



SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

滴滴出行在小程序的实践

嘉宾：张耀春

- 个人简介
- 基本知识
- 团队开发
- 开发者工具
- 编辑器开发
- 更多交流

个人简介

人称：小春

目前工作：主要负责滴滴出行平台公共前端团队

学习领域：前端、客户端、后端技术研发，喜欢学习总结

自我修炼：
最近带领团队出版了《Vue.js 权威指南》
最早翻译过《Professional Node.js》
在 Iteye 写了 5 年多博客，目前维护多个技术
公众号，如 DDFE - 滴滴前端、DDFE Weekly 技
术周刊



Didi
Front-End

似曾相识

最早: 在豌豆荚做 PC 端百宝袋应用接入

模式: 按照规则开发, 再在平台上传打包, 审核, 客户端拉取

之前: 应用 RN 开发一些 App

模式: 按照规则开发, 自己打包

现在: 小程序开发

模式: 提供开发者工具和新规则, 工具打包

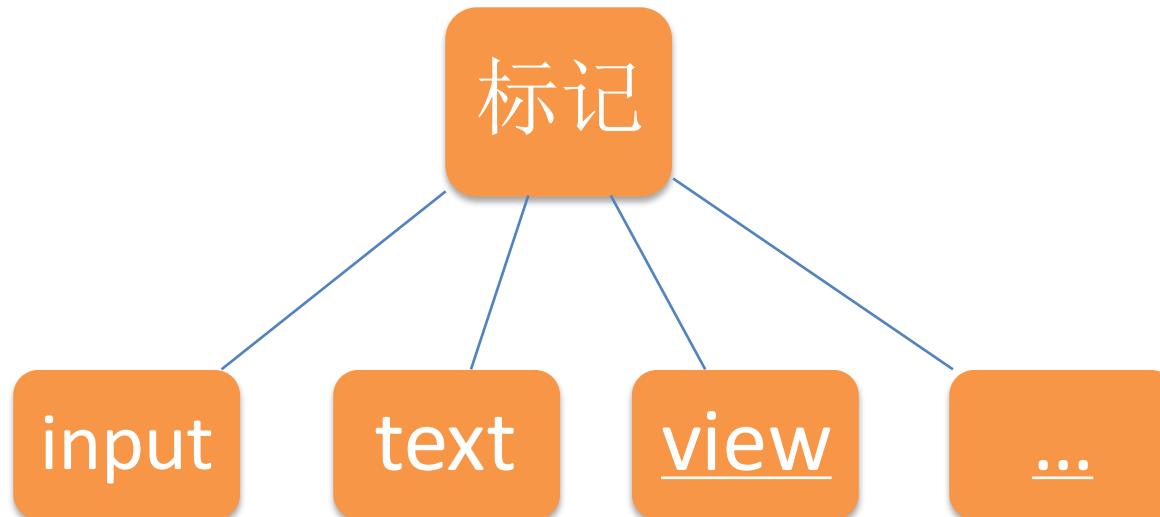
- 个人简介
- 基本知识
- 团队开发
- 开发者工具
- 编辑器开发
- 更多交流

视图 - 标记

新语言？： WXML – WeiXin Markup Language

使用： 和写 HTML 差不多

优势： 数据单向绑定、 循环、 条件判断



视图 - 样式

新语言? : WXSS – WeiXin Style Sheets

新名词: rpx



特殊元素

组件? : block 并不是一个组件, 一个包装元素, 不渲染

代码片段:

```
<block wx:for="{{cityList}}>
  <view class="didi-city-list">
    <text>{{item.name}}</text>
  </view>
</block>
```

事件

写法：以 bind 和 catch 开头，再加上事件类型

常用：bindtap、bindinput、binderror

实践技巧：e.target.dataset.id → data-id

事件对象示例：



```
▼ currentTarget: Object
  ► dataset: Object
    id: ""
    offsetLeft: 123
    offsetTop: 57
  ► __proto__: Object
▼ detail: Object
  cursor: 6
  value: "136411"
  ► __proto__: Object
▼ target: Object
  ► dataset: Object
    id: ""
    offsetLeft: 123
    offsetTop: 57
  ► __proto__: Object
  timestamp: 1479378620557
  ► touches: Array[0]
  type: "input"
```

指令 - 循环

循环: wx:for VS wx-for

默认: 序号 index , 当前变量 item, 可以自定义

应用标签: 一般 view 和 block

代码片段:

```
<view  
    wx:key="unique"  
    wx:for="{{array}}>  
    {{item.message}}  
</view>
```

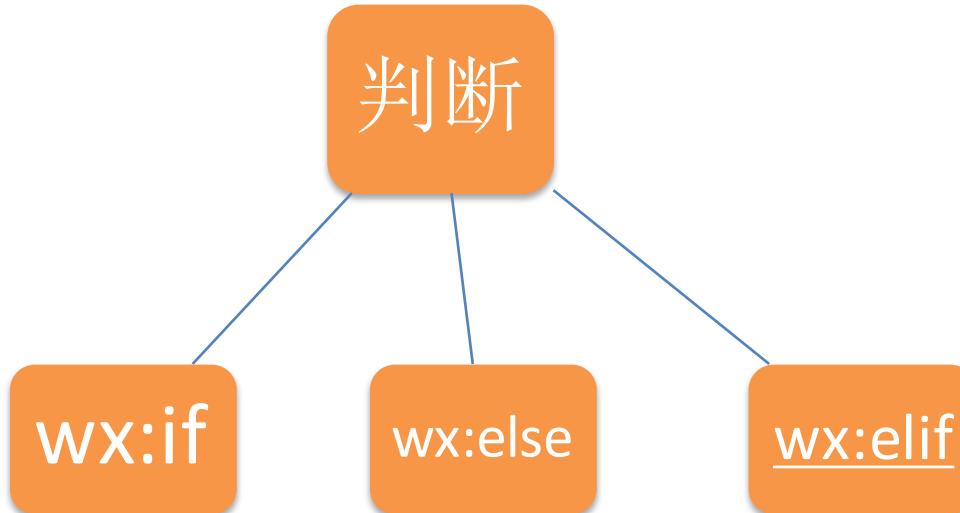
```
data:{  
    array: [{  
        message: '整洁'  
    }, {  
        message: '活地图'  
    }]  
}
```

指令 - 判断

API: `wx:if` VS `wx-else` `wx:elif`

应用场景: 一个模板里面通过变量区分展示快

代码片段:



页面

存放: pages 文件夹，一个页面一个目录

配置: app.json 里面的 pages 数组，通过删减项来增减页面

初始页: pages 数组的第一项

目录结构:

- pages
 - auth
 - home
 - driveInfo
 - rule
 - location

- pages
 - auth
 - auth.js
 - auth.json
 - auth.wxml
 - auth.wxss

生命周期

API 认知：多了一对前后台切换的

使用场景：比如在登录页面



页面切换 - back

使用场景：从 A 跳到 B，在 B 里面点取消



`wx.navigateBack()`

问题来了：参数能否能传递

会触发回退后的页面的生命周期哪个钩子函数

问题来了：onLoad 还是 onShow

会触发回退后的页面的生命周期哪个钩子函数

页面切换 - to

使用场景：从 A 跳到 B，在 B 里面点取消



```
wx.navigateTo({  
  ...  
})
```

问题来了：参数能否传递

会触发回退后的页面的生命周期哪个钩子函数

问题来了：onLoad 还是 onShow

会触发回退后的页面的生命周期哪个钩子函数

页面切换 - 参数

实例后感知：

- 1、wx.navigateTo 可以回传参数，同时触发 onLoad 和 onShow，但只在 onLoad 有参数传递
- 2、wx.navigateBack 不可以传参数，只触发 onShow

```
28 },
29   onLoad: function(options) { options = Object {from: "xiaochun"}
30     var that = this
31     quetyData = {
32       ...
```



服务端通讯

有 APPID：需要配置 https 域名

配置服务器信息

服务器域名需经过ICP备案，新备案域名需24小时后才可配置。域名格式只支持英文大小写字母、数字及“-”，不支持IP地址及端口号。

request合法域名

https://



https://



socket合法域名

wss://



uploadFile合法域名

https://



downloadFile合法域名

https://



目前我们基本使用 request 域名，一个月申请 3 次修改

服务端通讯

API: wx.request

感知: 和 ajax 很类似

代码片段:

```
wx.request({  
    url: "", data: {},  
    header: {  
        'content-type': 'application/json'  
    }  
})
```

服务端通讯 - header

如何自定义请求头

Transfer-Encoding: chunked

▼ Request Headers [view source](#)

```
Accept: */*
Accept-Encoding: gzip, deflate, sdch, br
Connection: keep-alive
Content-Type: application/json
Host: \[REDACTED\]
Referer: https://servicewechat.com/touristappid/devtools/page-frame.html
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36 appservice webview/100000
X-Requested-With: XMLHttpRequest
```

```
header: {
  'Content-Type': 'application/x-www-form-urlencoded'
}
```

服务端通讯 - jsonp

如何处理 JSONP ?

- 1、看一下请求地址: https://*****?****&callback=jsonp
- 2、响应结果: jsonp({})

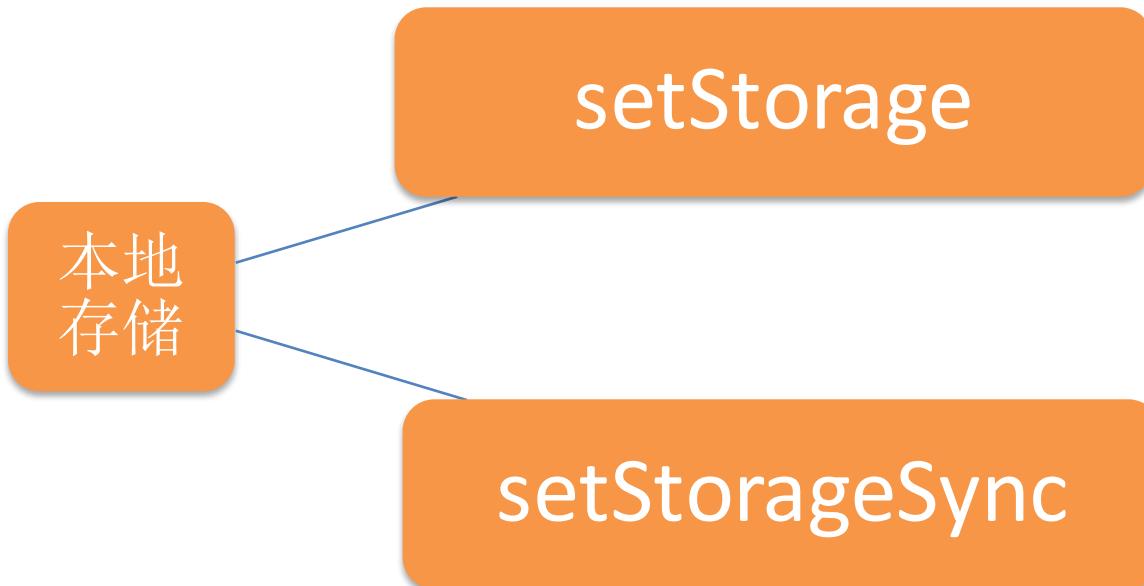
```
data: {  
    callback: 'jsonp'  
},  
complete: function() {  
}
```

```
> res  
< "jsonp({\"errno\":0,\"district\":\"010\",\"ready_time\":5,  
,\"can_create_order\":1})"  
> res.slice(0, res.length-1)  
< "jsonp({\"errno\":0,\"district\":\"010\",\"ready_time\":5,  
,\"can_create_order\":1})"  
> res.slice(6)  
< "  
  {"errno":0,"district":"010","ready_time":5,"arrive  
  reate_order":1})"  
> |
```

本地存储

API：有同步接口，最大 10 MB，其他和 H5 localStorage 类似

使用场景：目前我们一般会把一些信息存储在里面，跨页面获取



位置获取

API 感知：目前使用默认 type wgs84

使用场景：获取当前坐标，发请求找到附件车辆

```
wx.getLocation({  
    type: 'wgs84',  
    success: function(res) {  
        that.setData({  
            lat: res.latitude,  
            lng: res.longitude  
        })  
    }  
})
```

拨打电话

API 感知：是方法，而不是之前的那种 tel 协议

使用场景：客服入口以及司机卡片拨号图标

```
wx.makePhoneCall({  
  phoneNumber: '*'  
})
```

组件

Swiper : 轮播组件，主要用于我们的静态计价规则
swiper-item

Toast : `wx.showToast` 目前 icon 只支持 success 和 loading
`wx.hideToast`

Toast : `wx.showModal`

scroll-view: 滚动

表单 : 常规的基本都用了

动画

需求：实现一个等待应答的转圈

一个圆点让着一个圆圈线滚动

```
<view class='didi-crl-dot'  
animation='{{animationData}}'>  
</view>
```

```
wx.createAnimation
```

```
setInterval
```

```
this.setData({  
    animationData: animation.export()  
})
```



ES6

支持版本： >= 0.10.101000

关于Polyfill：只处理语法转换，不处理 Polyfill 包

如何开启：最开始导入项目的时候，打钩

代码片段：

```
<view class="container">
  <template is="driverInfo" data="{{...driverInfo}}"/>
</view>
```

- 个人简介
- 基本知识
- 团队开发
- 开发者工具
- 编辑器开发
- 更多交流

早期

无 AppID 那会

团队众包：

团队每一个同学一个内容章节，然后进行串讲，手把手

常见问题：

- `wx.login` 受限
- `wx.request` 开发者工具不检查安全域名
- 地图、支付功能

起步

UI 层面：设计师按照 iphone6 标准

IDE 层面：其实推荐在开发者工具里面开发，实践后的理由：

- 目前我们基本会把 `wxml` 面板移到第一个 tab
- 提供自动补全，比如 `json` 配置直接输入 `window`
- 和 `sublime` 等编辑器一样提供自动保存，保存会编译，然后在控制台抛错
- `wxss` 和 `wxml` 修改可以直接预览，`js`、`json` 点击编译后也可以预览
- 快捷键和其他 IDE 基本一致

关闭所有 VPN：一般发现工具控制台各种请求挂掉了的原因

通用性拆分

Ajax: 全局控制

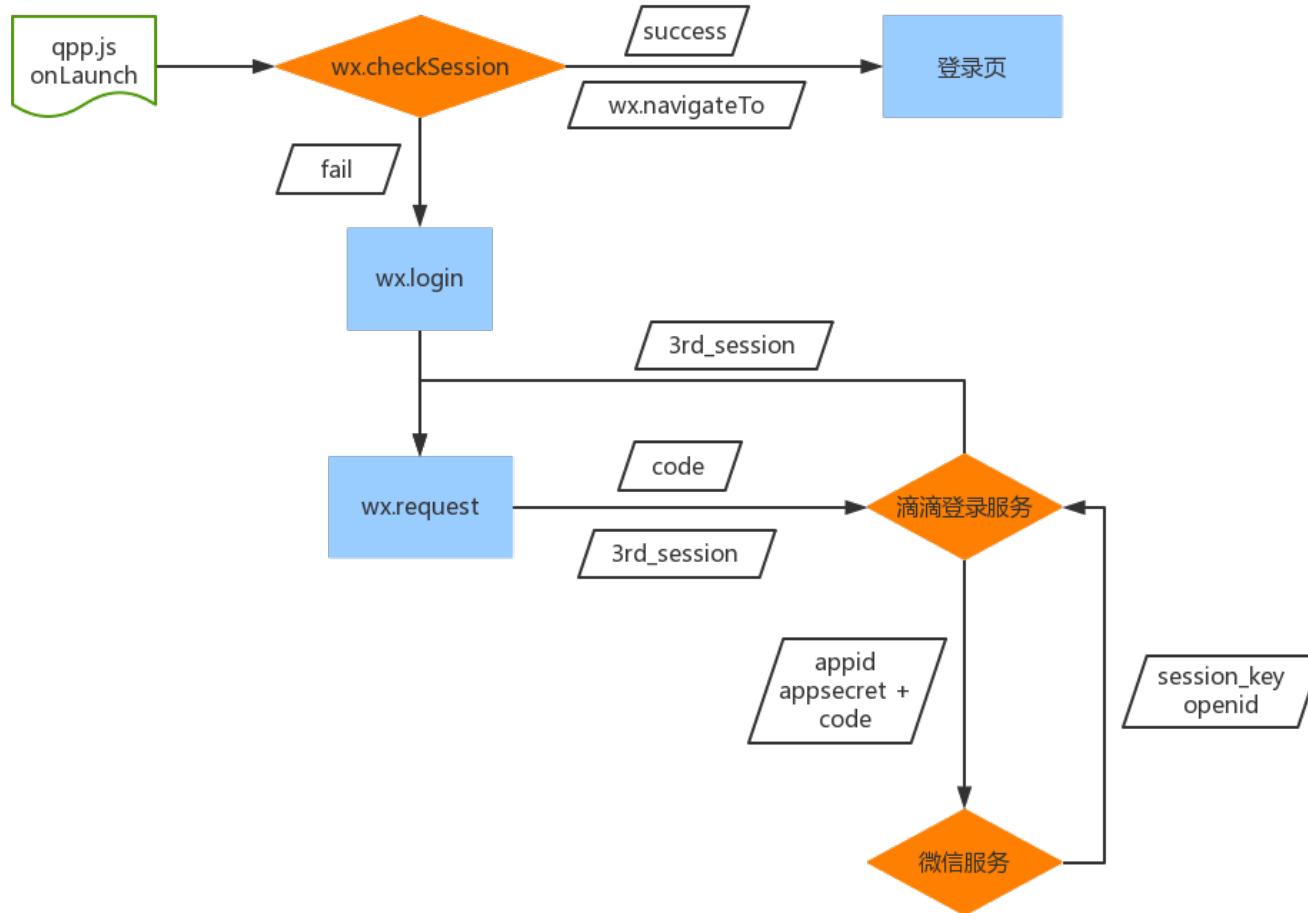
`wx.showToast`、`wx.request`、`wx.hideToast`

Util: 工具方法
字符串处理等

Config: 配置管理
Url、Action 等

登录

登录：除了微信登录之外还需要用户接入滴滴统一登录



支付

准备工作：需要在后台申请

接口：wx.requestPayment

支付参数、测试方式

团队开发 - 模块化

思考：发现一些页面部分内容是一致的：UI 和数据

实践：



复杂业务组件

ETA: 预估接驾时间

您好，您现在想去哪？

最快7分钟接驾

拼接分钟接驾

您好，您现在想去哪？

最快<text>4</text>分钟接驾

● 您在哪儿上车

● 您要去哪儿



```
complete: function(res){  
    var res = res.data,  
        result = JSON.parse(res.slice(0, res.length-1).slice(6))  
    that.setData({  
        minutes: '<text>' +result.ready_time + '</text>分钟'  
    })  
}
```

复杂业务组件

上车下车点：选择乘客上车点、下车点

北京市 您在哪上车	取消
北京市政府 正义路东	
正义路甲4号B座 东城区正义路甲4号B座	
正义路华纺酒店 东城区正义路甲4号	
北京东交民巷饭店写字楼 东城区东交民巷丙23号	
北京瑞安宾馆 东城区正义路7号(近最高人民法院)	
最高人民法院 东城区东交民巷27号	
东交民巷饭店 东城区东交民巷甲23号	
北京市经济委员会 东城区正义路2号	
正义路华纺酒店-B座 东城区正义路甲3号	
城市中文名或拼音	取消
当前城市：北京市	
★热门城市	
北京市	
上海市	
深圳市	
广州市	
武汉市	
A	
鞍山市	
安庆市	
安阳市	
阿拉善盟	
阿坝州	

- 个人简介
- 基本知识
- 团队开发
- **开发者工具**
- 编辑器开发
- 更多交流

开发者工具

安装目录： Resources/app.nw

第三方依赖：

- 前端
 - React
 - Flux
 - Imagesize.js
 - Jsqrcode.js
- “后” 端
 - Anyproxy
 - Request

配置

根目录 package.json 文件内容如下：

```
{  
  "main": "app/html/index.html",  
  "window": {  
    "title": "小春使用 - 微信web开发者工具 v0.10.102800",  
    ...  
  },  
  "inject_js_start": "app/dist/inject/jweixindebug.js",  
  "inject_js_end": "app/dist/inject/devtools.js",  
  "chromium-args": "--disable-devtools-ignores-certificate-errors  
-load-extension=/Applications/wechatwebdevtools.app/  
Contents/Resources/app.nw/app/dist/extensions/"  
  ...  
}
```

配置

配置文件： Resources/app.nw/package.json

基本配置： 入口文件 - app/html/index.html

- window
 - title
 - Icon
 - toolbar
 - width
 - height
 - frame
- chromium-args
 - load-extension
Resources/app.nw/app/dist/extensions/



调试器

文件目录: Resources/app.nw/app/dist/extensions

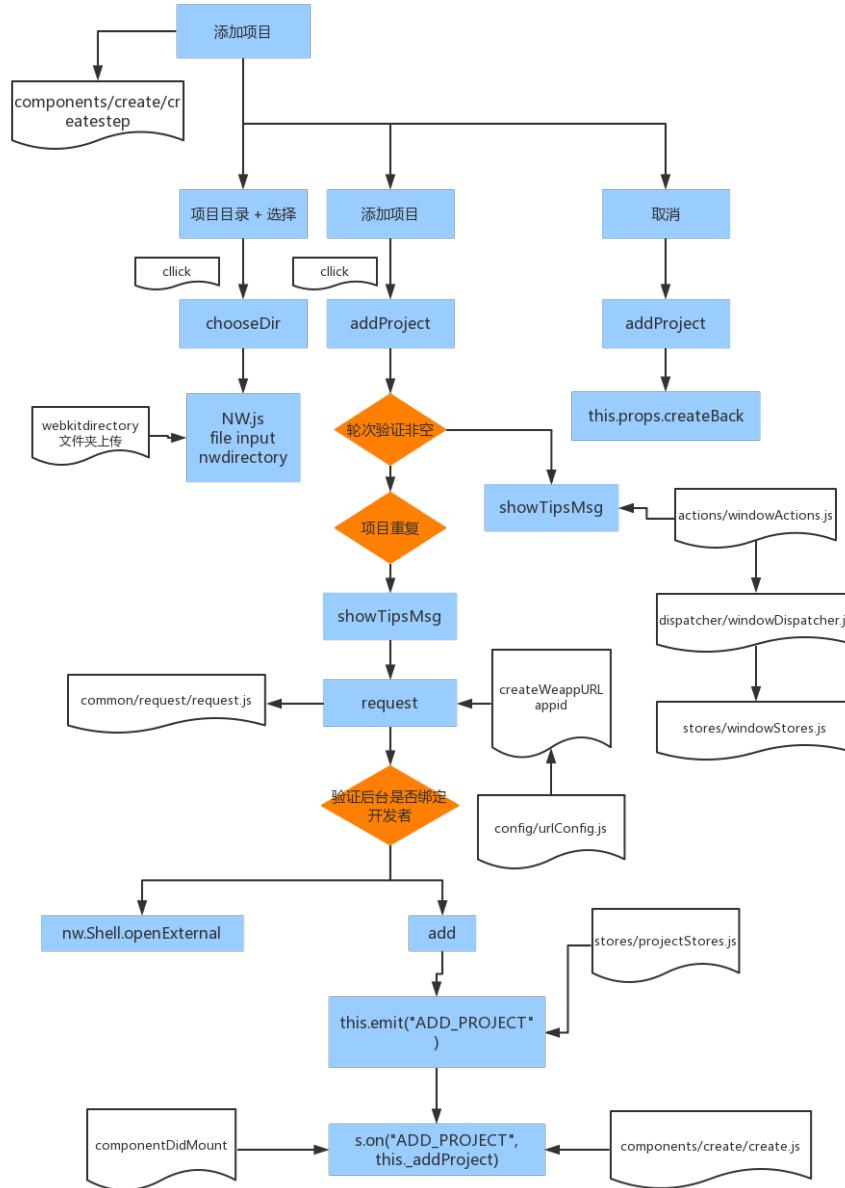


我的界面自定义的

修改 devtools.html

chrome.devtools.panels.create

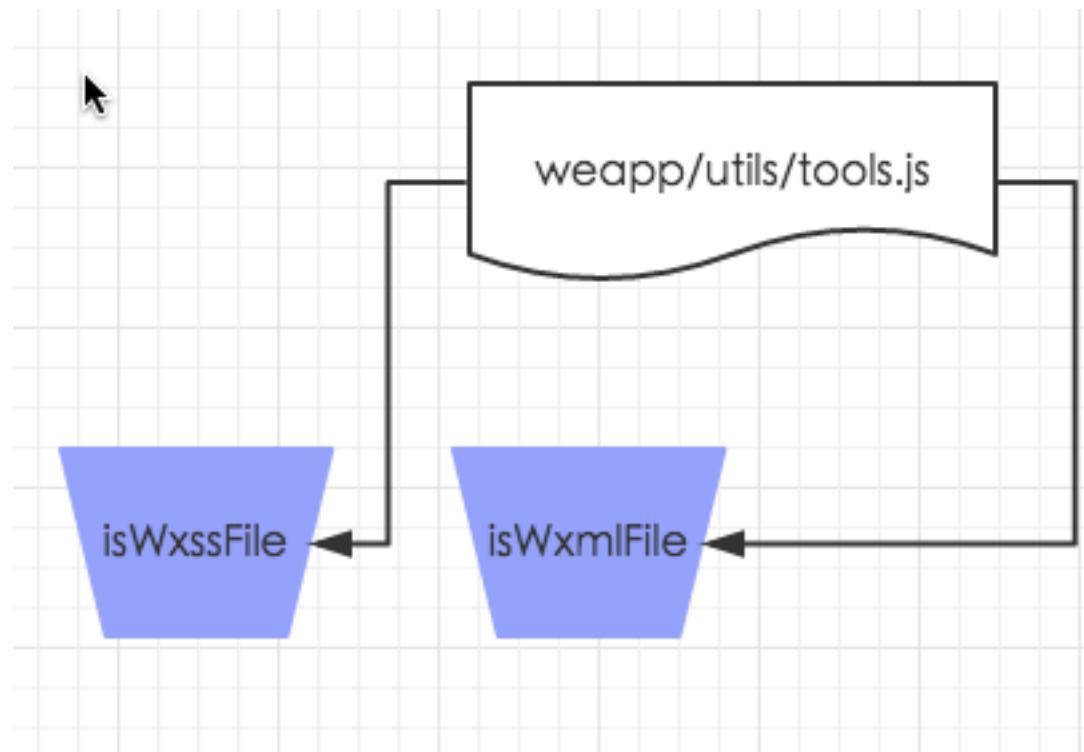
导入项目



文件判断

新的文件格式如何识别?

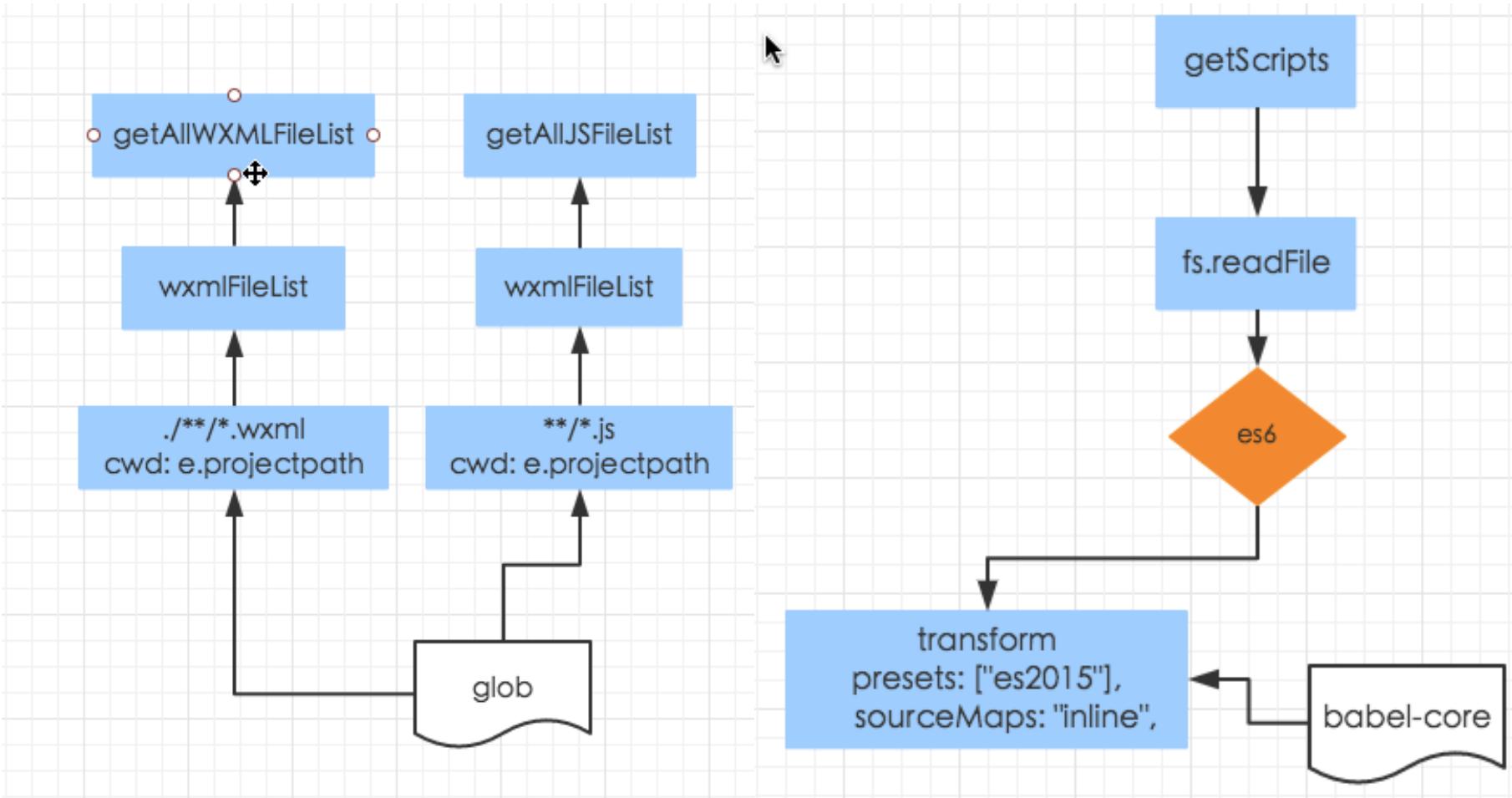
文件目录: Resources/app.nw/app/dist/webapp/utils



文件转换

新的文件格式如何转换?

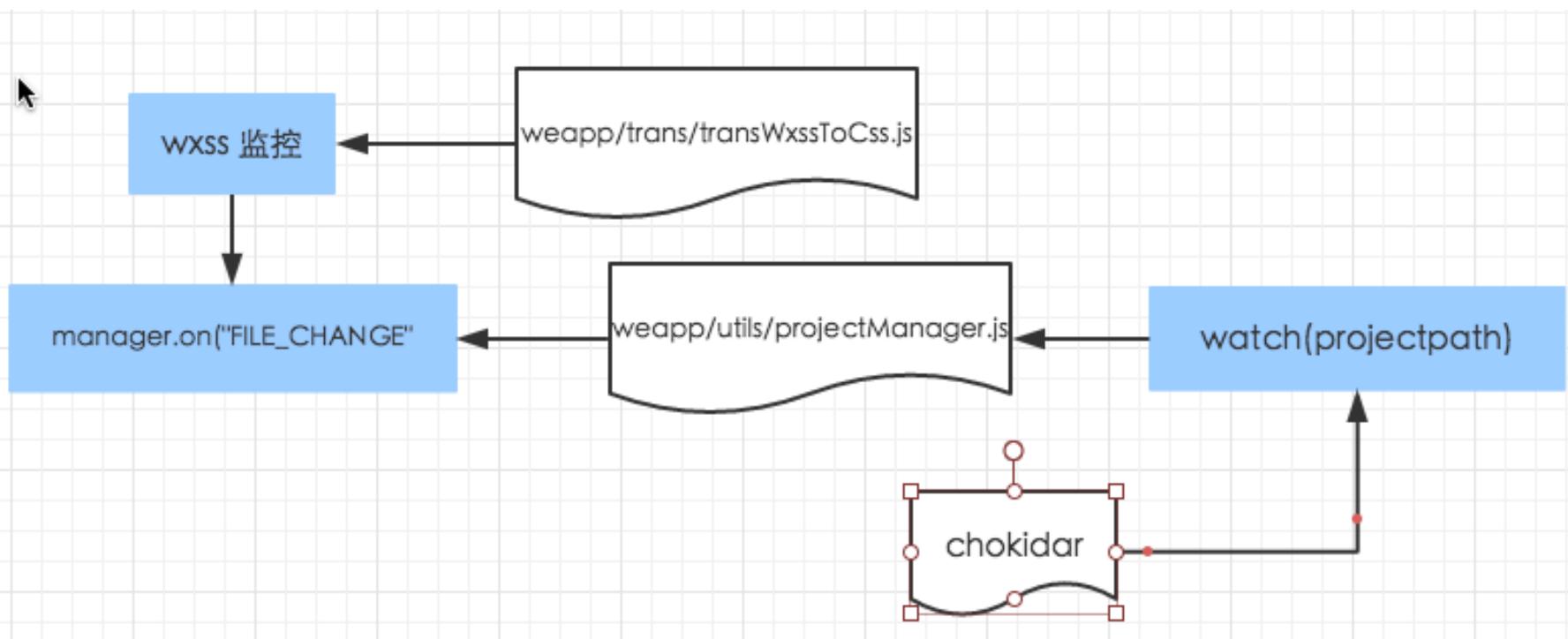
文件目录: Resources/app.nw/app/dist/webapp/trans



监控文件

如何监控 wxss 文件内容变更?

文件目录: Resources/app.nw/app/dist/webapp/trans

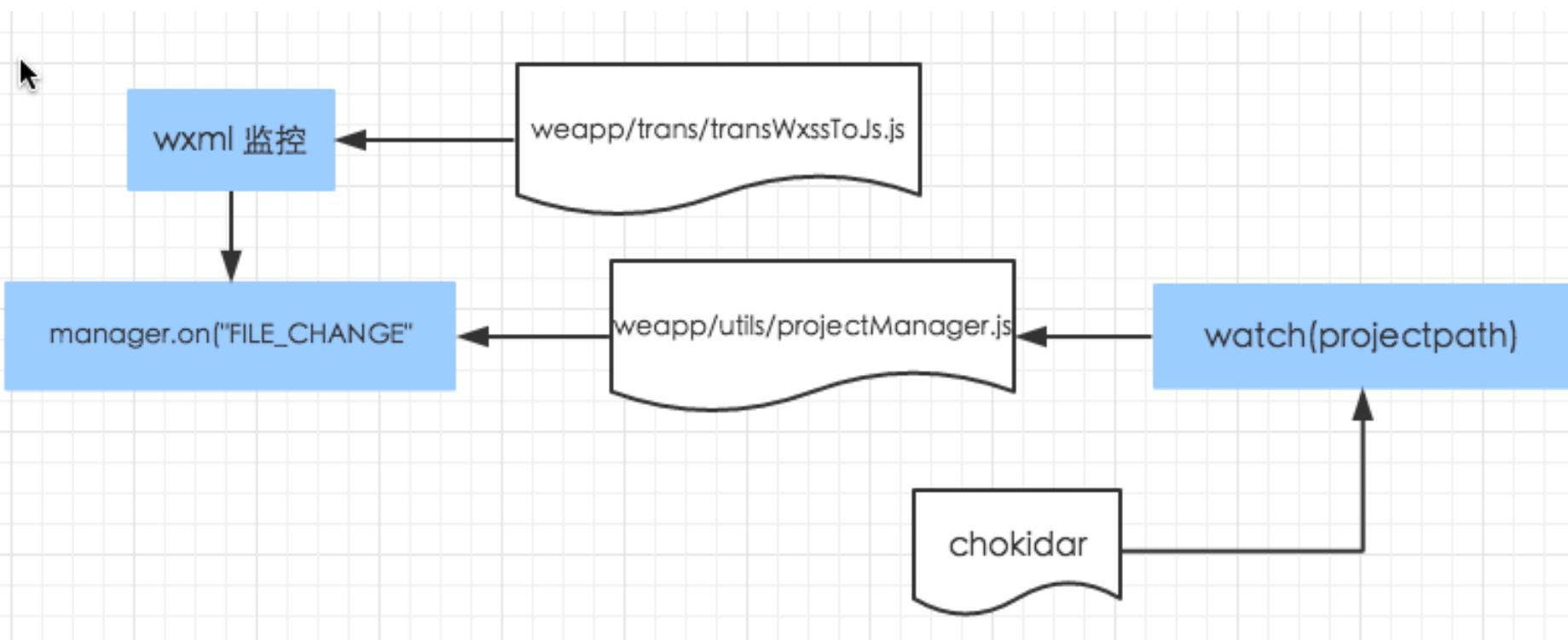


wxss 文件监控

监控文件

如何监控 wxml 文件内容变更?

文件目录: Resources/app.nw/app/dist/webapp/trans

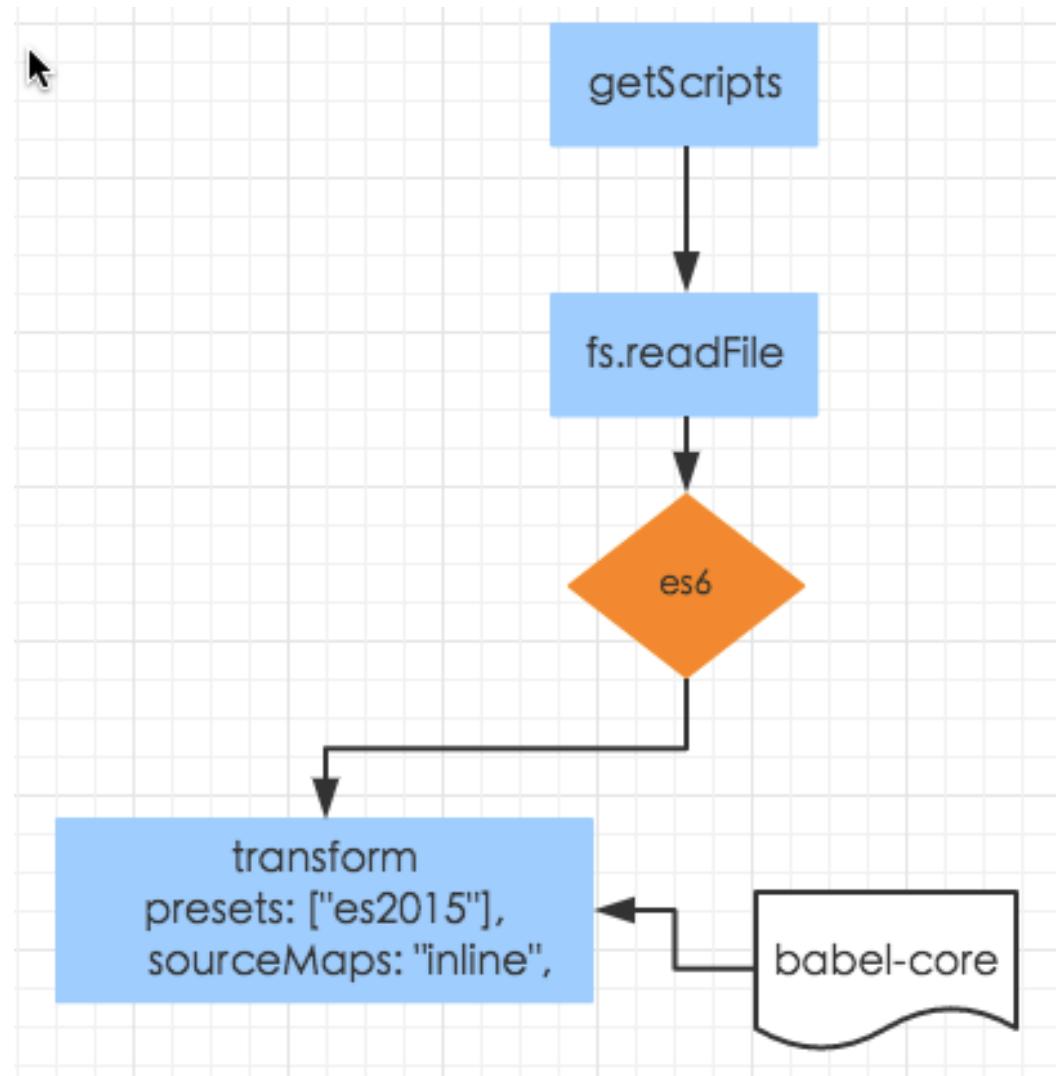


wxml 文件监控

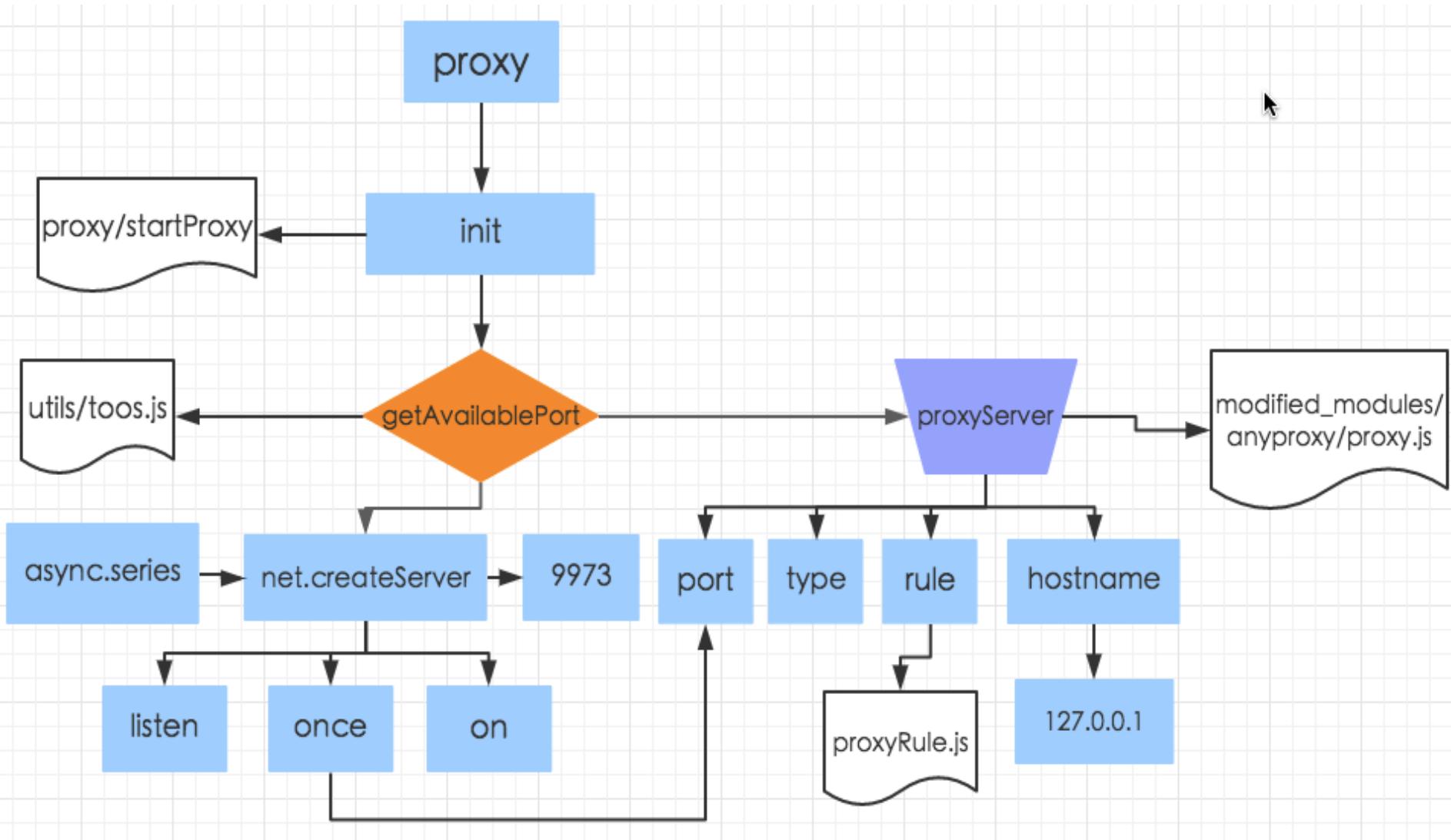
转换 ES6

ES6 编译环节：

-  开启 ES6 转 ES5
- 监听文件变化，自动刷新开发者工具
- 开启 代码压缩上传



代理服务



- 个人简介
- 基本知识
- 团队开发
- 开发者工具
- 编辑器开发
- 更多交流

编辑器研发

cloud9

两层目录： plugins-server | plugins-client

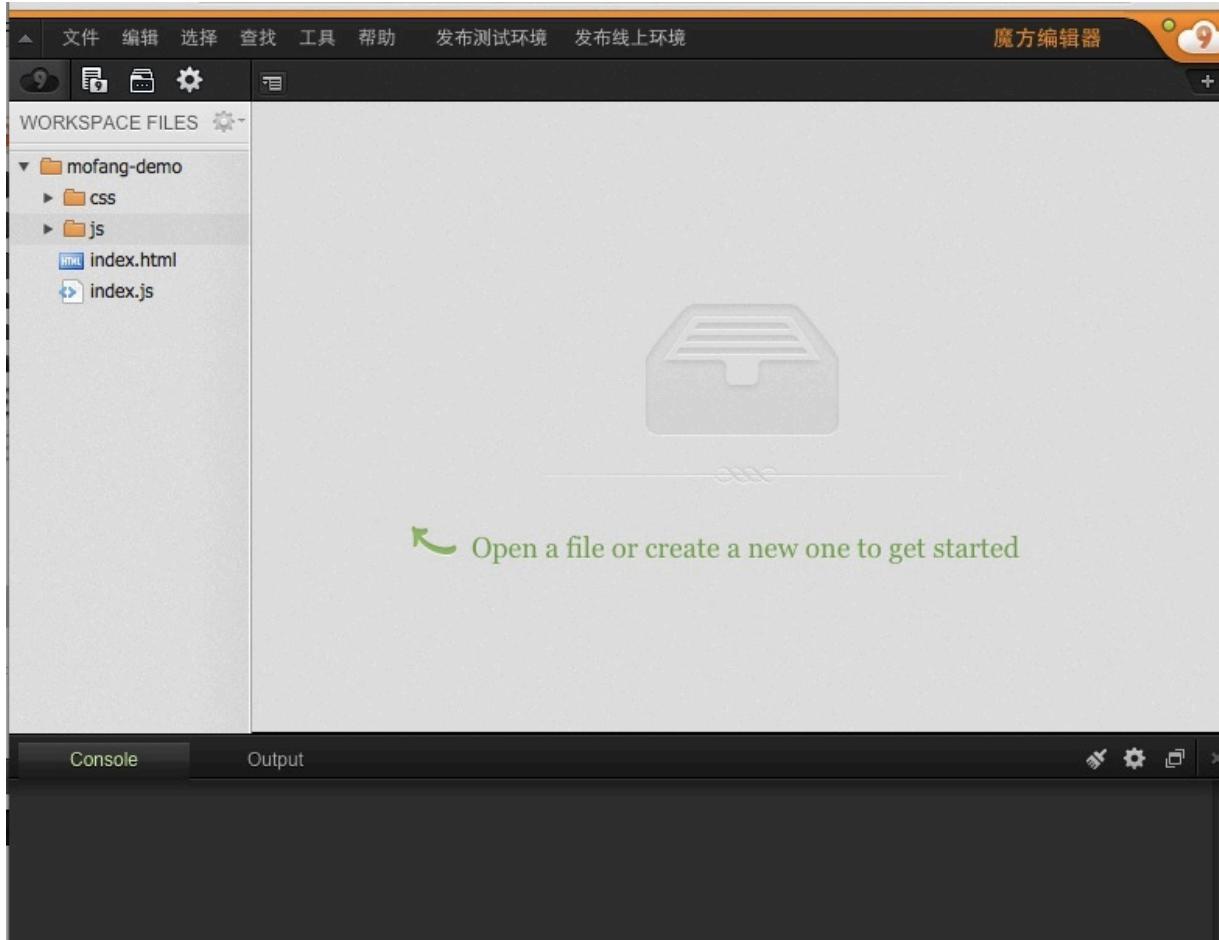
- 前端
 - ext.**
 - lib.**
- “后” 端 Node.js
 - jsDAV
 - glob
 - 核心包 fs 等

Electron – Atom Shell

也是使用 Chromium 显示， 最新： 1.4.7 基于 Node 6.5.0

编辑器研发

滴滴魔方 CODE IDE



更多交流

DDFE - 滴滴前端技术公众号





SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

谢 谢 !