开发微信小程序项目遇到的问题以及解决方案

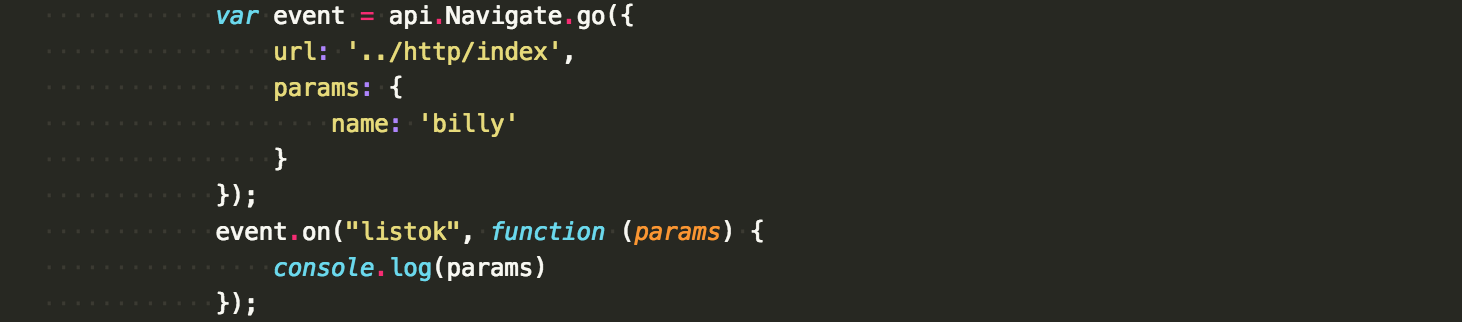
1. 页面间跳转如何监听事件

小程序提供打开新页面、页面重定向、页面返回、tabBar切换四种改变视图的机制，托管了页面的生命周期，并为应用提供了相应的生命周期事件，方便应用各阶段的业务处理，但页面之间相互跳转并没有相应的事件机制，例如——

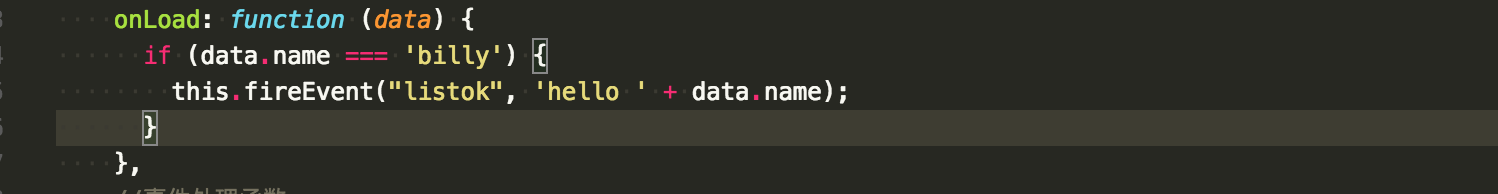
1. A页面打开二级页B，B页面做了一些操作，要通知A页面做相应的处理。
2. 从B页面携带一些数据，返回到A页面

以上两种场景很常见，归结为页面间如何方便的进行交互，当然我们可以通过页面跳转传参或全局数据对象来达到目的，但是使用上有些束缚(转换参数或维护全局对象)，在此elong小程序项目中采用事件机制解决这一问题。我们重写了navigateTo API，A页面调用该接口进行页面跳转后，方法返回给A页面一个事件对象 event，该对象可以注册自定义事件，目标页面(B 页面)可以按照业务需求触发事件响应，同时将相关数据作为参数传递到监听处callback。

A页面



B页面



API



Event



打开二级页面效果图——



1. 前后端同时开发过程中，前端如何方便的mock数据、特殊场景下前端如何方便的调整后端数据

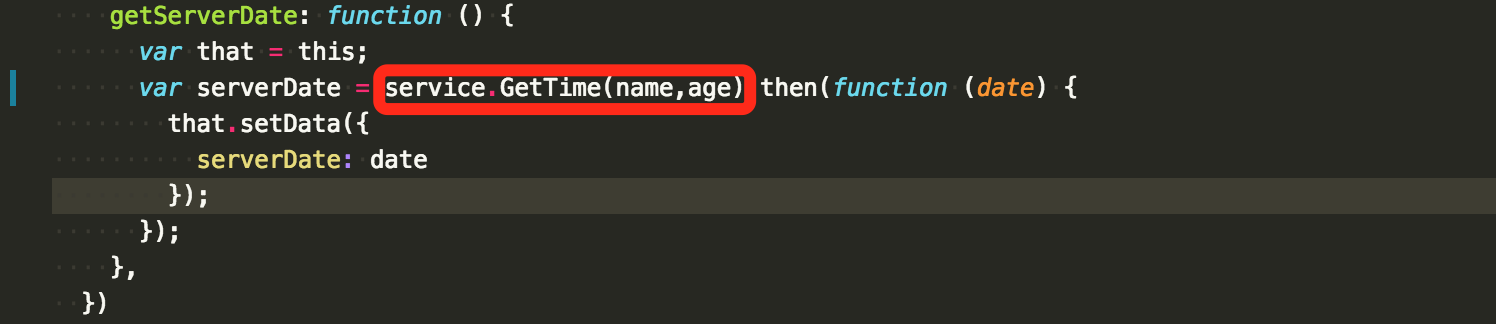
涉及到http(s)接口请求数据，除了上述问题外，其实还有一些hook任务要做，比如统一的 缺省入参、日志、行为统计、超时处理等，这就需要框架层面对小程序原生接口 wx.request做一层封装以便扩展。

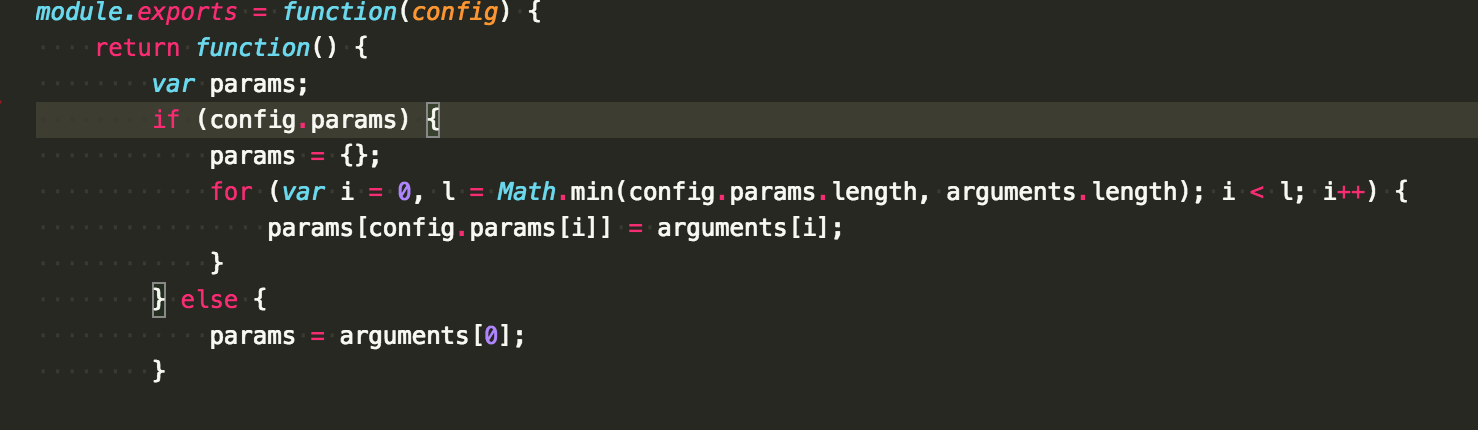
每个页面所需要的后端接口配置在独立service文件中，页面中引用对应的service，统一由此调用http接口，减少页面业务逻辑的复杂度。



url:后端接口url

params：接口入参，若配置了参数列表，则按参数列表自动序列化参数，若不配置，默认取调用接口时的第一个参数作为params





method: 默认为 GET，有效值：OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT

noLoading: 默认false，即每个网络接口均出现loading效果

mockData：模拟后端数据，并行开发过程中必不可少

dataTransform:转换数据，满足特殊性处理

loading效果图——



数据请求——



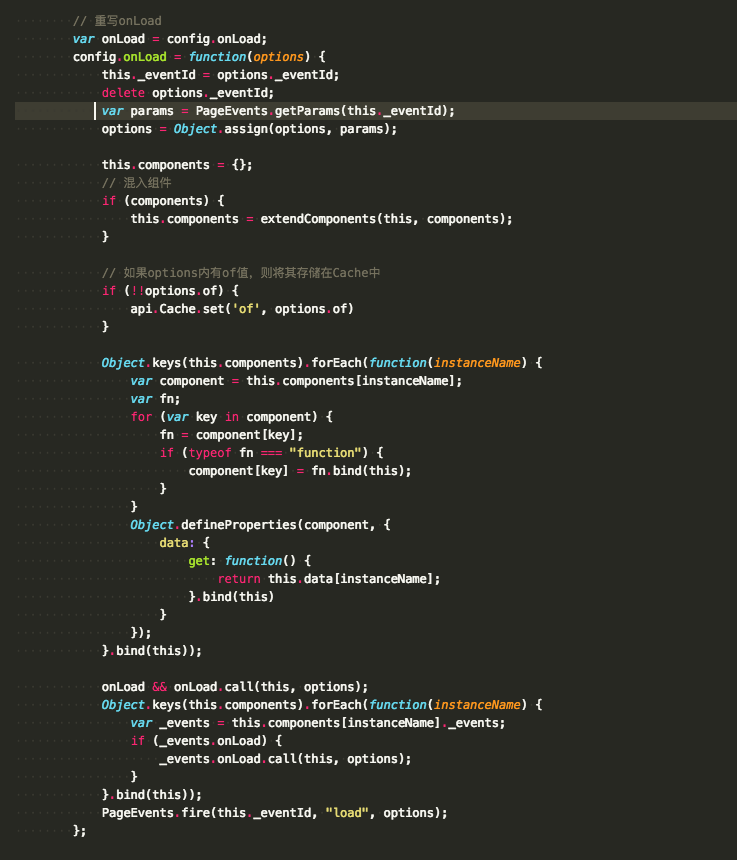
1. 若要所有的页面支持自定义事件、业务组件扩展等功能，又不想所有的页面中添加量大的方法，这就需要框架层提供统一的机制解决，elong小程序框架在此封装了Page、require等原生方法
2. require如果提供基准目录层次\_\_dirname
   1. require非相对目录时会自动补充相等层次
   2. 直接调用原生require
3. 未提供基准目录层次\_dirname，按当前目录寻找

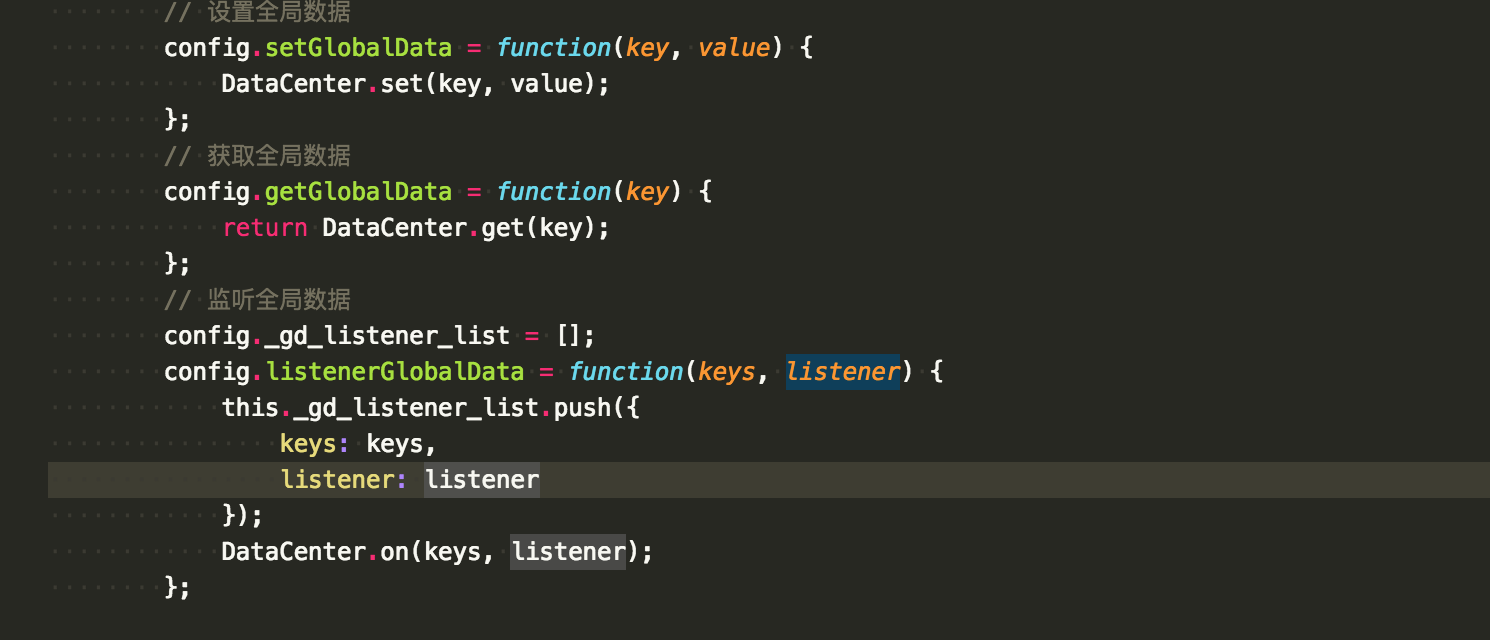


1. Page 构造函数用来注册一个页面。接受一个object参数，其指定页面的初始数据、生命周期函数、事件处理函数等，框架对Page做了重写，这样可以方便的使用扩展能力(页面事件、全局事件、内部组件、外部组件)，使用时仅需将原来的业务代码用包装器包装一下即可

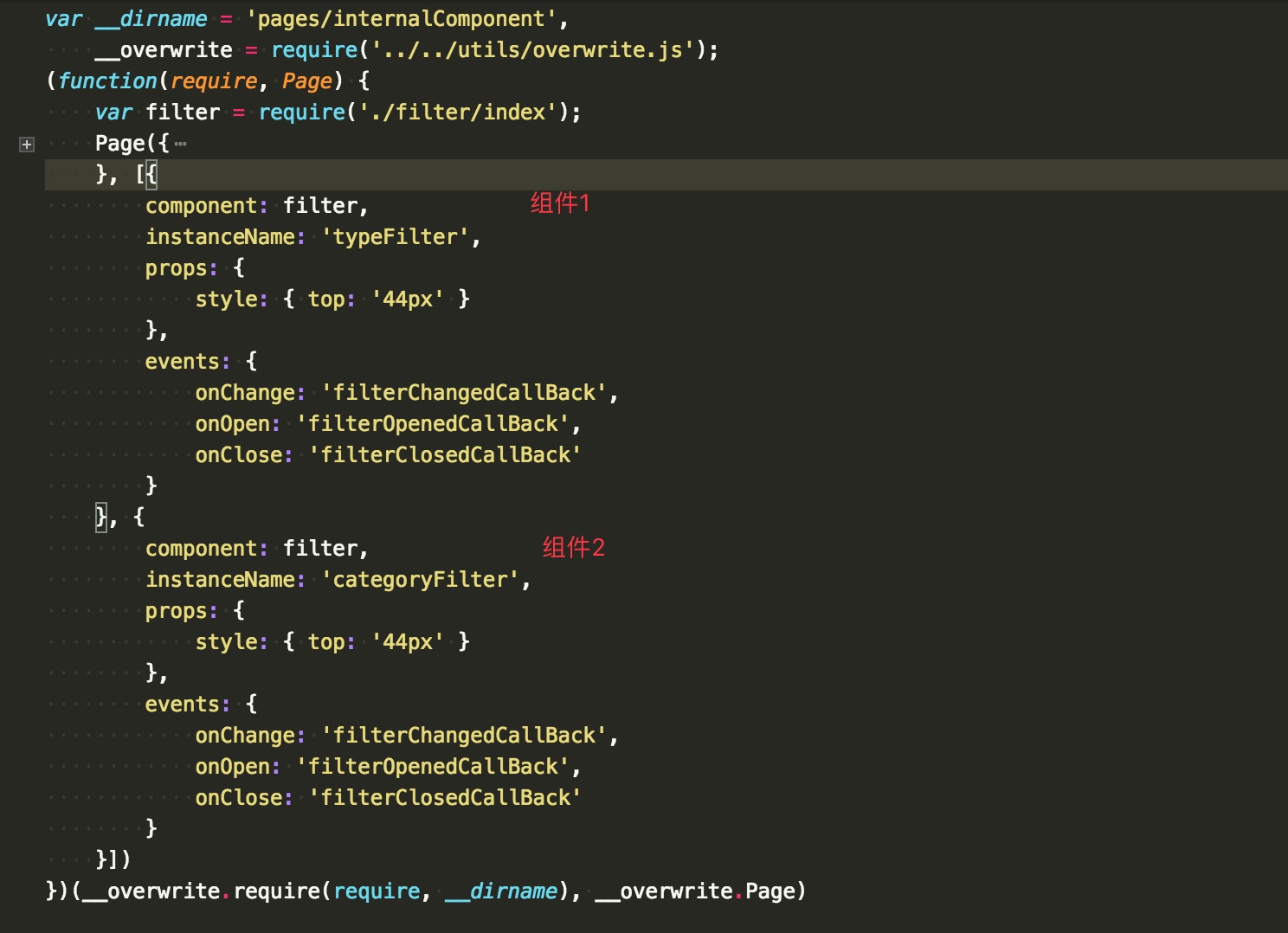


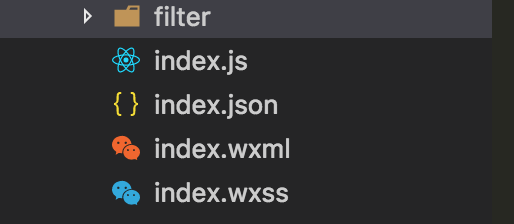
重写后的Page增加了 委托生命周期函数、扩展事件机制、内置内部组件和外部组件扩展等功能(详见overwrite.js 文件)





1. 小程序中提供了模板机制，可以方便的复用一段页面（wxml）代码，但需要手动将模板中绑定的事件、数据注册到当前页面的Page下，而且不可以出现重名属性和方法，相当于仅是一个“插入”操作，目前无法像使用原生组件一样使用一个外部的独立组件。比如，业务中有一个独立的功能日历，我们通常希望这个日历组件是独立一体的，放在一个component文件夹下，日历组件包含自己的wxml、wxss、js，其中js的事件、数据注册不受太多限制（不关心是否和使用者命名冲突等），为此 elong 小程序框架在重写Page时提供了扩展外部组件的功能。





详细使用规范——

1. wxml需要以模板的形式引入组件的wxml

<template is="filter" data="{{ ...typeFilter }}" />

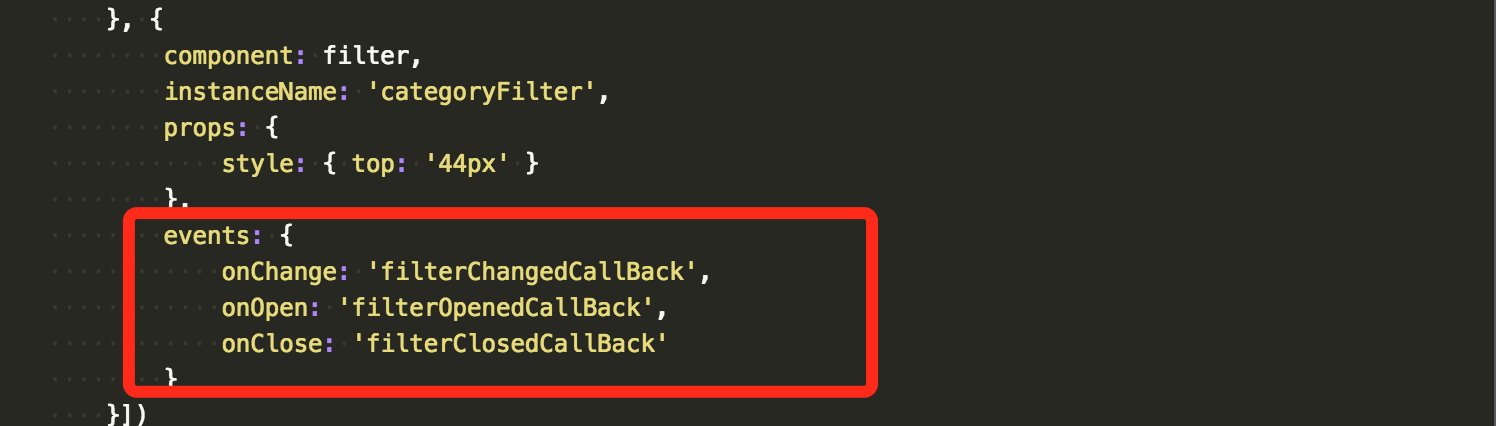
1. wxss 需要引入组件的wxss

@import "filter/index.wxss";

1. js需要引入组件的js文件

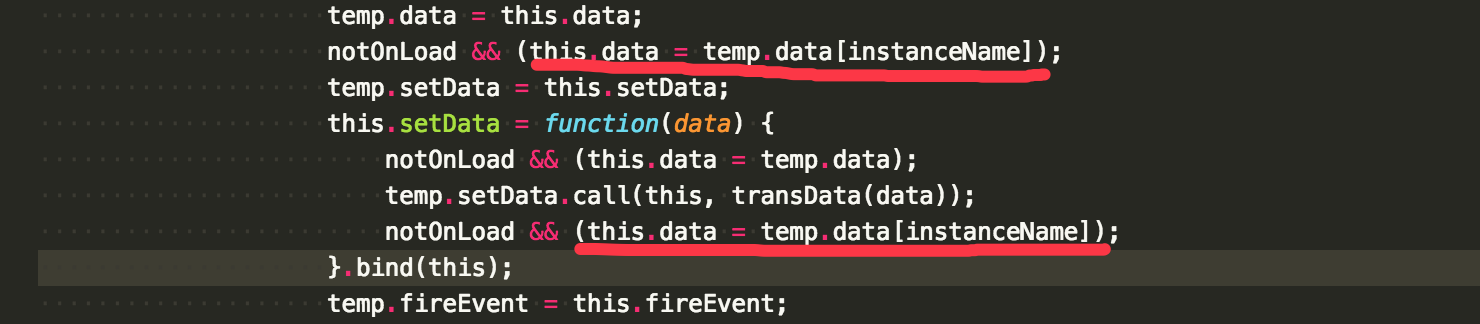
*var* filter = require('./filter/index');

1. 组件和页面见的事件传递需要在组件配置中声明



这些事件函数需要注册在页面中，组件中可以通过fireEvent的形式触发到这些事件。

1. 组件中不可使用Page构造方法，因为组件不是页面，不需要注入小程序页面列表中，其他的事件函数、方法、数据的定义没有任何限制，不需要担心方法和属性的同名冲突，框架层将组件的属性和方法命名空间化赋予页面对象，并且如果组件中的方法用到this，在组件的方法执行时，框架会动态改变要操作的数据对象，保证组件中词法this就是组件的。



\*实现详见 overwrite.js > extendComponents方法。

日历组件——



picker组件效果图——



独立外部组件效果图——

