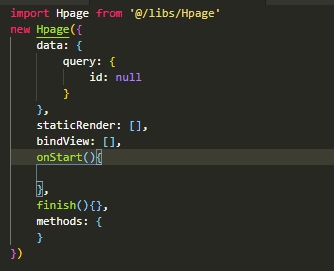
**Hpage文档**

在js内使用import引入Hpage.js即可得到Hpage类，使用new关键字即可创建实例。

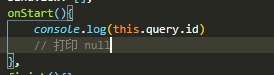
示例：



1. **Data 类型：Object**

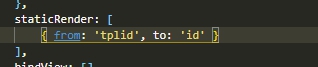
data为Hpage的静态参数对象，data内的元素可被this直接调用

例：



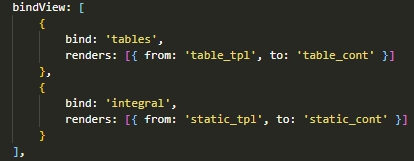
1. **staticRender 类型：Array**

staticRender为静态渲染数组，传入渲染对象一共两个元素，from和to



from : 模板id to: 模板最终被渲染到id容器

1. **bindView 类型：Array**



bindView是data内对象的自动渲染引擎，它可以bind一个data内的对象，当这个对象内的任意值发生改变时可以自动执行render内定义的模板进行数据渲染。

1. **onStart 类型：Function 无参 无返回**

onStart是一个周期函数，它在Hpage加载时执行

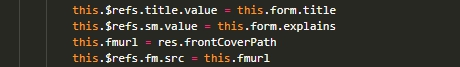
1. **Finish 类型：Function 无参 无返回**

Finish是一个周期函数，它在Hpage加载完成时执行

1. **methods 类型：Object**

methods是一个函数集合，可以在Hpage的作用域内同data一样使用this调用methods内的方法

1. **$refs 类型：Object**



C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml\wpsCFCA.tmp.jpg

$refs内存放的是被ref标记的dom，例如你在input标签上加入ref=”myinput”标记，myinput就会成为$refs内的一个元素使用this.$refs即可找到该dom对象这样操作dom时会更便捷。

1. **toData 类型：Method**

toData方法是为了方便给data对象的参数设置值的方法，一共两个参数，第一个是值，第二个是数据地址

<input ref="occupation" oninput="hpage.toData(this.value, 'query.occupation')">

1. **afterMinx 类型：Method 参数1：name, 参数2：Object**



afterMinx是Hpage解决组件化的一个方案它几乎拥有着一切Hpage所拥有的功能，只是相比Hpage要多出一个beforeMinx的周期函数，要使用afterMinx你需要定义一个模板，在模板内写入常规的html。

afterMinx方法有两个参数，第一个是name，当前组件的名称，html中调用方法时需“hpage.组件名称.组件方法”。

beforeMinx是在组件加载前执行的，通常是写把组件插入 到某个dom的操作。

注意：模板渲染data时不需要加“hpage.组件名称”调用

使用方法：

|  |
| --- |
| C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml\wpsCFDB.tmp.jpg |
|  | C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml\wpsCFDC.tmp.jpg |

Import导入组件，使用loadMinx方法传入组件名称即可加载组件，相比minx组件，它是拥有独立命名空间和作用域，在一些不必要直接渲染的组件可以手动加载，更节约资源。

1. **Components**

**该功能已被弃用**

1. **minx**



minx的写法与Hpage的写法一致，它需要提前把dom写入到主体界面内，minx内的this指向的是本身，但Hpage的this可以调用minx的方法和data。

调用min组件的方法就是引入当前写minx的js它就自动注册到Hpage内了

例如加载网页头部：

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml\wpsCFEE.tmp.jpg

它是需要创建父Hpage的支持的，如果单纯的引入header无效