

小程序-02

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。

把转换后的代码目录，粘贴到taro小程序的src目录

进入taro小程序目录的入口 app.js/app.jsx，添加目录页面

```
class App extends Component {
  config = {
    pages: [
      "pages/booksList/booksList",
      "pages/book/book",
      "pages/index/index"
    ],
    ...
    cloud: true
  };
}
```

里面会含有微信小程序和taro的混合语法，需要安装

```
npm i @tarojs/with-weapp
```

就可以在taro里支持微信小程序的语法了。

taro-UI

开课吧web全栈架构师

依赖taro脚手架

进入项目小程序目录执行

```
npm install taro-ui
```

使用 Taro UI

```
// page.js
import { AtButton } from 'taro-ui'
// 除了引入所需的组件，还需要手动引入组件样式
// app.js
import 'taro-ui/dist/style/index.scss' // 全局引入一次即可
```

美化图书列表页

```
import { AtCard } from "taro-ui";
class _C extends Taro.Component {
  config = {
    onReachBottomDistance: 100
  };

  render() {
    const { books } = this.data;
    return (
      <Block>
        <Text>mp/pages/booksList/booksList.wxml</Text>
        {books.map((item, index) => {
          return (
            <AtCard extra={index} title={item.title} thumb={item.image}>
              {item.summary}
            </AtCard>
          );
        })}
      </Block>
    );
  }
}
```

小程序支付

微信支付发起API文档：

<https://developers.weixin.qq.com/miniprogram/dev/api/open-api/payment/wx.requestPayment.html>

```
wx.requestPayment(Object object)
...
wx.requestPayment({
  timeStamp: '',
  nonceStr: '',
  package: '',
  signType: 'MD5',
  paySign: '',
  success (res) { },
  fail (res) { }
})
```

支付所需要用到的配置：

https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=3_1#

- APPID: appid
 - appid是微信小程序后台APP的唯一标识，在小程序后台申请小程序账号后，微信会自动分配对应的appid，用于标识该应用。可在小程序-->设置-->开发设置中查看。
- 微信支付商户号: mch_id
 - 商户申请微信支付后，由微信支付分配的商户收款账号
- API密钥: key
 - 交易过程生成签名的密钥，仅保留在商户系统和微信支付后台，不会在网络中传播。商户妥善保管该Key，切勿在网络中传输，不能在其他客户端中存储，保证key不会被泄漏。商户可根据邮件提示登录微信商户平台进行设置。也可按以下路径设置：微信商户平台 (pay.weixin.qq.com)-->账户设置-->API安全-->密钥设置
- Appsecret: secret
 - AppSecret是APPID对应的接口密码，用于获取接口调用凭证时使用。

字段解释：

- AppID（个人号不可以）
 - 申请小程序账号文档：https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=7_11&index=2
 - 小程序申请认证

档: <https://developers.weixin.qq.com/miniprogram/introduction/#%E5%B0%8F%E7%A8%B%E5%BA%8F%E7%BB%91%E5%AE%9A%E5%BE%AE%E4%BF%A1%E5%BC%80%E6%94%BE%E5%B9%B3%E5%8F%B0%E5%B8%90%E5%8F%B7>

- 商户号

- 商户平台申请地址: https://pay.weixin.qq.com/index.php/core/home/login?return_url=%2F

微信支付商户号, 获取商户号和商户密钥在微信支付商户管理平台中, 【账户中心】→【商户信息】下可以获取微信支付商户号。



在【账户中心】->【API安全】下可以设置商户密钥。



小程序支付发起流程:

https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=23_9&index=1#menu1.1

举个栗子🌰:

某用户小明在某电商小程序上购买一块肥皂，从浏览、下单到支付经历了什么样的过程呢？

1. 打开电商小程序，搜索浏览到某雕肥皂，点击查看详情后，发现大小、丝滑程度都很合适，点击直接下单
 1. 肥皂的价格：0.01元
2. wx.login获取用户临时登录凭证code，发送到后端服务器换取openId
3. 在下单时，小程序需要将小明购买的商品Id，商品数量，以及小明这个用户的openId传送到服务器
4. 服务器在接收到商品Id、商品数量、openId后，生成服务期订单数据，同时经过一定的签名算法，向微信支付发送请求，获取预付单信息(prepay_id)，同时将获取的数据再次进行相应规则的签名，向小程序端响应必要的信息(必须字段信息将在下文进行详细说明)
5. 小程序端在获取对应的参数后，调用wx.requestPayment()发起微信支付，唤醒支付工作台，进行支付

云开发支付

博客：

1.借助小程序云开发实现小程序支付功能

<https://www.jianshu.com/p/ddccf5f95e8c>

2.小程序支付003~借助云开发10行代码快速实现小程序支付

https://blog.csdn.net/qiushi_1990/article/details/106679387

代码发布

<https://developers.weixin.qq.com/miniprogram/introduction/#%E6%8F%90%E4%BA%A4%E5%AE%A1%E6%A0%B8>