

微信小程序开发初体验

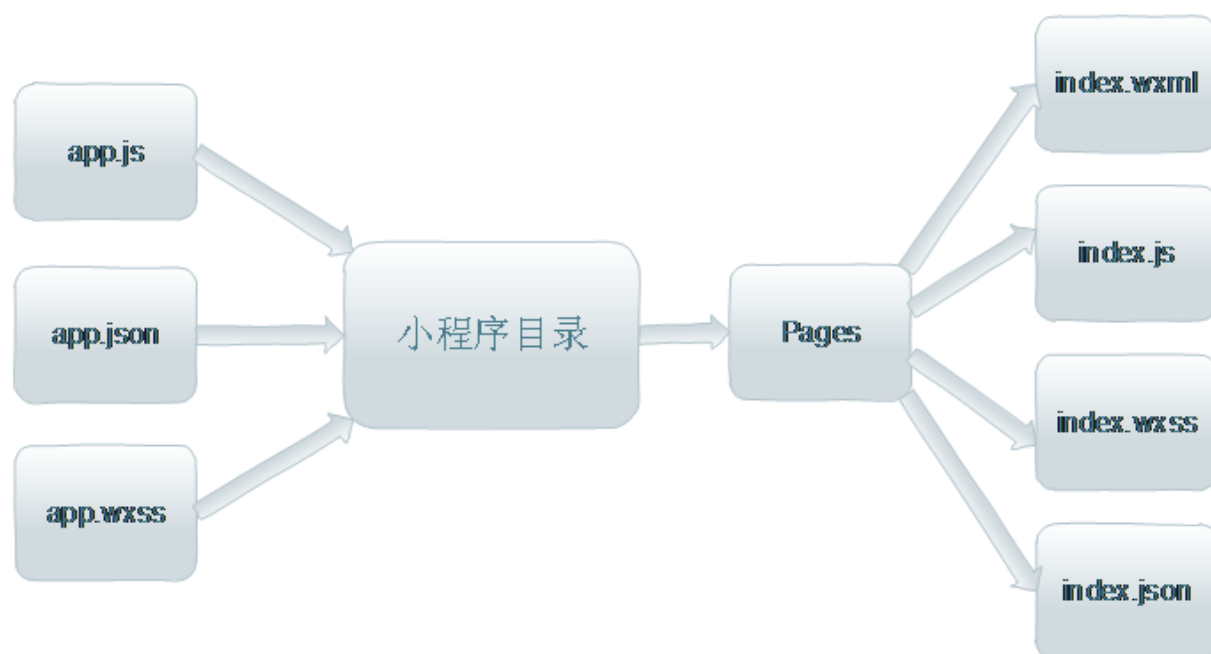
“解放军报融媒体”小程序简介

“解放军报融媒体”微信小程序是解放军报首款微信小程序，由解放军报社技术运营部技术协作处独立设计、开发上线，“解放军报融媒体”微信小程序将《解放军报》纸质报刊进行电子化，内容触手可及，并提供纸媒传统版面及目录列表两种阅读方式，供读者自由选择。作为一款便捷的军事资讯获取工具，它还提供了智能语音读报及视频观看以及重复图片搜索等功能。

最初，我们对小程序的功能要求就是能够将每一版面的纸质报纸“扫描”进手机屏幕，实现纸质报刊的小程序阅览。为此，我们设计了两种呈现方式：沿袭纸媒传统的“版面”呈现方式和一目了然的“目录”呈现方式。

这里特别介绍下“版面”呈现方式。它沿袭了解放军报客户端用户的使用习惯，通过左右滑动小程序，即可切换“手机报纸”的版面；通过点击版面上各文章的热区，可以看到该文的数字报版，即通过了全新数字化排版、符合电子阅读习惯的内容。阅读结束后，通过点击右下角的“返回版面”按钮，即可返回“电子版面”。

微信小程序代码总览



上图是一个最简单的微信小程序目录结构。小程序的代码从控制范围主要分为两部分，全局配置与单个page配置。全局配置主要是三个文件，分别是app.js,app.json和

app.wxss。app.js是小程序的入口文件，每个小程序启动后首先执行的就是该文件。通常情况下，建议在app.js中进行一些基础操作，比如获取屏幕高度、宽度以及缓存数据的处理。app.json文件可对小程序的外观（包含tab键）进行统一配置，可对小程序含有的功能页（page）进行声明。app.wxss是统一样式表，对整个小程序通用的样式进行定义。这里对app.json功能详细介绍如下。

app.json是针对微信小程序的全局配置，主要包含以下几个配置：

pages：页面路径的数组，表示小程序要加载的所有页面，其中数组第一项代表小程序的初始页面。

window：微信原生功能，定制化不强。可设置小程序的状态栏、导航条、标题以及窗口背景色。

tabBar：微信原生功能，定制化不强。适用于常规的Tab应用，Tab栏可置于顶部或底部；tabBar是一个数组，仅支持2-5个tab。

networkTimeout：配置小程序网络请求的超时时间。

debug：调试模式开关，开发模式下建议开启，正式发布之后关闭。

除了上面提到的全局配置，每个页面还可以单独配置page.json，page.json会覆盖app.json中的配置，并只对当前页面生效。

page.json只能对window配置，有两个比较有用的配置项分别是：

enablePullDownRefresh：是否开启下拉刷新。

disableScroll：只能在page.json配置，禁止页面上下滚动。

rpx

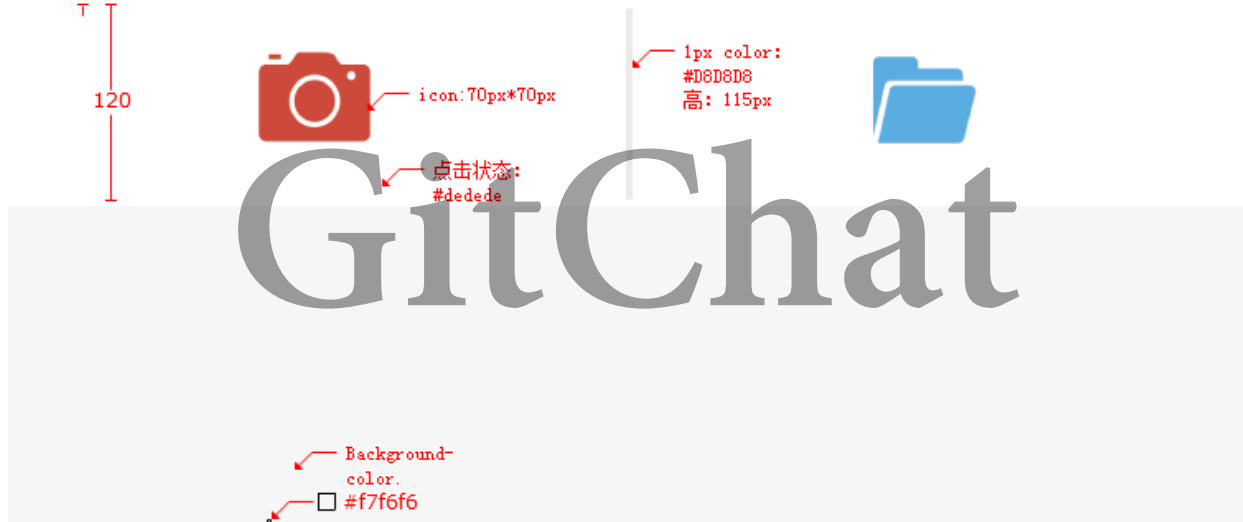
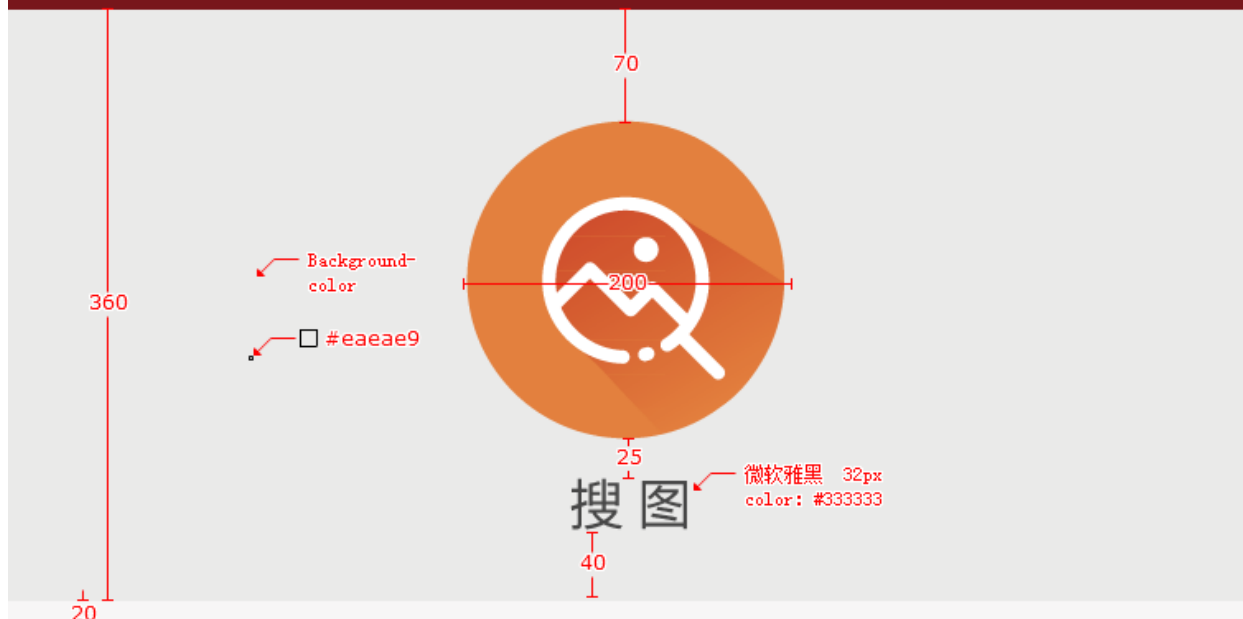
rpx是小程序特有的长度单位，可用于表示宽（高）度、字体大小、边距等，其使用方法类同于px。rpx的主要优势是在一种屏幕尺寸（iphone 6）下进行的设计能够自适应其他屏幕尺寸。建议在设计中能够统一使用rpx。一般建议，设计师依据iphone 6的屏幕尺寸进行设计，给出尺寸大小（实际使用中需要注意的是，PC标定的像素尺寸即rpx值，而手机上显示的尺寸是rpx值得一半）。

flex 布局

严格按照flex布局进行小程序界面的代码实现。简单介绍一下flex布局。flex布局中分容器（container）与项目(item)。容器的属性有display, display-direction, flex-wrap, flex-flow, justify-content, align-items, align-content。item的属性有order, flex-grow, flex-shrink, flex-basis, flex, align-self。我们依据实际中的例子讲解一下flex布局具体用法。



解放军报融媒体



使用说明

拍照：点击相机 - 拍图 - 选择右上角✓ - 选择完成。

上传照片：点击文件夹浏览相册 - 勾选图片 - 完成。

本图库仅收集2014年7月1日至今在《解放军报》上发表的所有图片



从设计稿不同背景色中，可以分为上（header）与下（content）两部分。header可分为logo与logo文字两部分，content可分为menu与文字说明两部分。menu分为相机、分割线与相册三部分。根据上述的文字说明采用flex可以很快的进行实现。需要着重强调的几点是：

1. 小程序的wxss中不支持使用本地图片作为背景图片的CSS属性（background-image），需要在wxml中用image标签代替。
2. flex由于其弹性布局特性能够支持适应不同的屏幕，但在多数布局情况下，若干元素高度是指定的，比如如此设计中的header，而content则需要填满剩下的空间。关键的几个元素及其属性值设计如下：
page{height:100%};container{minheight:100%};content{flex:auto}。

canvas 画图

要在微信小程序中使用canvas，首先要在wxml文件中使用canvas组件，canvas组件相关属性介绍如下：

属性名	类型	默认值	说明
canvas-id	String		canvas 组件的唯一标识符
disable-scroll	Boolean	FALSE	当在 canvas 中移动时，禁止屏幕滚动以及下拉刷新
bindtouchstart	EventHandle		手指触摸动作开始
bindtouchmove	EventHandle		手指触摸后移动
bindtouchend	EventHandle		手指触摸动作结束
bindtouchcancel	EventHandle		手指触摸动作被打断，如来电提醒，弹窗
bindlongtap	EventHandle		手指长按 500ms 之后触发，触发了长按事件后进行移动不会触发屏幕的滚动
binderror	EventHandle		当发生错误时触发 error 事件，detail = {errMsg: 'something wrong'}

微信小程序同时也提供了绘图api接口和大量的canvas对象方法供开发者使用，介绍如下：

context 对象的方法列表	
颜色，样式，阴影	
方法	说明
<u>setFillStyle</u>	设置填充样式
<u>setStrokeStyle</u>	设置线条样式
<u>setShadow</u>	设置阴影
渐变	
<u>createLinearGradient</u>	创建一个线性渐变
<u>createCircularGradient</u>	创建一个圆形渐变
<u>addColorStop</u>	在渐变中的某一点添加一个颜色变化
线条样式	
<u>setLineWidth</u>	设置线条宽度
<u>setLineCap</u>	设置线条端点的样式
<u>setLineJoin</u>	设置两线相交处的样式
<u>setMiterLimit</u>	设置最大倾斜
矩形	
<u>rect</u>	创建一个矩形
<u>fillRect</u>	填充一个矩形
<u>strokeRect</u>	画一个矩形（不填充）
<u>clearRect</u>	在给定的矩形区域内，清除画布上的像素
路径	
<u>fill</u>	对当前路径进行填充
<u>stroke</u>	对当前路径进行描边
<u>beginPath</u>	开始一个路径
<u>closePath</u>	关闭一个路径
<u>moveTo</u>	把路径移动到画布中的指定点，但不创建线条。
<u>lineTo</u>	添加一个新点，然后在画布中创建从该点到最后指定点的线条。
<u>arc</u>	添加一个弧形路径到当前路径，顺时针绘制。
<u>quadraticCurveTo</u>	创建二次方贝塞尔曲线
<u>bezierCurveTo</u>	创建三次方贝塞尔曲线
变形	
<u>scale</u>	对横纵坐标进行缩放
<u>rotate</u>	对坐标轴进行顺时针旋转
<u>translate</u>	对坐标原点进行缩放
文字	
<u>fillText</u>	在画布上绘制被填充的文本
<u>setFontSize</u>	设置字体大小
图片	
<u>drawImage</u>	在画布上绘制图像

微信小程序中没有HTML中的area标签，但是通过canvas组件完全可以实现area标签的所有功能。HTML中，area标签是带有可点击区域的图像映射，一般来讲，具有两个属性，

一个是定义其可点击区域的坐标值coords属性，另一个是定义此区域的目标URL的href属性。

那么，在微信小程序中，首先我们要通过绘图api接口createCanvasContext创建canvas绘图上下文，然后使用lineTo（添加一个新点，然后在画布中创建从该点到最后指定点的线条）将后台服务器传过来的文章区域坐标点集合逐个连接起来，连接到最后一个点之后使用closePath关闭当前路径，最后调用draw进行绘图，这样就在报纸版面图片上为每篇文章画出了该文章的“热区”。

此时，还有一个问题需要解决，那就是如何将用户点击位置和文章区域对应起来，也就是说，如何获知用户点击了哪篇文章的热区？在这里，我们采用[GitHub上某开源项目](#)提供的算法，该算法主要用于判断点是否在多边形区域内。利用该算法，遍历当前版面页的每个文章“热区”，判断用户点击位置是否在该“热区”范围内即可在用户每一次点击后获知用户到底点击了哪篇文章的热区。

如此一来，area标签的“可点击区域的坐标值”coords属性所实现的功能就已经具备了，然后，再结合路由api接口navigateTo，当用户点击后，给该热区指定当前文章ID后就能准确的跳转到当前热区所对应的文章详细内容页面，实现了area标签的“定义此区域目标URL”的href属性。综上，微信小程序中虽然没有提供类似于area标签的“热区”组件供开发者使用，但是，利用canvas组件和navigateTo接口api，我们完美实现了area标签的所有关键功能。

注意事项

1. canvas 组件是由客户端创建的原生组件，它的层级是最高的。
2. 不要在 scroll-view 中使用 canvas 组件。
3. css 动画对 canvas 组件无效。

网络数据传输

微信小程序中的网络数据传输必须是HTTPS协议，而且必须先微信公众平台设置中加入到可信任列表中方能进行正常的网络通讯，包括数据请求、下载、上传和socket通信。网络数据传输的相关api介绍如下：

wx.request(OBJECT)，OBJECT参数说明：

参数名	类型	必填	说明
<u>url</u>	String	是	开发者服务器接口地址
<u>data</u>	Object、String	否	请求的参数
<u>header</u>	Object	否	设置请求的 header，header 中不能设置 <u>Referer</u>
<u>method</u>	String	否	默认为 GET，有效值：OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT
<u>dataType</u>	String	否	默认为 <u>json</u> 。如果设置了 <u>dataType</u> 为 <u>json</u> ，则会尝试对响应的数据做一次 JSON.parse
<u>success</u>	Function	否	收到开发者服务成功返回的回调函数，res = {data: '开发者服务器返回的内容'}
<u>fail</u>	Function	否	接口调用失败的回调函数
<u>complete</u>	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

success返回参数说明：

参数	说明
<u>data</u>	开发者服务器返回的数据
<u>statusCode</u>	开发者服务器返回的状态码

wx.uploadFile(OBJECT)，客户端发起一个 HTTPS POST 请求将本地资源上传到开发者服务器。OBJECT参数说明：

参数名	类型	必填	说明
<u>url</u>	String	是	开发者服务器接口地址
<u>filePath</u>	String	是	要上传文件资源的路径
<u>name</u>	String	是	文件对应的 key，开发者在服务器端通过这个 key 可以获取到文件二进制内容
<u>header</u>	Object	否	设置请求的 header，header 中不能设置 <u>Referer</u>
<u>formData</u>	Object	否	HTTP 请求中其他额外的 form data
<u>success</u>	Function	否	收到开发者服务成功返回的回调函数，res = {data: '开发者服务器返回的内容'}
<u>fail</u>	Function	否	接口调用失败的回调函数
<u>complete</u>	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

success返回参数说明：

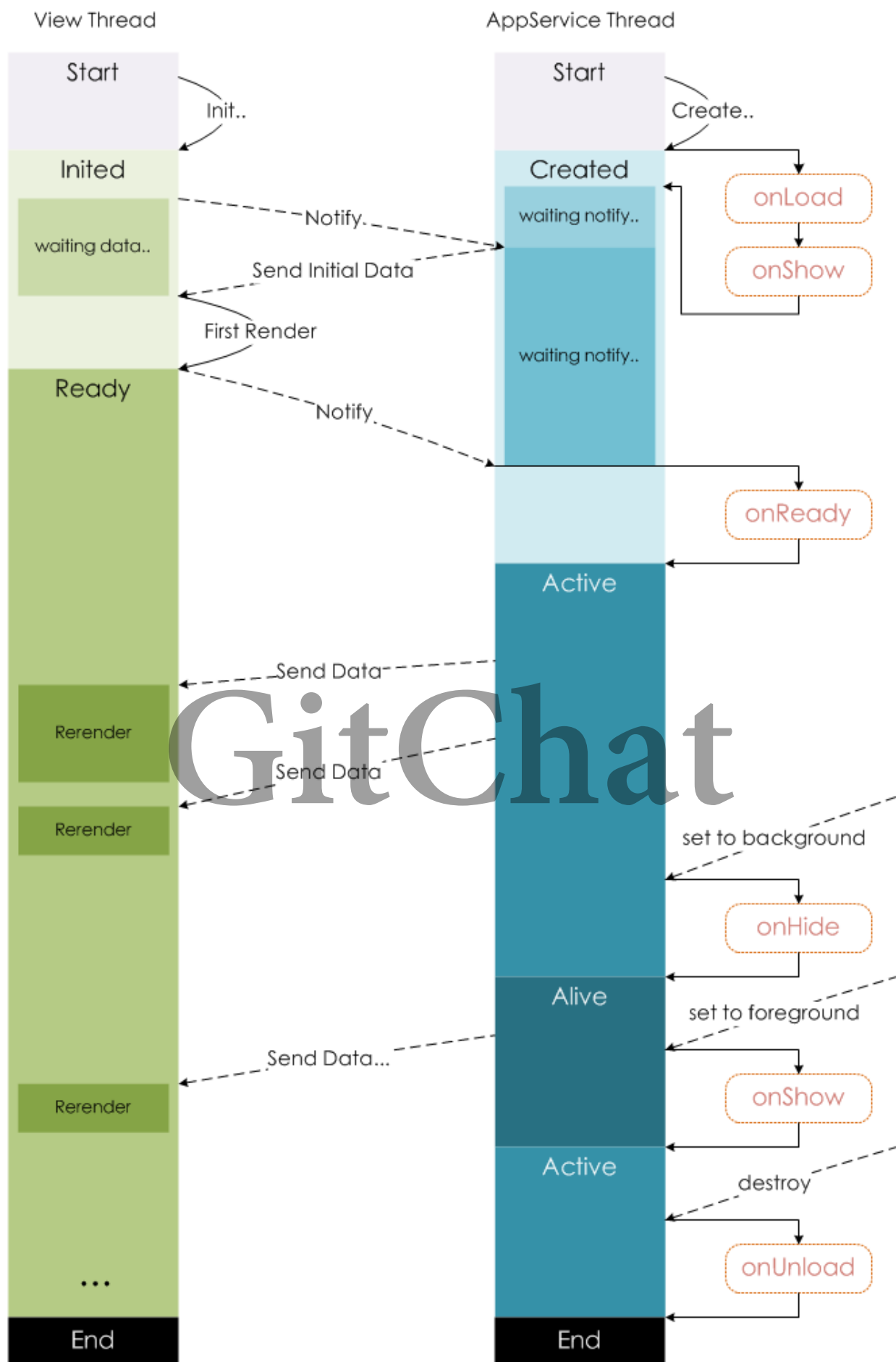
参数	说明
data	开发者服务器返回的数据
statusCode	开发者服务器返回的状态码

wx.downloadFile(OBJECT)，客户端直接发起一个 HTTP GET 请求，返回文件的本地临时路径。OBJECT参数说明：

参数名	类型	必填	说明
url	String	是	开发者服务器接口地址
header	Object	否	设置请求的 header，header 中不能设置 Referer
success	Function	否	下载成功后以 tempFilePath 的形式传给页面，res = {tempFilePath: '文件的临时路径'}
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

为了准确理解微信小程序网络数据传输，首先需要理解小程序生命周期。

GitChat



如上图所示，微信小程序的生命周期函数介绍如下：

函数	描述
<code>onLoad</code>	生命周期函数--监听页面加载，一个页面只会调用一次，接收 <code>wx.navigateTo</code> 和 <code>wx.redirectTo</code> 传递的参数。
<code>onReady</code>	生命周期函数--监听页面初次渲染完成，一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。
<code>onShow</code>	生命周期函数--监听页面显示，每次打开页面都会调用一次。
<code>onHide</code>	生命周期函数--监听页面隐藏，当 <code>wx.navigateTo</code> 到其他页面或点击底部 tab 切换时被调用
<code>onUnload</code>	生命周期函数--监听页面卸载，当 <code>wx.redirectTo</code> 到其他页面或 <code>wx.navigateBack</code> 的时候调用

小程序进行网络数据请求一般在onLoad函数中进行，当然，如果必要也可以在其他非生命周期函数中进行。以《解放军报融媒体》小程序版面页为例，onLoad函数中，首先判断参数options是否定义，若已定义则说明用户是通过好友分享的版面页进入了小程序，此时需要对options进行处理，也就是将分享页面携带的参数作为请求参数以便进行数据请求。若未定义，则直接利用默认赋值参数进行数据请求。

在《解放军报融媒体》小程序设计中，存在网络交互的几个过程分别是：请求版面相关数据、请求目录相关数据、请求文章内容相关数据和相似图片数据。前三个都是request请求，最后一个uploadFile请求。在这里一定要充分理解回调函数的概念，回调函数不是由该函数的实现方直接调用，而是在特定的事件或条件发生时由另外的一方调用的，用于对该事件或条件进行响应。也就是说，success回调函数只有在数据请求成功后才会被执行，因此为了确保我们对所获取数据操作的合法性，必须保证所有对请求到的数据进行操作的代码必须位于success等回调函数内部，而不是其他位置。

注意事项

1. 微信小程序中的网络请求是并发执行的，request 的最大并发数是5，uploadFile和downloadFile的最大并发限制是 10 个。
2. url 中不能有端口。
3. request 的默认超时时间和最大超时时间都是 60s。
4. method 的 value 必须为大写（例如：GET）。
5. image和video组件的src属性也支持http.