

本期主题：微信小程序架构解析

**讲座嘉宾：腾讯前端开发高级工程师
渠宏伟**

开始时间：15:00

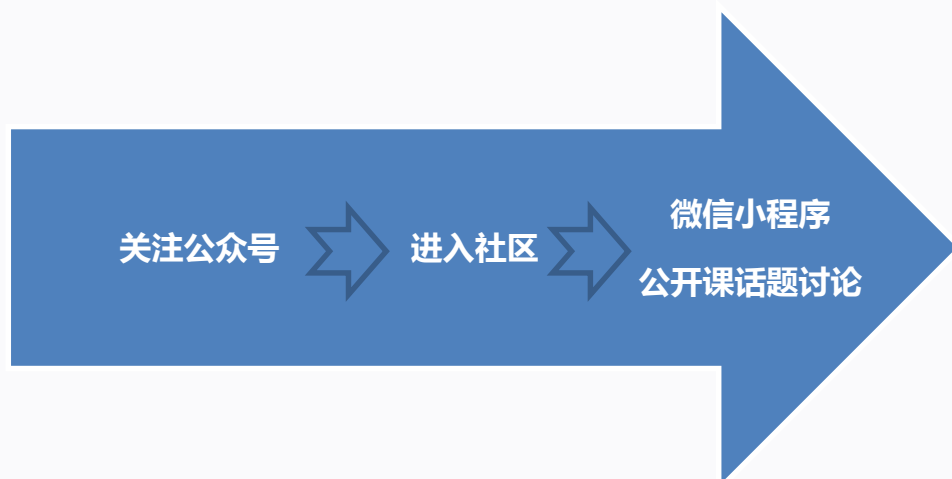


业界最顶尖的技术大咖 / 最权威的实战分享
最前沿的行业资讯 / 尽在腾讯课堂Coding学院



Coding学院
带你领略编程之美

• 互动提问方式：



腾讯课堂Coding学院公众号

微信小程序框架解析

hongweiqu(渠宏伟)



CONTENTS

01 小程序介绍和演示

02 小程序架构

03 小程序视图层

04 小程序逻辑层

05 小程序开发经验



小程序介绍

小程序的特点

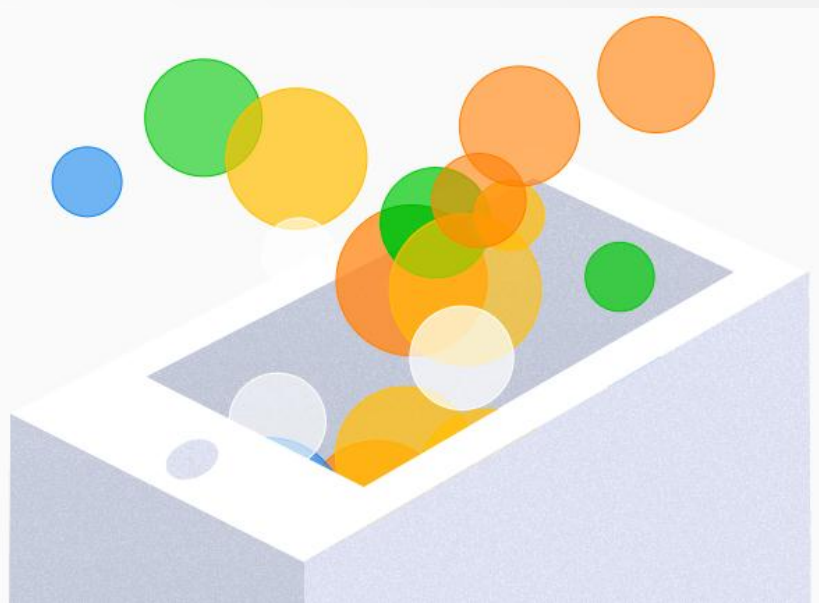
类WEB 不是HTML5

即用即走，随手可得


拥有离线能力

基于微信跨平台

媲美原生操作体验





 微信扫一扫，使用小程序

企鹅电竞小程序体验

为什么那么快



common lib
user common.js

Page Frame

Native预先加载一个WebView

当打开指定页面时，无需加载额外资源直接渲染

返回显示历史View

退出小程序，View状态不销毁

小程序入口



Allen Zhang

程序猿的一小步，程序的一大步



27分钟前



♡ 香文斌, ayrton, tao@西藏骑行, Hata

扫码进入小程序

搜索小程序

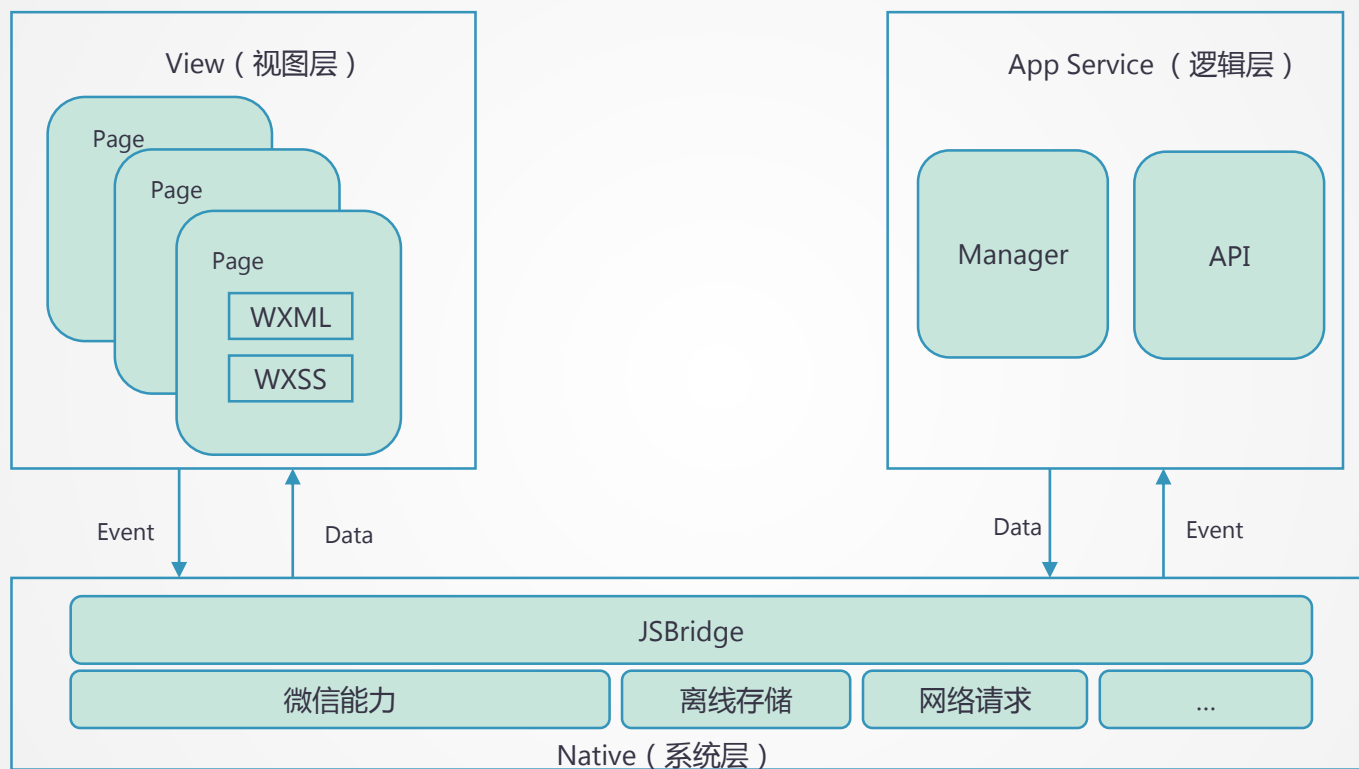
发现->小程序

小程序发送到桌面 (android)

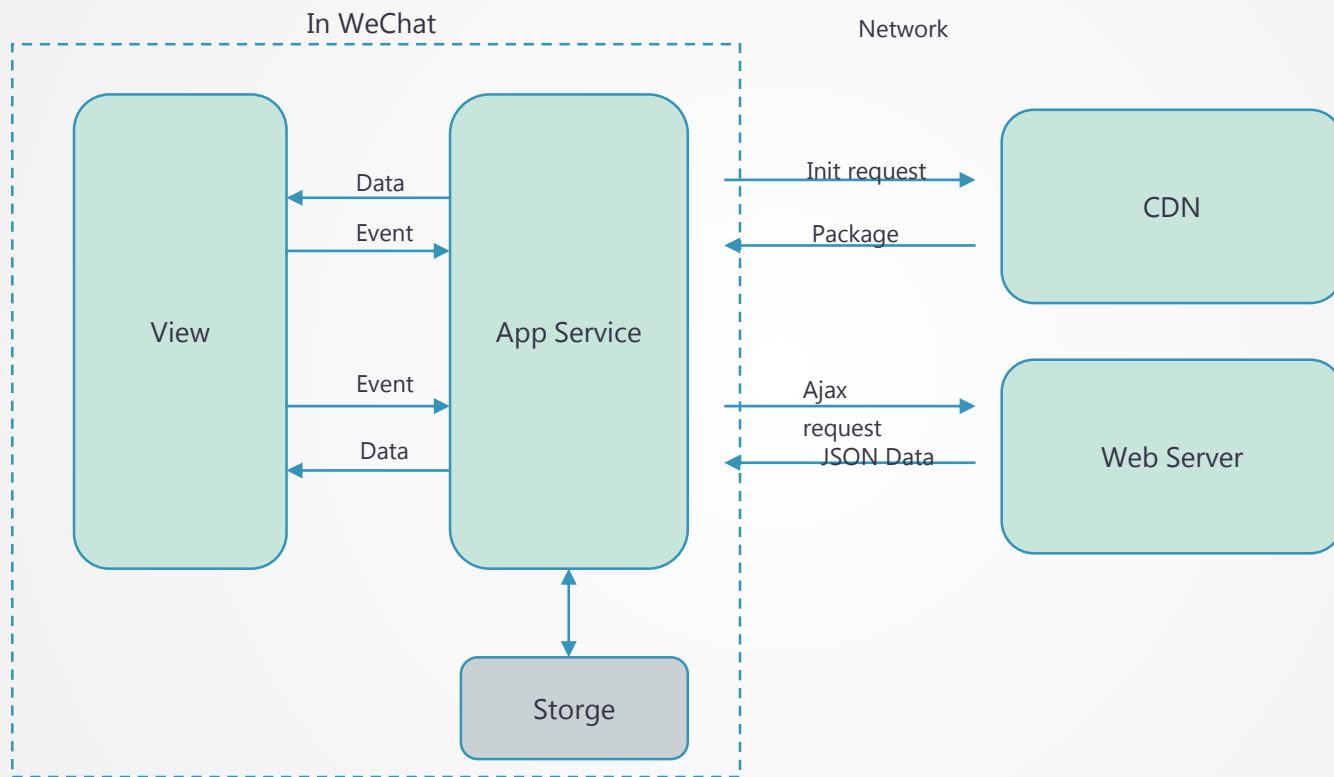


小程序架构

小程序架构



小程序架构





View (页面视图)

WXML (WeiXin Markup Language)

支持数据绑定

支持逻辑算术、运算

支持模板、引用

支持添加事件 (bindtap)



WXSS(WeiXin Style Sheets)

支持大部分CSS特性

添加尺寸单位rpx，可根据屏幕宽度自适应

使用@import语句可以导入外联样式表

不支持多层选择器-避免被组件内结构破坏



View – WXSS Selectors

选择器	样例	样例描述
.class	.intro	选择所有拥有 class="intro" 的组件
#id	#firstname	选择拥有 id="firstname" 的组件
element	view	选择所有 view 组件
element, element	view, checkbox	选择所有文档的 view 组件和所有的 checkbox 组件
::after	view::after	在 view 组件后边插入内容
::before	view::before	在 view 组件前边插入内容

View - Component

大类	细分	WXML	HTML
视图容器	普通视图容器	view	div、ul、li、article、section.....
	滚动视图	scroll-view	
	滑块视图	swiper	
基础内容	文本	text	span、em、p.....
	图标	icon	span、em、i.....
	进度条	progress	div.....
链接、导航	链接、导航	navigator	a
表单	按钮	button	button、input
	单选	radio	input
	多选	checkbox	input
	表单	form	form
	输入框	input	input
	改进表单可用性	label	label
	滚动选择器	picker	select
	开关选择器	switch	input、div.....
操作反馈	底部菜单	action-sheet	div、ul.....
	弹窗	modal	div.....
	toast	toast	div.....
	加载	loading	img、span、div.....
媒体	图片	image	img
	音频	audio	audio
	视频	video	video
地图	地图	map	map
画布	画布	canvas	canvas

View - Component

小程序的组件基于Web Component标准

使用Polymer框架实现Web Component

```
<!-- add style to your element -->
<dom-module id="my-styled-namecard">
  <template>
    <style>
      /* This would be crazy in non webcomponents. */
      span {
        font-weight: bold;
      }
    </style>

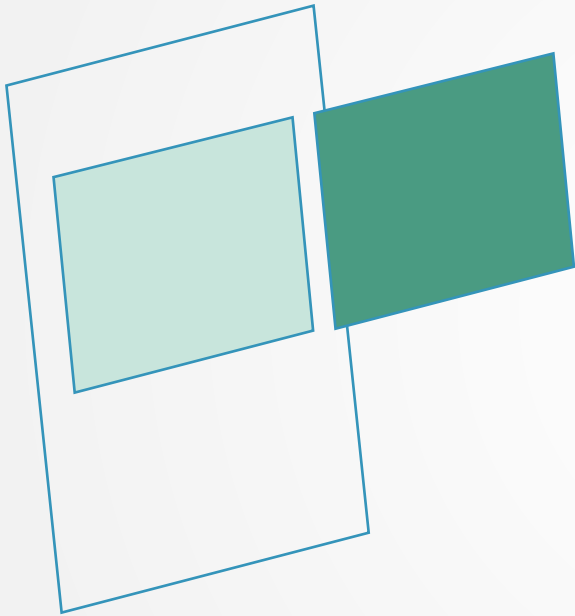
    <div>
      Hi! My name is <span>{{myName}}</span>
    </div>
  </template>

  <script>
    Polymer({
      is: 'my-styled-namecard',
      properties: {
        myName: {
          type: String
        }
      }
    });
  </script>
</dom-module>
```

```
<!-- using the element -->
<my-styled-namecard my-name="Jesse"></my-styled-namecard>
```

Hi! My name is Jesse

View - Native Component



Native组件层在WebView层之上

<canvas / ><video / ><map / ><textarea / >



App Service(逻辑层)

逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈

- 1、App() 小程序的入口；Page() 页面的入口
- 3、提供丰富的 API，如微信用户数据，扫一扫，支付等微信特有功能。
- 4、每个页面有独立的作用域，并提供模块化能力。
- 5、数据绑定、事件分发、生命周期管理、路由管理

运行环境

IOS - JSCore

Android - X5 JS解析器

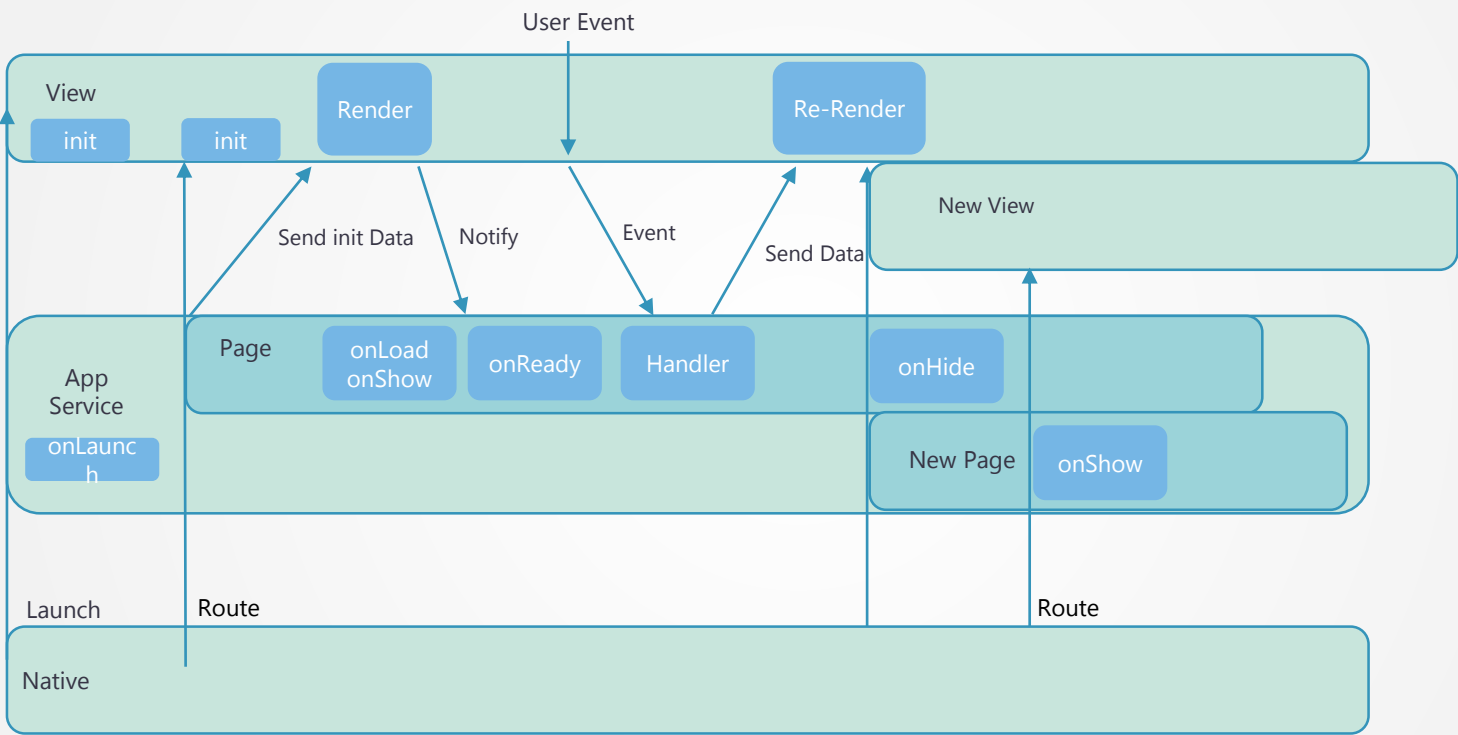
DevTool - nwjs Chrome 内核

App Service - Binding

```
<swiper class="banner-swiper" bindchange="bindSwiperChange" catchtap="goLive" indicator-dots="{{indicatorDots}}"
  autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}">
  <block wx:for="{{banner}}">
    <swiper-item>
      <image src="{{item.pic}}" class="slide-image" width="100%" height="200"/>
    </swiper-item>
  </block>
</swiper>
```

```
var app = getApp()
Page({
  data: {
    banner: [],
    indicatorDots: false,
    autoplay: true,
    interval: 5000,
    duration: 1000
  },
  onLoad: function () { ...
  },
  bindSwiperChange: function(event){ ...
  },
  goLive: function(event){
    var anchorid=event.target.dataset.anchorid;
    wx.navigateTo({
      url: ['/live/index?anchorid=',anchorid].join('')
    })
    return false;
  },
  buildBanner: function(items){ ...
  },
  getLiveHome: function(callback){ ...
  }
})
```

App Service - Life Cycle



App Service - API

API通过JSBridge和Native 进行通信

分类	名称	API
网络	发起请求	request
	上传下载	uploadFile / downloadFile
	WebSocket	connectSocket / onSocketOpen / onSocketError / sendSocketMessage / onSocketMessage / closeSocket / onSocketClose
媒体	图片	chooseImage / previewImage / getImageInfo
	视频	chooseVideo
	音频	playVoice / pauseVoice / stopVoice
	录音	startRecord / stopRecord
	文件	saveFile / getSavedFileList / getSavedFileInfo / removeSavedFile
数据	数据缓存	setStorage / getStorage / removeStorage / clearStorage
位置	获取位置	getLocation
	查看位置	openLocation
设备	网络状态	getNetworkType
	系统信息	getSystemInfo
	重力感应	onAccelerometerChange
	罗盘	onCompassChange
	拨打电话	makePhoneCall
界面	Canvas	drawCanvas / drawCanvas / canvasToTempFilePath
	动画	createAnimation
	导航、设置导航条	navigateTo / redirectTo / navigateBack
	交互反馈	showToast / showModal / showActionSheet
开放接口	登录	login
	用户信息	getUserInfo
	微信支付	requestPayment
	模板消息	

`navigateTo(OBJECT)`

保留当前页面，跳转到应用内的某个页面，使用`navigateBack`可以返回到原页面。页面路径只能是五层

`redirectTo(OBJECT)`

关闭当前页面，跳转到应用内的某个页面。

`navigateBack(OBJECT)`

关闭当前页面，返回上一页面或多级页面。可通过 `getCurrentPages()` 获取当前的页面栈，决定需要返回几层。



小程序开发经验

小程序可以借鉴的优点

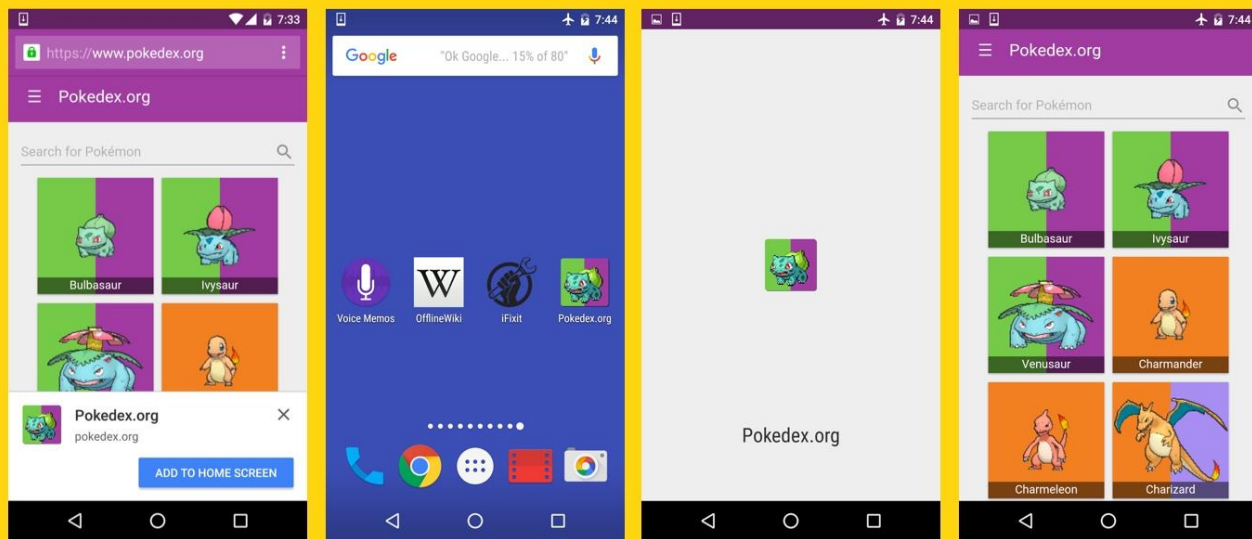
- 1、提前新建WebView，准备新页面渲染。
- 2、View层和逻辑层分离，通过数据驱动，不直接操作DOM。
- 3、使用Virtual DOM，进行局部更新。
- 4、全部使用https，确保传输中安全。
- 5、前端组件化开发。
- 6、加入rpx单位，隔离设备尺寸，方便开发。

小程序存在的问题

- 1、小程序仍然使用WebView渲染，并非原生渲染
- 2、需要独立开发，不能在非微信环境运行。
- 3、开发者不可以扩展新组件。
- 4、服务端接口返回的头无法执行，比如：Set-Cookie。
- 5、依赖浏览器环境的js库不能使用，因为是JSCore执行的，没有window、document对象。
- 6、WXSS中无法使用本地（图片、字体等）。
- 7、WXSS转化成js 而不是css，为了兼容rpx。
- 8、WXSS不支持级联选择器。
- 9、小程序无法打开页面，无法拉起APP。

脱离微信的“小程序”：PWA 渐进式应用

PWA - Progressive Web App



Web App install
banner for engagement

Launch from user's
home screen

Splash screen
(Chrome for Android 47+)

Works offline with
Service Worker

PWA 特点

- **渐进增强** - 支持的新特性的浏览器获得更好的体验，不支持的保持原来的体验。
- **离线访问** - 通过 Service Workers 可以在离线或者网速差的环境下工作。
- **类原生应用** - 使用app shell model做到原生应用般的体验。
- **可安装** - 允许用户保留对他们有用的应用在主屏幕上，不需要通过应用商店。
- **容易分享** - 通过 URL 可以轻松分享应用。
- **持续更新** - 受益于 service worker 的更新进程，应用能够始终保持更新。
- **安全** - 通过 HTTPS 来提供服务来防止网络窥探，保证内容不被篡改。
- **可搜索** - 得益于 W3C manifests 元数据和 service worker 的登记，让搜索引擎能够找到 web 应用。
- **再次访问** - 通过消息推送等特性让用户再次访问变得容易。

PWA - Web App Manifest使Web更像Native

Web App Manifest以JSON的格式定义Web应用的相关配置（应用名称、图标或图像连接、启动URL、自定义特性、启动默认配置、全屏设置等）。

```
{
  "short_name": "Pokedex.org",
  "name": "Pokedex.org",
  "display": "standalone",
  "icons": [
    {
      "src": "img/icon-48.png",
      "sizes": "48x48",
      "type": "image/png"
    },
    {
      "src": "img/icon-96.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "img/icon-144.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "img/icon-196.png",
      "sizes": "196x196",
      "type": "image/png"
    },
    {
      "src": "img/icon-384.png",
      "sizes": "384x384",
      "type": "image/png"
    }
  ],
  "start_url": "index.html?launcher=true",
  "theme_color": "#a040a0",
  "background_color": "#EEEEEE"
}
```

App Manifest

<https://www.pokedex.org/manifest.json>

Identity

Name	Pokedex.org
Short name	Pokedex.org

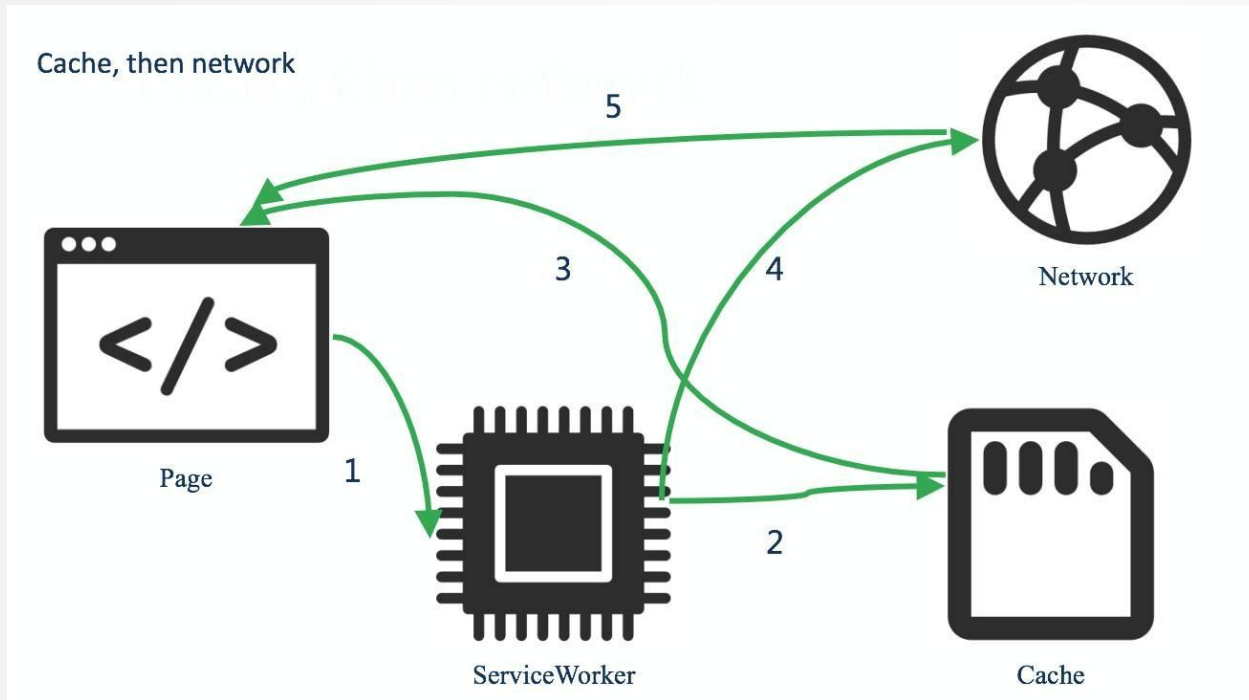
Presentation

Start URL	index.html?launcher=true
Theme color	■ #a040a0
Background color	□ #EEEEEE
Orientation	
Display	standalone

Icons



PWA - Service Workers增强Web能力

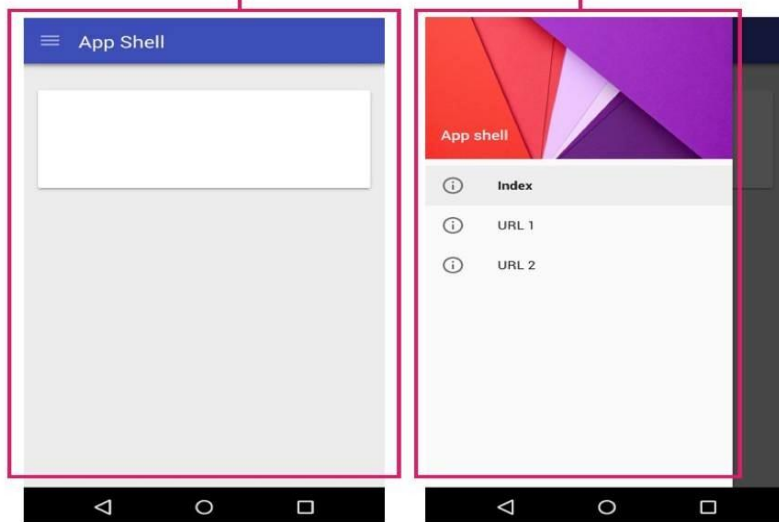


通过Service Works实现资源离线缓存和更新

PWA - App Shell 提升显示效率

App Shell (应用外壳) 是应用的用户界面所需的最基本的 HTML、CSS 和 JavaScript，首次加载后立刻被缓存下来，不需要每次使用时都被下载，而是只异步加载需要的数据，以达到UI保持本地化。

application shell



Cached shell loads **instantly** on repeat visits.

content



Dynamic content then populates the view



THANK YOU

本期主题：微信小程序架构解析
讲座嘉宾：腾讯前端开发高级工程师
渠宏伟
开始时间：15:00



业界最顶尖的技术大咖 / 最权威的实战分享
最前沿的行业资讯 / 尽在腾讯课堂Coding学院



微信小程序框架解析

hongweiqu(渠宏伟)



CONTENTS

01 小程序介绍和演示

02 小程序架构

03 小程序视图层

04 小程序逻辑层

05 小程序开发经验



小程序介绍

小程序的特点

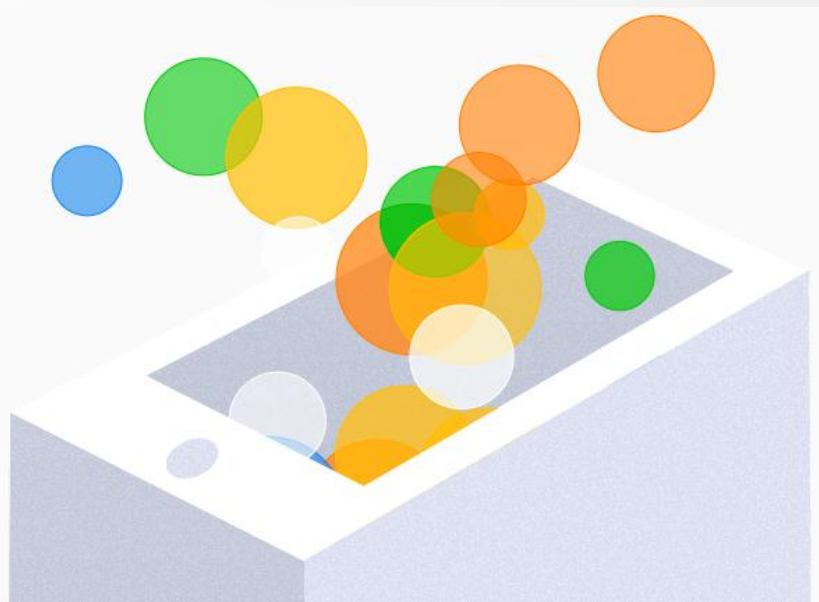
类WEB 不是HTML5

即用即走，随手可得


拥有离线能力

基于微信跨平台

媲美原生操作体验





 微信扫一扫，使用小程序

企鹅电竞小程序体验

为什么那么快



common lib
user common.js

Page Frame

Native预先加载一个WebView

当打开指定页面时，无需加载额外资源直接渲染

返回显示历史View

退出小程序，View状态不销毁

小程序入口



Allen Zhang

程序猿的一小步，程序的一大步



27分钟前



♡ 香文斌, ayrton, tao@西藏骑行, Hata

扫码进入小程序

搜索小程序

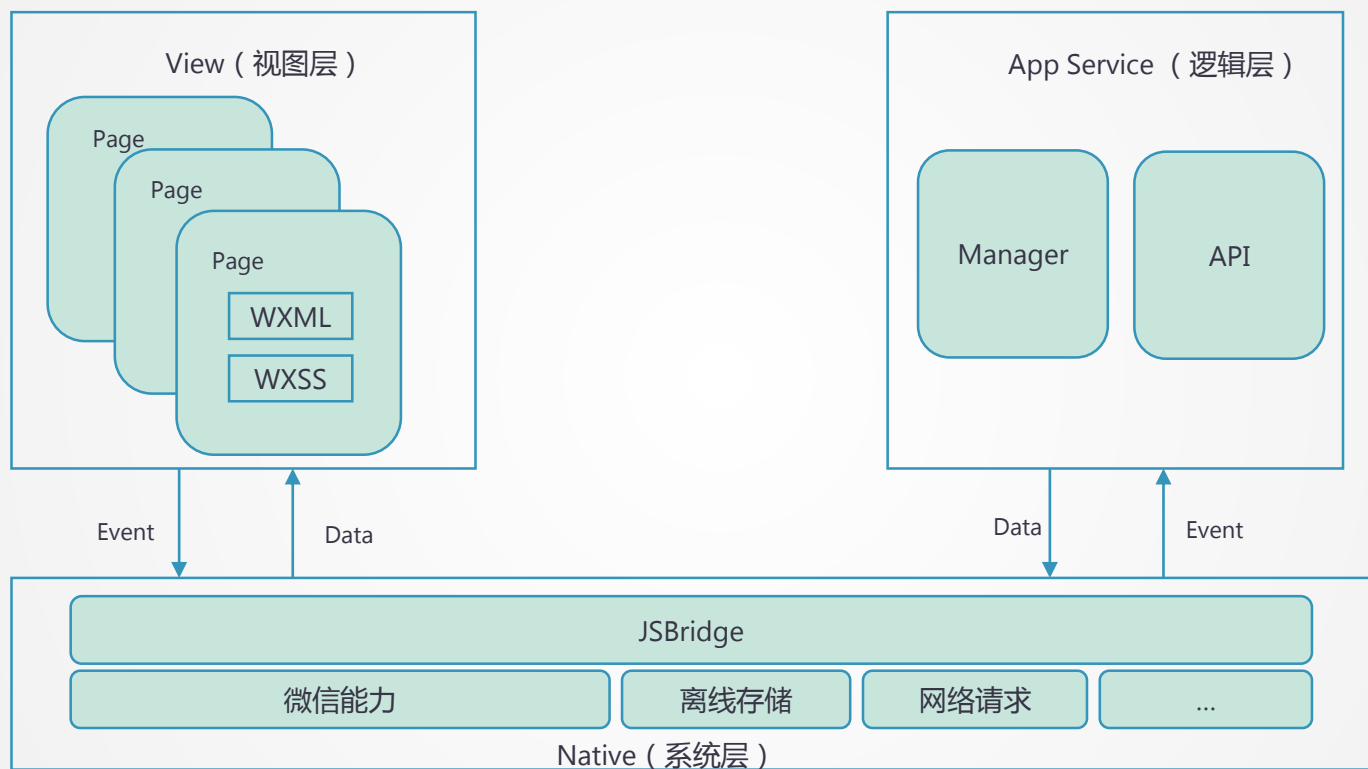
发现->小程序

小程序发送到桌面 (android)

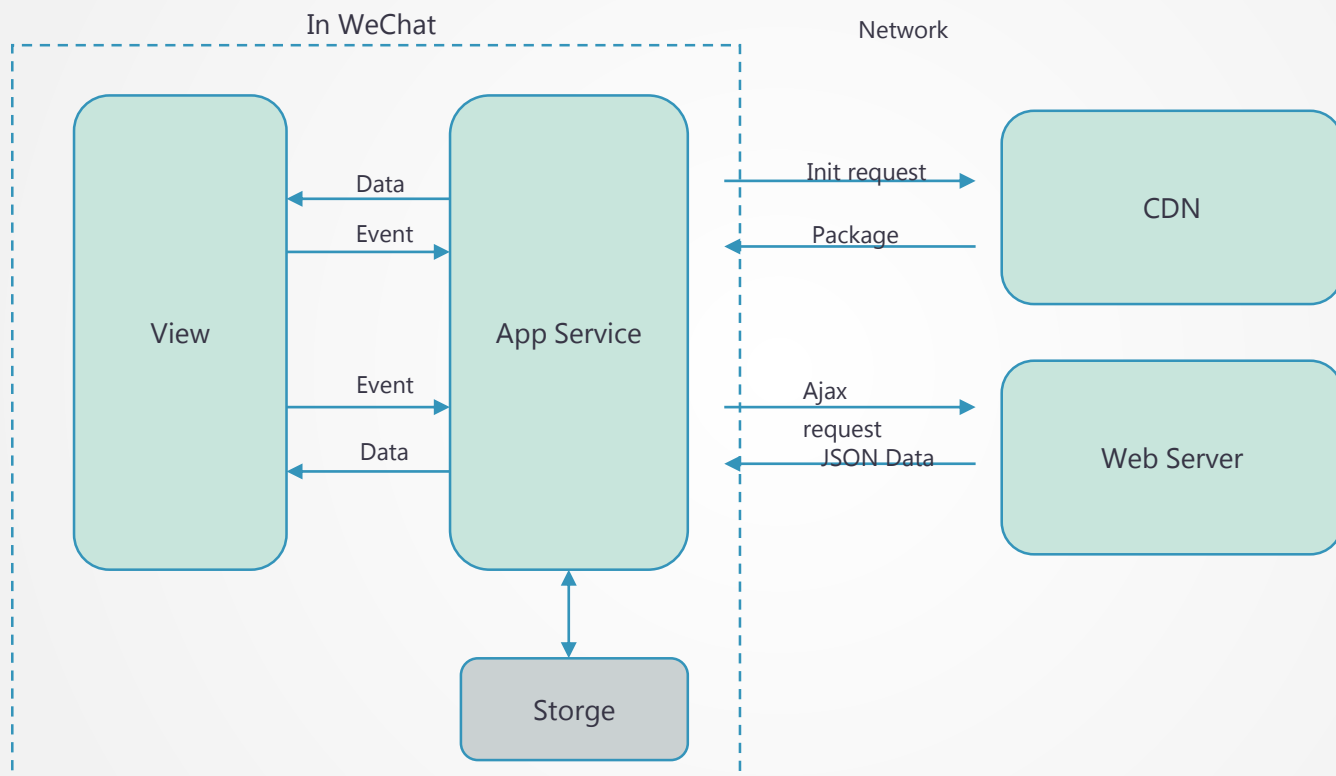


小程序架构

小程序架构



小程序架构





View (页面视图)

WXML (WeiXin Markup Language)

支持数据绑定

支持逻辑算术、运算

支持模板、引用

支持添加事件 (bindtap)



WXSS(WeiXin Style Sheets)

支持大部分CSS特性

添加尺寸单位rpx，可根据屏幕宽度自适应

使用@import语句可以导入外联样式表

不支持多层选择器-避免被组件内结构破坏



View – WXSS Selectors

选择器	样例	样例描述
.class	.intro	选择所有拥有 class="intro" 的组件
#id	#firstname	选择拥有 id="firstname" 的组件
element	view	选择所有 view 组件
element, element	view, checkbox	选择所有文档的 view 组件和所有的 checkbox 组件
::after	view::after	在 view 组件后边插入内容
::before	view::before	在 view 组件前边插入内容

View - Component

大类	细分	WXML	HTML
视图容器	普通视图容器	view	div、ul、li、article、section.....
	滚动视图	scroll-view	
	滑块视图	swiper	
基础内容	文本	text	span、em、p.....
	图标	icon	span、em、i.....
	进度条	progress	div.....
链接、导航	链接、导航	navigator	a
表单	按钮	button	button、input
	单选	radio	input
	多选	checkbox	input
	表单	form	form
	输入框	input	input
	改进表单可用性	label	label
	滚动选择器	picker	select
	开关选择器	switch	input、div.....
操作反馈	底部菜单	action-sheet	div、ul.....
	弹窗	modal	div.....
	toast	toast	div.....
	加载	loading	img、span、div.....
媒体	图片	image	img
	音频	audio	audio
	视频	video	video
地图	地图	map	map
画布	画布	canvas	canvas

View - Component

小程序的组件基于Web Component标准

使用Polymer框架实现Web Component

```
<!-- add style to your element -->
<dom-module id="my-styled-namecard">
  <template>
    <style>
      /* This would be crazy in non webcomponents. */
      span {
        font-weight: bold;
      }
    </style>

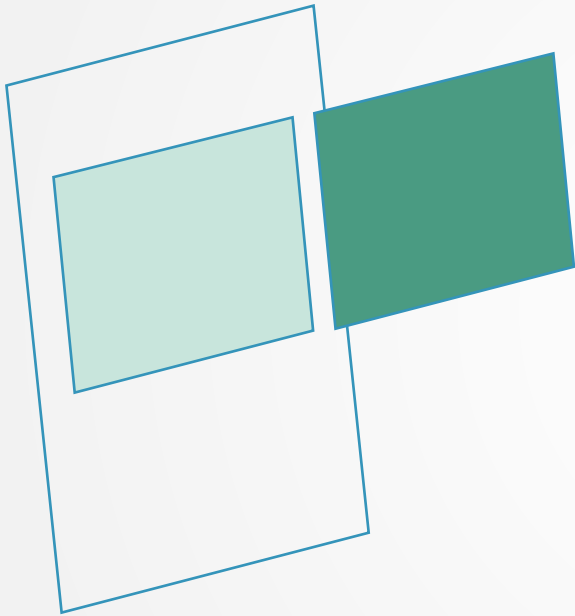
    <div>
      Hi! My name is <span>{{myName}}</span>
    </div>
  </template>

  <script>
    Polymer({
      is: 'my-styled-namecard',
      properties: {
        myName: {
          type: String
        }
      }
    });
  </script>
</dom-module>
```

```
<!-- using the element -->
<my-styled-namecard my-name="Jesse"></my-styled-namecard>
```

Hi! My name is Jesse

View - Native Component



Native组件层在WebView层之上

<canvas / ><video / ><map / ><textarea / >



App Service(逻辑层)

逻辑层将数据进行处理后发送给视图层，同时接受视图层的事件反馈

- 1、App() 小程序的入口；Page() 页面的入口
- 3、提供丰富的 API，如微信用户数据，扫一扫，支付等微信特有功能。
- 4、每个页面有独立的作用域，并提供模块化能力。
- 5、数据绑定、事件分发、生命周期管理、路由管理

运行环境

IOS - JSCore

Android - X5 JS解析器

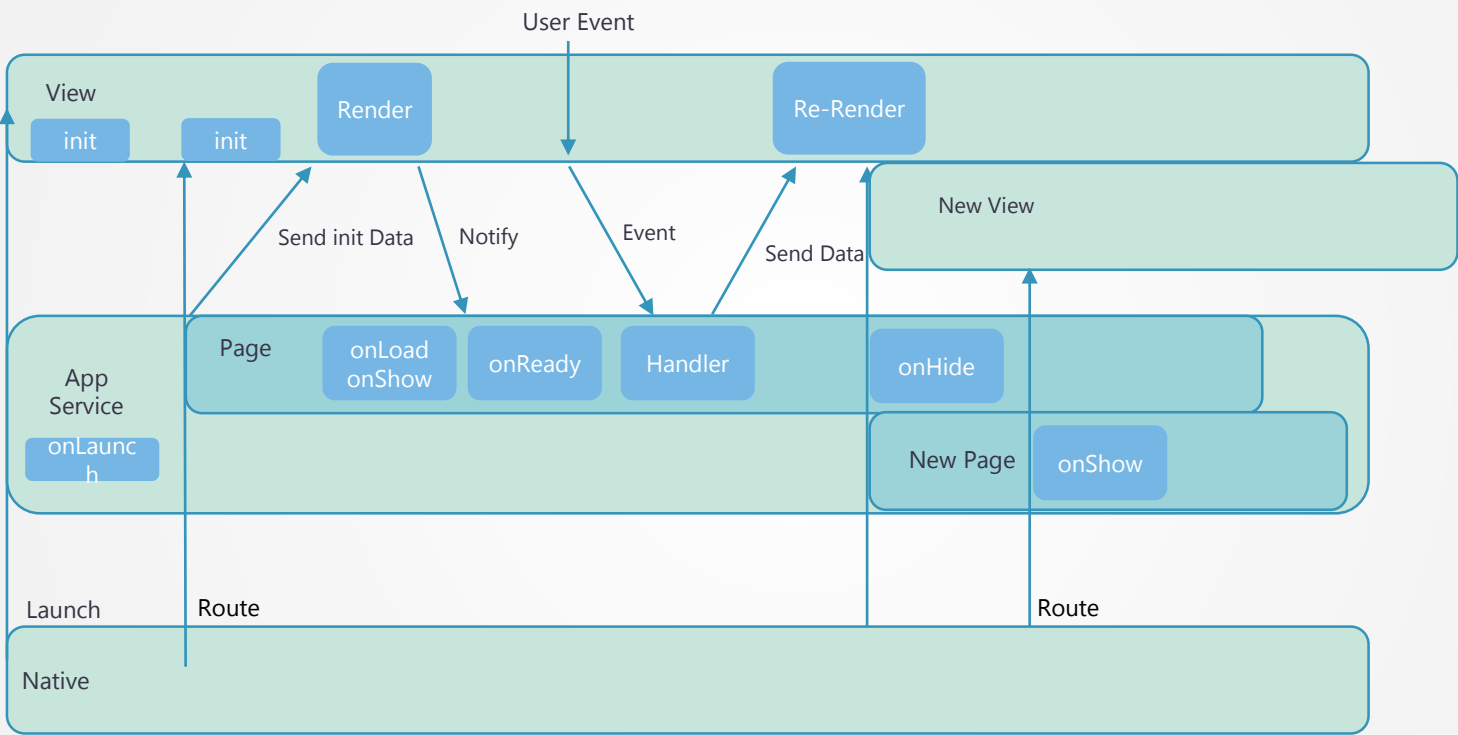
DevTool - nwjs Chrome 内核

App Service - Binding

```
<swiper class="banner-swiper" bindchange="bindSwiperChange" catchtap="goLive" indicator-dots="{{indicatorDots}}"
  autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}">
  <block wx:for="{{banner}}">
    <swiper-item>
      <image src="{{item.pic}}" class="slide-image" width="100%" height="200"/>
    </swiper-item>
  </block>
</swiper>
```

```
var app = getApp()
Page({
  data: {
    banner: [],
    indicatorDots: false,
    autoplay: true,
    interval: 5000,
    duration: 1000
  },
  onLoad: function () { ...
  },
  bindSwiperChange: function(event){ ...
  },
  goLive: function(event){
    var anchorid=event.target.dataset.anchorid;
    wx.navigateTo({
      url: ['/live/index?anchorid=',anchorid].join('')
    })
    return false;
  },
  buildBanner: function(items){ ...
  },
  getLiveHome: function(callback){ ...
  }
})
```

App Service - Life Cycle



App Service - API

API通过JSBridge和Native 进行通信

分类	名称	API
网络	发起请求	request
	上传下载	uploadFile / downloadFile
	WebSocket	connectSocket / onSocketOpen / onSocketError / sendSocketMessage / onSocketMessage / closeSocket / onSocketClose
媒体	图片	chooseImage / previewImage / getImageInfo
	视频	chooseVideo
	音频	playVoice / pauseVoice / stopVoice
	录音	startRecord / stopRecord
	文件	saveFile / getSavedFileList / getSavedFileInfo / removeSavedFile
数据	数据缓存	setStorage / getStorage / removeStorage / clearStorage
位置	获取位置	getLocation
	查看位置	openLocation
设备	网络状态	getNetworkType
	系统信息	getSystemInfo
	重力感应	onAccelerometerChange
	罗盘	onCompassChange
	拨打电话	makePhoneCall
界面	Canvas	drawCanvas / drawCanvas / canvasToTempFilePath
	动画	createAnimation
	导航、设置导航条	navigateTo / redirectTo / navigateBack
	交互反馈	showToast / showModal / showActionSheet
开放接口	登录	login
	用户信息	getUserInfo
	微信支付	requestPayment
	模板消息	

`navigateTo(OBJECT)`

保留当前页面，跳转到应用内的某个页面，使用`navigateBack`可以返回到原页面。页面路径只能是五层

`redirectTo(OBJECT)`

关闭当前页面，跳转到应用内的某个页面。

`navigateBack(OBJECT)`

关闭当前页面，返回上一页面或多级页面。可通过 `getCurrentPages()` 获取当前的页面栈，决定需要返回几层。



小程序开发经验

小程序可以借鉴的优点

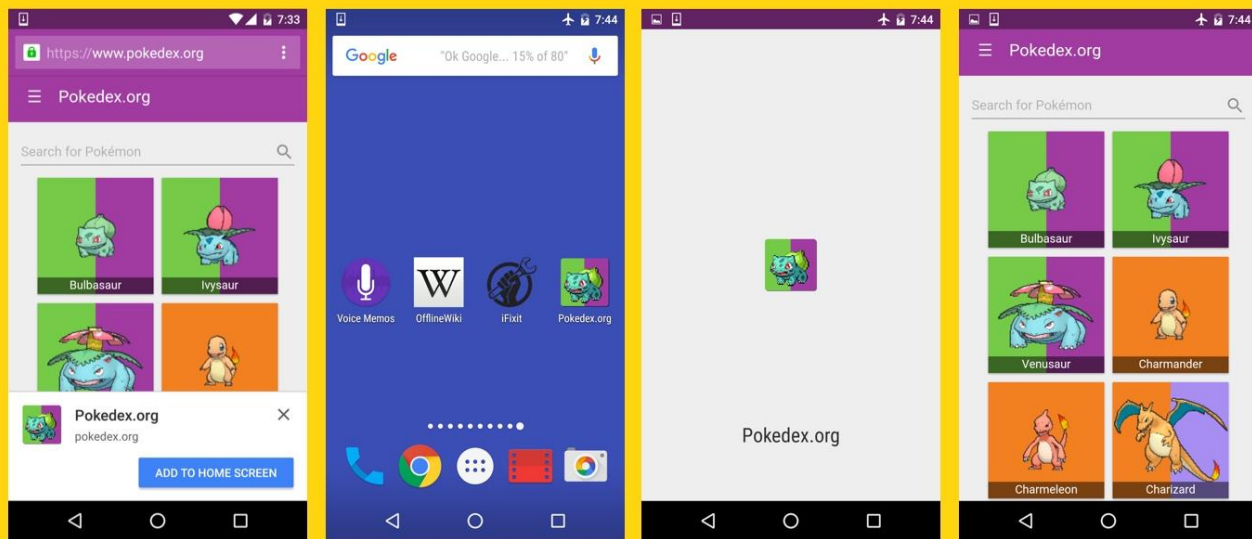
- 1、提前新建WebView，准备新页面渲染。
- 2、View层和逻辑层分离，通过数据驱动，不直接操作DOM。
- 3、使用Virtual DOM，进行局部更新。
- 4、全部使用https，确保传输中安全。
- 5、前端组件化开发。
- 6、加入rpx单位，隔离设备尺寸，方便开发。

小程序存在的问题

- 1、小程序仍然使用WebView渲染，并非原生渲染
- 2、需要独立开发，不能在非微信环境运行。
- 3、开发者不可以扩展新组件。
- 4、服务端接口返回的头无法执行，比如：Set-Cookie。
- 5、依赖浏览器环境的js库不能使用，因为是JSCore执行的，没有window、document对象。
- 6、WXSS中无法使用本地（图片、字体等）。
- 7、WXSS转化成js 而不是css，为了兼容rpx。
- 8、WXSS不支持级联选择器。
- 9、小程序无法打开页面，无法拉起APP。

脱离微信的“小程序”：PWA 渐进式应用

PWA - Progressive Web App



Web App install
banner for engagement

Launch from user's
home screen

Splash screen
(Chrome for Android 47+)

Works offline with
Service Worker

PWA 特点

- **渐进增强** - 支持的新特性的浏览器获得更好的体验，不支持的保持原来的体验。
- **离线访问** - 通过 Service Workers 可以在离线或者网速差的环境下工作。
- **类原生应用** - 使用app shell model做到原生应用般的体验。
- **可安装** - 允许用户保留对他们有用的应用在主屏幕上，不需要通过应用商店。
- **容易分享** - 通过 URL 可以轻松分享应用。
- **持续更新** - 受益于 service worker 的更新进程，应用能够始终保持更新。
- **安全** - 通过 HTTPS 来提供服务来防止网络窥探，保证内容不被篡改。
- **可搜索** - 得益于 W3C manifests 元数据和 service worker 的登记，让搜索引擎能够找到 web 应用。
- **再次访问** - 通过消息推送等特性让用户再次访问变得容易。

PWA - Web App Manifest使Web更像Native

Web App Manifest以JSON的格式定义Web应用的相关配置（应用名称、图标或图像连接、启动URL、自定义特性、启动默认配置、全屏设置等）。

```
{
  "short_name": "Pokedex.org",
  "name": "Pokedex.org",
  "display": "standalone",
  "icons": [
    {
      "src": "img/icon-48.png",
      "sizes": "48x48",
      "type": "image/png"
    },
    {
      "src": "img/icon-96.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "img/icon-144.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "img/icon-196.png",
      "sizes": "196x196",
      "type": "image/png"
    },
    {
      "src": "img/icon-384.png",
      "sizes": "384x384",
      "type": "image/png"
    }
  ],
  "start_url": "index.html?launcher=true",
  "theme_color": "#a040a0",
  "background_color": "#EEEEEE"
}
```

App Manifest

<https://www.pokedex.org/manifest.json>

Identity

Name	Pokedex.org
Short name	Pokedex.org

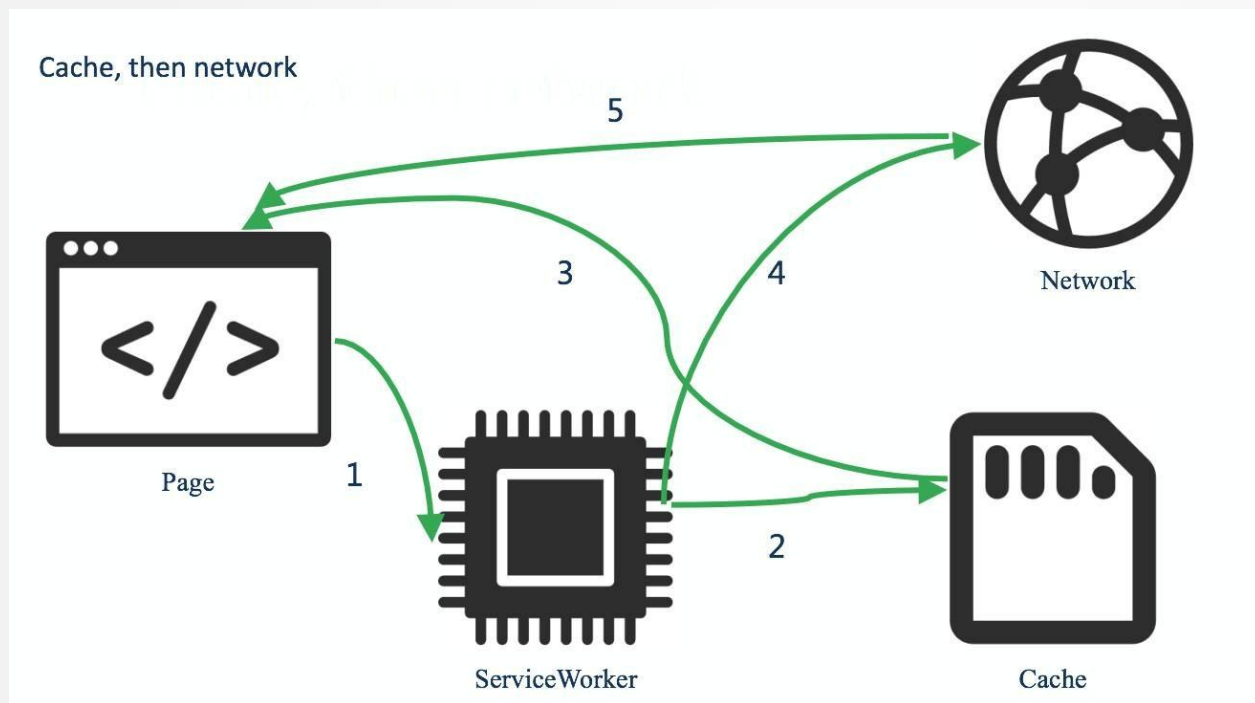
Presentation

Start URL	index.html?launcher=true
Theme color	■ #a040a0
Background color	□ #EEEEEE
Orientation	
Display	standalone

Icons



PWA - Service Workers增强Web能力

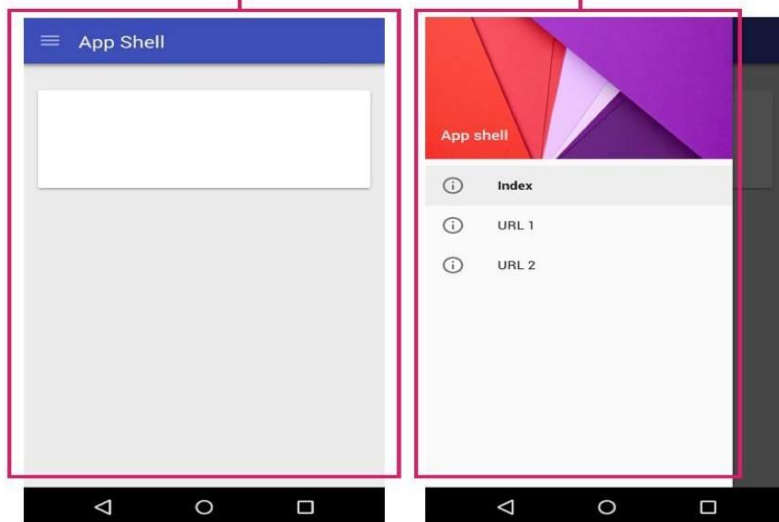


通过Service Works实现资源离线缓存和更新

PWA - App Shell 提升显示效率

App Shell (应用外壳) 是应用的用户界面所需的最基本的 HTML、CSS 和 JavaScript，首次加载后立刻被缓存下来，不需要每次使用时都被下载，而是只异步加载需要的数据，以达到UI保持本地化。

application shell



Cached shell loads **instantly** on repeat visits.

content



Dynamic content then populates the view



THANK YOU