColor='#b4282d'
  interval={5000}
  circular
  indicatorDots
  autoplay>

  {this.state.top.map(t=>{
    return <SwiperItem>
      <Image className='swiper-img' mode='aspectFit' src=
        {'`${global.url}course/${t.img}`'}></Image>

      </SwiperItem>

    })}

</Swiper>
```

```
.swiper-img{
  width:750px;
}

.swiper-container{
  height:480px;
}

.taro-img{
  text-align: center;
}

.detail{
  word-break: break-all;
}
```

商品列表

```
// server.js
app.get("/api/goods", (req, res) => {
  const page = req.query.page || 1

  const start = (page - 1) * 10
  const end = start + 10
  setTimeout(() => {
    res.json({
      code: 0,
      data: allData.slice(start, end)
    })
  }, 1000)
})
```

获取数据 简单的 先获取数据，显示

```
getGoods() {
  global.getData('api/goods?page=' + this.page).then(goods => {
    this.setState({ goods })
  })
}

componentDidMount() {
  this.getGoods()
}
```

```
import { AtCard, AtBadge, AtIcon } from 'taro-ui'

constructor(props) {
  super(props)
  this.page = 1
  this.state = {

    top: [],
    goods: []
  }
}
```



```

<view style="margin-top:10px">
  {this.state.goods.map(item => {
    const thumb = item.solded > 200 ? global.url + 'fire.png' : ''
    return <AtCard
      title={item.name}
      note="课程简介简介简介课程简介简介简介课程简介简介简介课程简介简介简介"
      extra={`¥${item.price}`}
      thumb={thumb}
    >

    <view className='at-row'>
      <view className='at-col at-col-4'>
        <Image mode='aspectFit' className="card-img" src=
        `${global.url}course/${item.img}` ></Image>

      </view>
      <view className='at-col at-col-7'>

        <view>已有{item.solded}人购买</view>
        <view>
          <AtBadge value={item.tag} />

        </view>

      </view>
      <view className='at-col at-col-1'>

        <AtIcon onClick={() => this.addCart(item)} value="add"
        color="#e93b3d" size="30"></AtIcon>
      </view>
    </view>
  </AtCard>
  )}
</view>

```

```
// app.scss
```

```

.card-img{
  width:100%;
  height:100%;
  min-height: 100px;
}

```

flex布局

小程序和rn都力推flex布局,

http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html?utm_source=tuicool

下拉刷新

H5暂时没兼容，小程序端自带

```
// pages/home/home.js
config = {
  enablePullDownRefresh: true
}
constructor(props) {
  super(props)
  this.page = 1
  this.state = {

    top: [],
    goods: [],
    hasMore: false
  }
}
onPullDownRefresh(){
  this.getGoods()
}
```

触底加载更多

小程序自带，H5需要hack一下，比如点击加载

1. 小程序onReachBottom 生命周期

```
loadMore(){
  this.page = this.page+1
  this.getGoods(true)
}
```

```

getGoods(append) {

  global.getData('api/goods?page=' + this.page).then(goods => {
    if(append){
      // 追加模式, 分页
      this.setState({
        goods:[...this.state.goods, ...goods]
      })
    }else{
      // 初始化
      this.setState({ goods })
    }
    this.setState({
      hasMore:goods.length===10
    })
  })
}

onReachBottom(){
  console.log('')

  if(!this.state.hasMore){
    // 数据到底了 不需要加载更多了
    return
  }

  this.loadMore()
}

```

2. H5端设置一个点击加载 使用条件编译

条件编译解决H5不支持的问题

<https://nervjs.github.io/taro/docs/envs.html>

```
import { AtDivider, AtNavBar, AtCard, AtBadge, AtIcon, AtLoadMore } from
'taro-ui'

this.state = {
  isH5: process.env.TARO_ENV === 'h5',
  top: [],
  goods: []
}

{this.state.isH5 ? <AtLoadMore onClick={() => this.loadMore()} />: null}
```

无更多数据

1. 前端判断返回的length 如果小于10, 就判定为没有更多数据

```
getGoods(append) {
  global.getData('api/goods?page=' + this.page).then(goods => {
    if(append){
      // 追加模式, 分页
      this.setState({
        goods:[...this.state.goods, ...goods]
      })
    }else{
      // 初始化
      this.page = 1
      this.setState({ goods })
    }
    this.setState({
      hasMore:goods.length===10
    })
  })
}

{
  this.state.hasMore ?<View>
    {this.state.isH5 ?<AtLoadMore onClick={() => this.loadMore()}
/>>:null}
    </View> : <AtDivider content='没有更多了' color="#e93b3d"></AtDivider>
}
```

2. server端返回hasMore 更可信

购物车

多页面共享数据，mobx登场

```
// src/store/cart.js
import Taro from '@tarojs/taro'
import { observable } from 'mobx'

const cartStore = observable({
  carts: [],

  addCart(item) {

    // const
    const goodInCart = this.carts.findIndex(v=>v.id===item.id)
    if(goodInCart>=-1){
      // 购物车里有了
      this.carts[goodInCart].count+=1
    }else{
      item.count = 1
      this.carts.push(item)
    }

    const len = this.carts.reduce((sum,a)=>sum+a.count,0)

    Taro.setTabBarBadge({
      index: 1,
      text: len+' '
    })
    Taro.showToast({
      title: '添加成功',
      duration:2000
    })
  },

})

export default cartStore
```

```
// page/home/home.js
```

```
import { observer, inject } from '@tarojs/mobx'
@inject('cartStore')
@observer
export default class Home extends Component {

  constructor(props) {
    super(props)
  }

  addCart = (item) => {
    this.props.cartStore.addCart(item)
  }

  <view className='at-col at-col-1'>
    <AtIcon onClick={() => this.addCart(item)} value="add"
color="#e93b3d" size="30"></AtIcon>
  </view>
```

购物车页面

```
import Taro, { Component } from '@tarojs/taro'
import { View, Text, Input, Image } from '@tarojs/components'

import { AtDivider, AtNavBar, AtCard, AtBadge, AtIcon, AtLoadMore, AtButton } from
'taro-ui'

import { observer, inject } from '@tarojs/mobx'

@inject('cartStore')
@observer
export default class Index extends Component {

  config = {
    navigationBarTitleText: '购物车'
  }
  constructor(props) {
    super(props)
    this.state = {
```

```

    // carts:this.props.cartStore.carts
  }
}
cartSub = (index)=>{
  this.props.cartStore.cartSub(index)
}
cartAdd = (index)=>{
  this.props.cartStore.cartAdd(index)
}
render () {
  const {cartStore} = this.props

  return (
    <View className='index'>

{cartStore.carts.length ?

<View style="margin:10px 0;">
{cartStore.carts.map((item,index) => {
  const thumb = item.solded > 200 ? global.url + 'fire.png' : ''
  return <AtCard
    title={item.name}
    note="课程简介简介简介课程简介简介简介课程简介简介简介课程简介简介简介"
    extra={`¥${item.price}`}
    thumb={thumb}
  >

    <View className='at-row'>
      <View className='at-col at-col-4'>
        <Image mode='aspectFit' className="card-img" src=
{`${global.url}course/${item.img}`} ></Image>

        </View>
        <View className='at-col at-col-8'>
          <AtIcon onClick={()=>this.cartSub(index)} value="subtract"
color="#e93b3d" size="30"></AtIcon>
          <Text>{item.count}</Text>
          <AtIcon onClick={()=>this.cartAdd(index)} value="add" color="#e93b3d"
size="30"></AtIcon>
        </View>
      </View>

    </AtCard>
  )}
<View style="margin:10px;" >
  <AtButton type='primary'> ¥{cartStore.totalPrice}下单</AtButton>
</View>

```

```

</View>

:<AtDivider content='购物车空空如也' color="#e93b3d"></AtDivider>}

    </View>
  )
}
}

```

```

// store/cart.js

totalPrice:0,
cartSub(i){
  // console.log(this.carts)
  // const newCart =
  // console.log(newCart)
  // newCart.count -=1
  const newCount = this.carts[i].count-1

  if(newCount<1){
    this.carts.splice(i,1)
  }else{
    this.carts[i]= {...this.carts[i], count:newCount}
  }
},
cartAdd(i){
  const newCount = this.carts[i].count+1
  this.carts[i]= {...this.carts[i], count:newCount}
},

```

持久化

使用setStorageSync和getStorageSync


```
import { observable ,computed,autorun} from 'mobx'

carts: Taro.getStorageSync('cartData') || [],

const totalCount =
computed(()=>cartStore.carts.reduce((sum,a)=>sum+a.count,0))
const totalPrice =
computed(()=>cartStore.carts.reduce((sum,a)=>sum+a.count*a.price,0))
autorun(()=>{

  Taro.setTabBarBadge({
    index: 1,
    text: totalCount.get()+''
  })
  Taro.setStorageSync('cartData',cartStore.carts.toJS())

  cartStore.totalPrice = totalPrice.get()
})
```

转发

<https://developers.weixin.qq.com/miniprogram/dev/api/wx.showShareMenu.html>

模板消息

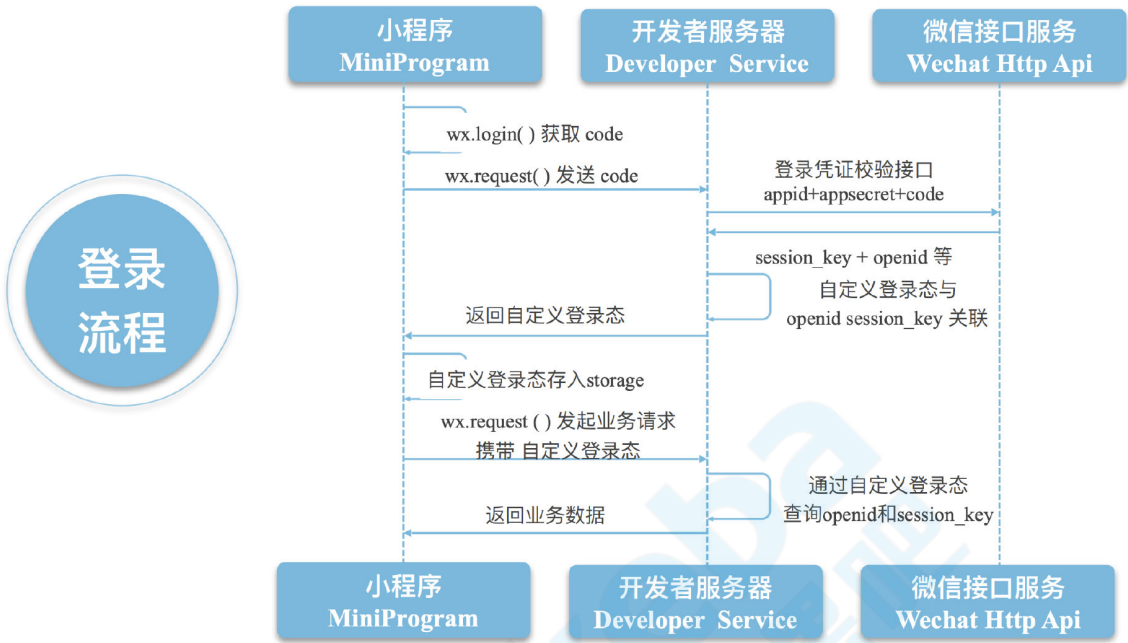
必须要表单提交，或者支付后，才能给用户发一次 并且时效性是7天以内

1. 通过模板消息管理接口获取模板 ID
2. form组件设置 `report-submit` 后，返回的formId 或者支付后，返回的 `prepay_id`
3. 调用sendTemplateMessage接口发送消息

<https://developers.weixin.qq.com/miniprogram/dev/api/sendTemplateMessage.html>

登录

云开发登录贼简单，实际开发确需要费点劲



支付

32位密钥，下载证书，对https是强需求 需要企业账号才能开通

首页

管理

版本管理

成员管理

反馈管理

统计

功能

附近的小程序

微信支付

物流助手

客服

模板消息

小程序插件

开发

推广

流量主

微信支付

微信支付 未申请

开通

介绍

微信支付支持已认证的服务号，政府或媒体订阅号接入支付功能。公众号接入支付后，可以通过JSAPI支付产品来完成在公众号、朋友圈、聊天窗口等微信内的收款需求。[申请指引](#) | [开发文档](#)

申请材料

企业 个体工商户 政府及事业单位、民办非企业、社会团体、基金会

1、营业执照：彩色扫描件或数码照片

2、组织机构代码证：彩色扫描件或数码照片，若已三证合一，则无需提供

3、对公银行账户：包含开户行省市信息，开户账号

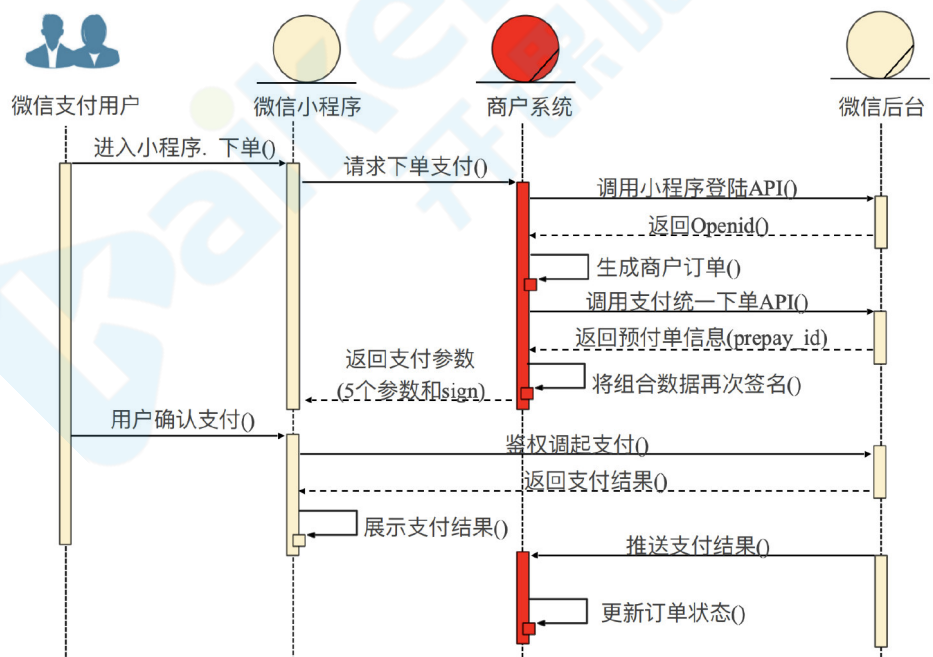
4、法人身份证：彩色扫描件或数码照片

申请流程

1 提交资料

2 签署协议

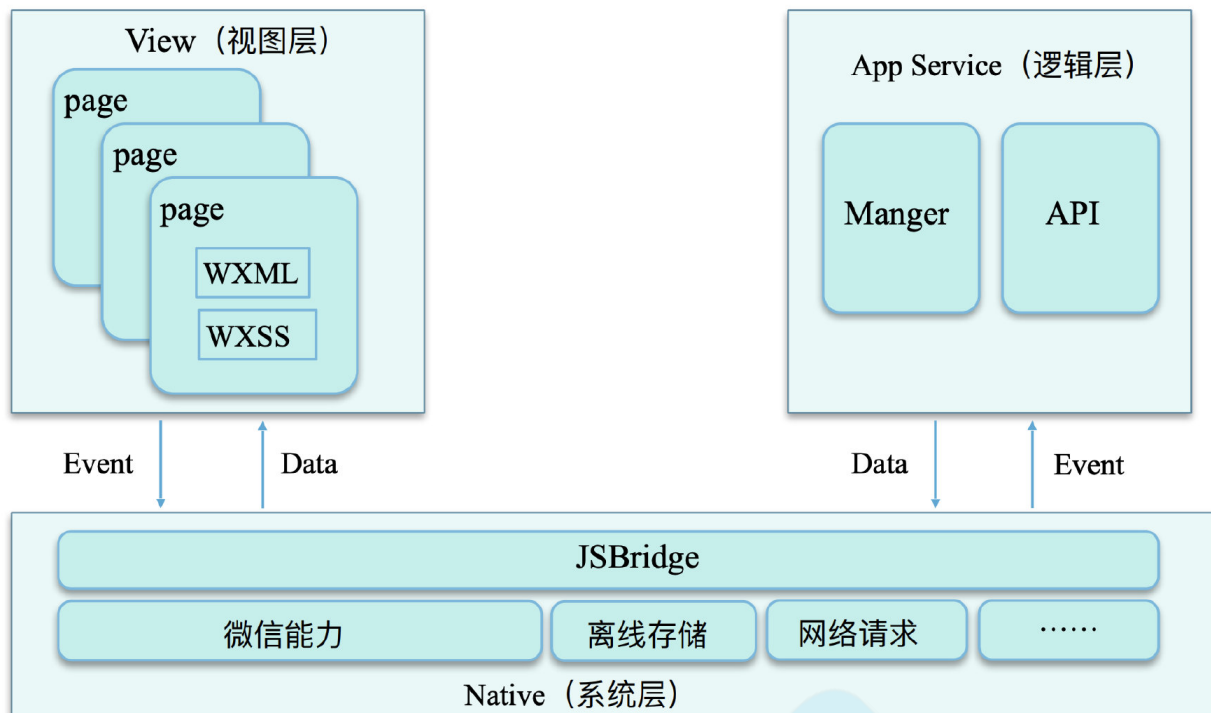
支付交互图



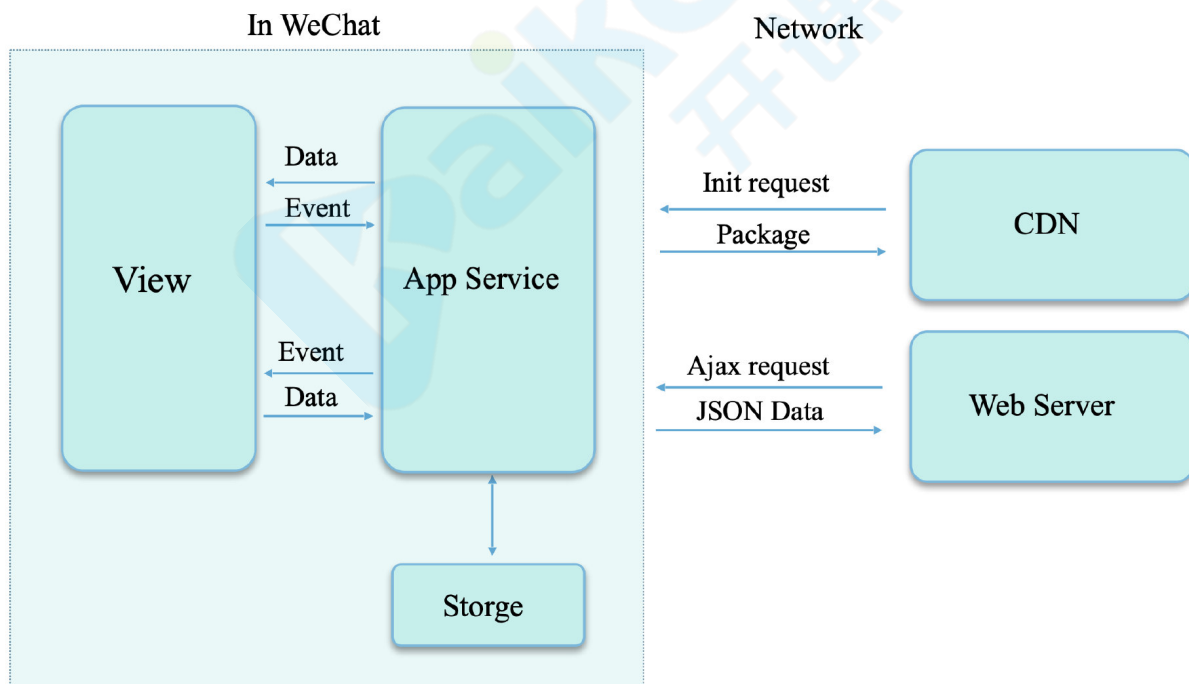
分包

<https://developers.weixin.qq.com/miniprogram/dev/framework/subpackages.html>

小程序MINA框架

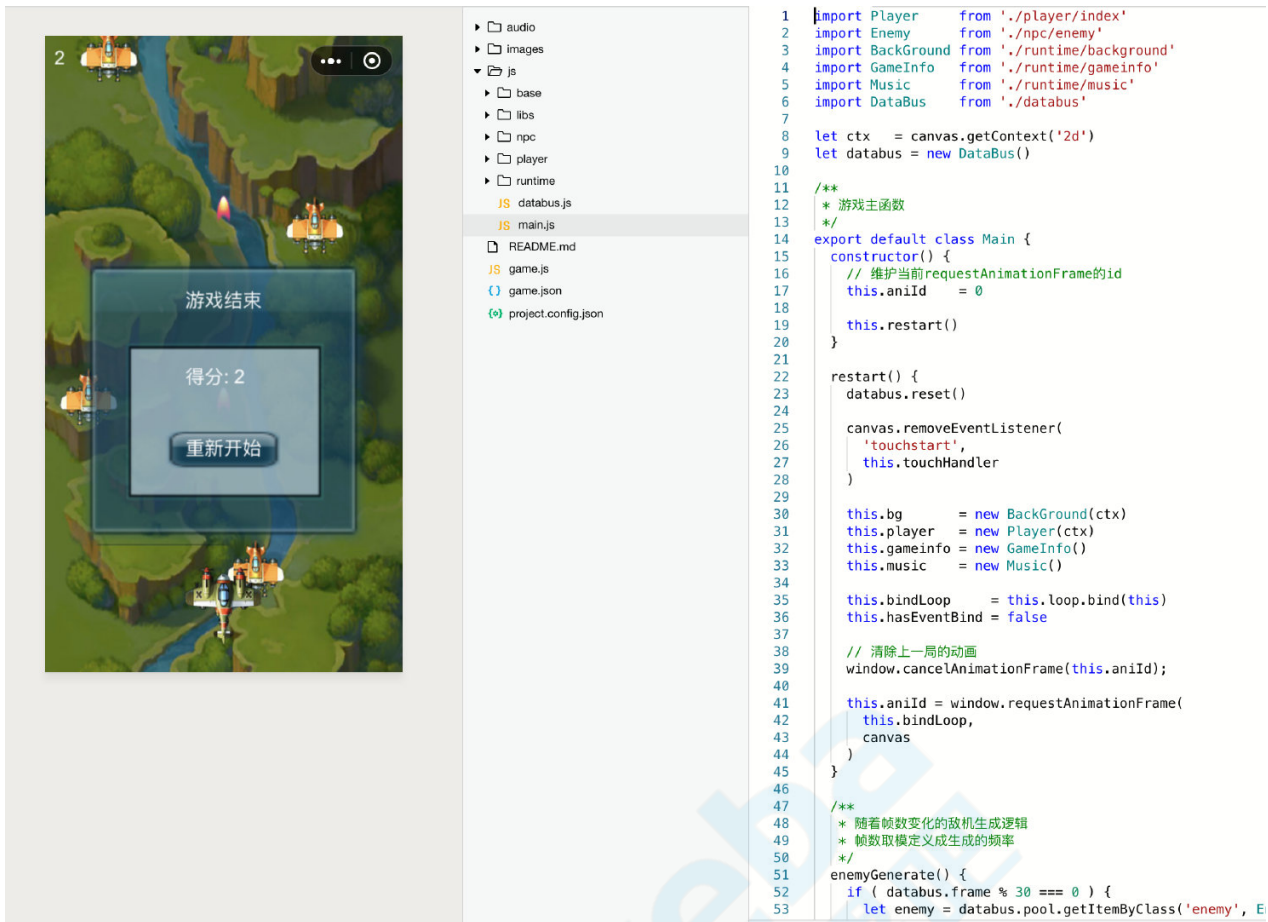


小程序运行机制



小游戏

并不是课堂内容，可以体验下demo



6. 扩展点

1. 思考多端融合
2. Uni-app 使用vuejs做一样的事

7. 总结

1. 回顾知识点
2. 提示学习方法
3. 提示本次课重点和必会知识

小程序实战开发

1. 课前准备
2. 回顾
3. 课堂主题
4. 课堂目标
5. 知识点

通用小程序能力

自定义taro主题

tabbar

网络请求

轮播图

图片mode
rpx
商品列表
flex布局
下拉刷新
触底加载更多
条件编译解决H5不支持的问题
无更多数据
购物车
购物车页面
持久化
转发
模板消息
登录
支付
分包
小程序MINA框架
小程序运行机制
小游戏

- 6. 扩展点
- 7. 总结
- 8. 作业 && 答疑
- 9. 下节课预告

8. 作业 && 答疑

9. 下节课预告

夏老师 微信公众号开发

<https://github.com/tencentyun/wafer>