

小程序03

taro框架入门

小程序高频需求实战

1、Taro跨端小程序

- Taro环境安装，新建Taro项目
- Taro多端编译
- 微信小程序转Taro
- Taro 3

2、小程序常见需求实战

- 朋友圈分享海报生成
- 订阅消息
- 微信支付
- 自定义导航
- 激励视频广告组件
- 微信小程序反编译

关于Taro

京东-凹凸实验室

Taro 是一套遵循 [React](#) 语法规范的 **多端开发** 解决方案。

现如今市面上端的形态多种多样，Web、React-Native、微信小程序等各种端大行其道，当业务要求同时在不同的端都要求有所表现的时候，针对不同的端去编写多套代码的成本显然非常高，这时候只编写一套代码就能够适配到多端的能力就显得极为需要。

使用 **Taro**，我们可以只书写一套代码，再通过 **Taro** 的编译工具，将源代码分别编译出可以在不同端（微信/百度/支付宝/字节跳动/QQ/京东小程序、快应用、H5、React-Native 等）运行的代码。

快速开始

快速开始文档: <https://taro-docs.jd.com/taro/docs/GETTING-STARTED.html>

node 环境 ($\geq 8.0.0$)

CLI 工具安装

首先, 你需要使用 npm 或者 yarn 全局安装 @tarojs/cli, 或者直接使用[npx](#):

```
# 使用 npm 安装 CLI
$ npm install -g @tarojs/cli
# OR 使用 yarn 安装 CLI
$ yarn global add @tarojs/cli
# OR 安装了 cnpm, 使用 cnpm 安装 CLI
$ cnpm install -g @tarojs/cli
```

###

项目初始化

使用命令创建模板项目

```
$ taro init myApp
```

项目目录结构

├─ dist	编译结果目录
├─ config	配置目录
│ └─ dev.js	开发时配置
│ └─ index.js	默认配置
│ └─ prod.js	打包时配置
├─ src	源码目录
│ └─ pages	页面文件目录
│ └─ index	index 页面目录
│ └─ index.js	index 页面逻辑
│ └─ index.css	index 页面样式
│ └─ app.css	项目总通用样式
│ └─ app.js	项目入口文件
└─ package.json	

入口文件

入口文件默认是 `src` 目录下的 `app.js`。

项目组织

以下文件组织规范为最佳实践的建议

所有项目源代码请放在项目根目录 `src` 目录下, 项目所需最基本的文件包括 **入口文件** 以及 **页面文件**

- 入口文件为 `app.js`
- 页面文件建议放置在 `src/pages` 目录下

一个可靠的 Taro 项目可以按照如下方式进行组织

```

├─ config                      配置目录
│  ├─ dev.js                  开发时配置
│  ├─ index.js                默认配置
│  └─ prod.js                 打包时配置
├─ src                        源码目录
│  ├─ components              公共组件目录
│  ├─ pages                    页面文件目录
│  │  └─ index                index 页面目录
│  │     └─ banner            页面 index 私有组件
│  │     └─ index.js          index 页面逻辑
│  │     └─ index.css         index 页面样式
│  └─ utils                    公共方法库
├─ app.css                    项目总通用样式
├─ app.js                     项目入口文件
└─ package.json

```

微信小程序转taro

<https://taro-docs.jd.com/taro/docs/taroize.html>

微信原生小程序转 Taro 的操作非常简单，首先必须安装使用 `npm i -g @tarojs/cli` 安装 Taro 命令行工具，其次在命令行中定位到小程序项目的根目录，根目录中运行：

小程序的目录，不是云函数的

```
$ taro convert
```

即可完成转换。转换后的代码保存在根目录下的 `taroConvert` 文件夹下。你需要定位到 `taroConvert` 目录执行 `npm install` 命令之后就可以使用 `taro build` 命令编译到对应平台的代码。

Taro3开放式跨端跨框架解决方案

跨框架：React、Nerv、Vue 2、Vue 3、jQuery

跨端

开放式插件系统

上车 Taro 最好的时机是从前，其次是现在。

安装Taro 3

```
# 使用 npm 安装 CLI
$ npm install -g @tarojs/cli
# OR 使用 yarn 安装 CLI
$ yarn global add @tarojs/cli
# OR 安装了 cnpm, 使用 cnpm 安装 CLI
$ cnpm install -g @tarojs/cli
```

```
taro init myApp
```

环境的安装

编译，多端调试

预览

发布 build

微信小程序开发高频需求

众奖: wx8bda751c6e3e9174

小花盆: wx13036a11656af72a

要分享到朋友圈，就肯定要用到小程序码，微信提供了三种小程序码接口

<https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/qr-code.html>

 image-20200802081838784

接口A，可以接受 pages/index/index?param=128byte （10万次）

?userid=xxx&from=123213

开课吧web全栈架构师

接口C，官方已经不推荐使用了，方形不建议使用了。

接口B，必须是已经发布的小程序存在的页面（否则报错），例如 `pages/index/index`，根路径前不要填加 `/`，不能携带参数（参数请放在scene字段里），如果不填写这个字段，默认跳主页面

<https://developers.weixin.qq.com/miniprogram/dev/api-backend/open-api/qr-code/wxacode.getUnlimited.html>

```
try {
  const result = await cloud.openapi.wxacode.getUnlimited({
    scene: event.scene,
    page: event.page
  })
  return result
} catch (err) {
  return err
}
```

```
const query = wx.createSelectorQuery()
query.select('#myCanvas')
  .fields({ node: true, size: true })
  .exec((res) => {
    const canvas = res[0].node
    const ctx = canvas.getContext('2d')

    const dpr = wx.getSystemInfoSync().pixelRatio
    canvas.width = res[0].width * dpr
    canvas.height = res[0].height * dpr
    ctx.scale(dpr, dpr)
    ctx.fillStyle = 'white'
    ctx.fillRect(0, 0, 300, 300)

    wx.cloud.callFunction({
      name: "getmpcode",
      data: {
        scene: "a=1"
      },
      success: (res) => {
        console.log(res);

        let image = canvas.createImage();
        image.src = "data:image/jpeg;base64," +
        wx.arrayBufferToBase64(res.result.buffer);

        image.onload = (res) => {
          console.log(res);

          ctx.drawImage(image, 0, 0, 100, 100);
        }
      }
    });
  })
```

```
//画布转本地图片路径
wx.canvasToTempFilePath({
  canvas: canvas,
  success: (res) => {
    console.log(res);
  }
})
```

之后，接wx.previewImage或者wx.saveImageToPhotosAlbum

调试扫码进入

订阅消息

开发者工具不支持

```
wx.requestSubscribeMessage({
  tmplIds: ["9CrIcgm_VRaS1qvboYxUX_P6N8lgMlwBOjpwQ6y_zzs"],
  success(res){
    console.log(res)
  },
  fail(err){
    console.log(err)
  }
})
```

<https://developers.weixin.qq.com/miniprogram/dev/api-backend/open-api/subscribe-message/subscribeMessage.send.html>

```
const wxContext = cloud.getWXContext()

try {
  const result = await cloud.openapi.subscribeMessage.send({
    touser: wxContext.OPENID,
    page: 'index',
    lang: 'zh_CN',
    data: {
      date1: {
        value: '2019年10月1日'
      },
      thing2: {
        value: "大家好"
      }
    },
    templateId: '9CrIcgm_VRaS1qvboYxUX_P6N8lgMlwBOjpwQ6y_zzs',
    miniprogramState: 'developer'
  })
  return result
} catch (err) {
  return err
}
```

微信支付

个人号不可以申请

```
wx.requestPayment
```

```
const res = await cloud.cloudPay.unifiedOrder({
  "body" : "大帅-超市5",
  "outTradeNo" : "12321321134",
  "spbillcreateIp" : "127.0.0.1",
  "subMchId" : "1236636002",
  "totalFee" : 100,
  "envId": "zhongjiang-3z81s",
  "functionName": "pay_cb"
})
return res
```

激励视频广告

微信小程序反编译

使用网易mumu模拟器

安装微信，RE文件管理器，并开启root权限

打开小程序，data/data/com.tencent.com/MicroMsg/xxxxxx/appbrand/pkg

复制到mumu共享文件夹

反编译工具仓库

<https://gitee.com/ksd/wxappUnpacker>

cd wxappUnpacker

npm install

node wuWxapkg.js xxxxxx.wxapkg