

H5跨平台代码 解决方案

成都 2017.11.19

如何理解H5跨平台？

这里的平台通俗一点即是h5代码运行时的webview环境，常规的比如手机自带浏览器，APP提供的webview，微信webview等等。

除了H5自身规范，这些平台大多提供了更强的原生能力

前端面临的问题？

平台越来越多，前端开发人员不够，但其实业务流程都相对统一；

如何以不变应万变需要大家根据自身公司情况去探索。



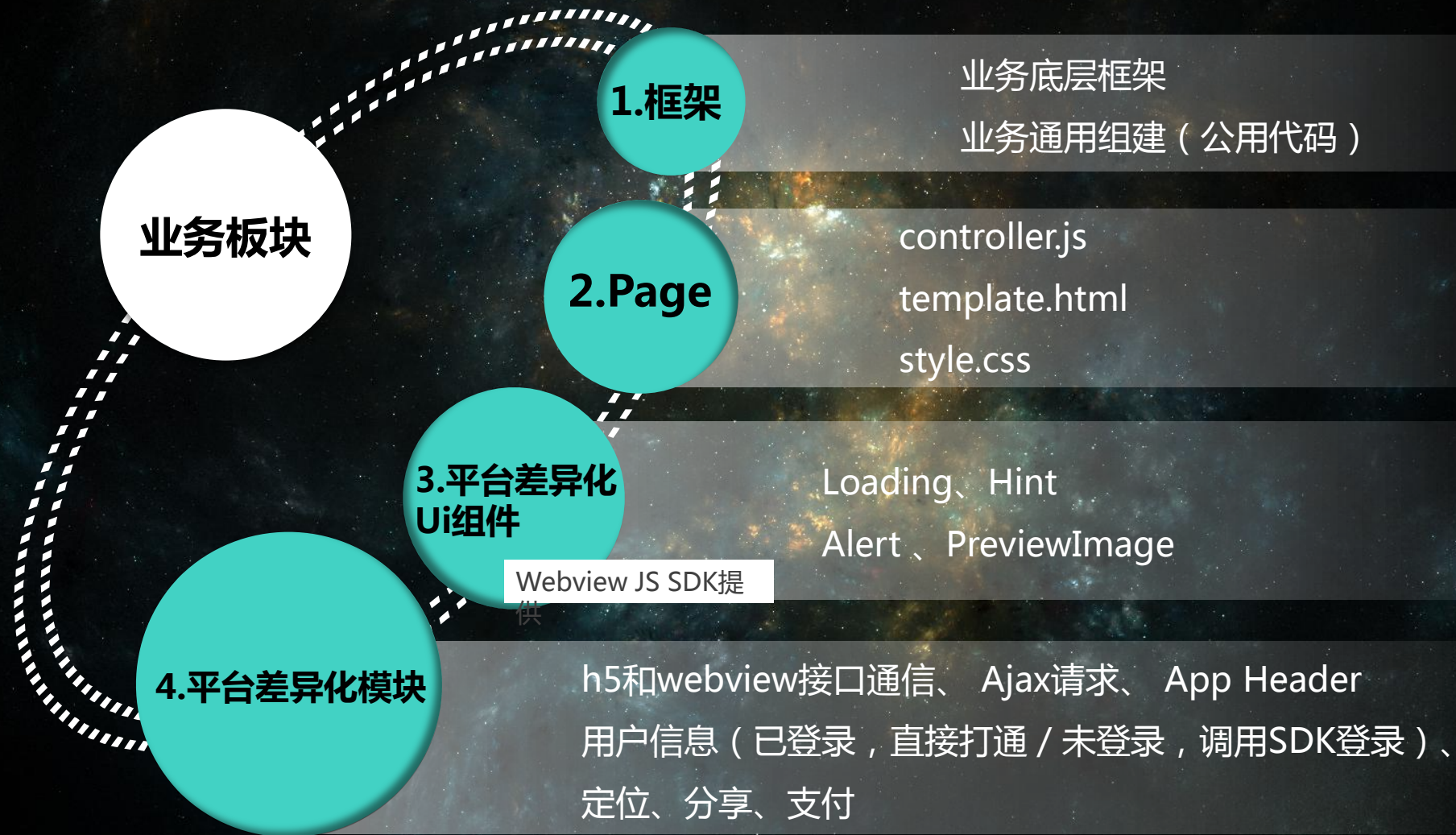
乐车邦前端面临的问题

定制化的流量平台非常多，与传统的营销参数不同，这些平台都必须对接他们的API。

常用的如：乐车邦APP, 微信公众号，微信（支付宝）小程序，手淘（天猫），百度地图，京东，汽车之家，百度糯米，58到家，新美大等等，虽然渠道众多，但是业务流程相对统一。

如何利用“一套”h5代码解决多终端运行问题成了前端重中之重！





如何理解代码统一

这里的代码统一指的是业务代码统一，底层去封装流量平台的差异性。

封装h5定位

```
export function getPosition () {  
  return new Promise((resolve, reject) =>  
  {  
    if ( '乐车邦APP' ) {  
    } else if ( '微信' ) {  
    } else if (navigator.geolocation) {  
    }  
  })  
}
```

业务代码

```
getPosition().then(() => {  
  })
```

框架代码设计

```
var Lizard = {  
  config () {  
  }  
  goTo () {  
  }  
  goBack () {  
  }  
  alert () {  
  }  
  hint () {  
  }  
  showLoading () {  
  }  
  hideLoading () {  
  }  
  ...  
}
```

PAGEVIEW类

```
class PageView {  
  onCreate () {  
  }  
  ajax () {  
  }  
  dataLink () {  
  }  
  onShow () {  
  }  
  onCache () {  
  }  
  onHide () {  
  }  
  onRemove () {  
  }  
  ...  
}
```


UI 抽象类

```
class UIAbstract {  
}  
class ScrollList extends UIAbstract {  
}  
class SlideShow extends UIAbstract {  
}  
class Loading extends UIAbstract {  
}  
class Switch extends UIAbstract {  
}  
...
```

*各平台可以自行实现Alert,Loading,Hint等等UI组件，但是接口必须保持一致

桥架模块设计

```
export function call () {  
}  
export function request() {  
}  
export function setHeader () {  
}  
export function auth () {  
}  
export function getPosition () {  
}  
export function payment () {  
}  
...
```

*LizardBridge各流量平台自行实现，接口保持一致，主要负责h5和webview交互的api

通用WEBVIEW平台

1. 都是直连https模式
2. 大部分流量平台都是通用的webview，提供了自己的JS SDK
3. 只要设计好统一的API,通过桥接总是能保证业务代码是一致的

APP平台

通过打包程序，把h5代码本地化，节省静态资源的加载

微信小程序平台

1. controller.js复用
2. page.html 通过打包程序，转成wxml模板
3. page.css 通过打包程序，转成wxsss

微信小程序

1. controller.js如何复用

- * 提取Vue2.0数据监听模块

- * 利用setData，在小程序里面实现MVVM

2. Vue模板编译成wxml

3. css编译成wxss

- *在小程序平台下面实现一套框架，提供一致的API给业务使用，保证业务代码的复用性

小程序MVVM 核心代码

```
vm.$watch(function() {  
  let result = {}  
  
  for (let key in data) {  
    result[key] = cloneDeep(this[key])  
  }  
  
  for (let key2 in computed) {  
    result[key2] = this[key2]  
  }  
  
  return result  
}, function(newVal, oldVal) {  
  let diff = diffObject(newVal, oldVal)  
  wxPage.setData(diff)  
})
```



THANK YOU!