<http://blog.csdn.net/yuxiaohen/article/details/20864469>

qwebkit为c++和html建起了一个沟通的桥梁,让软件的底层开发和ui的开发解耦,只要底层的api足够稳定强大,那ui的开发就算变化多端都会很轻松了(对于qt c++开发者来说,不用编译了...^ \_ ^),同时可以即时更新ui,或者淘汰旧的版本(更换服务端的html即可),或者增加js统计..等等,万维网之所以有趣,我想很多也来自于提供商有超级强大的控制能力和内容分发能力.而传统的native应用更新之后往往很难分发到用户手上,甚至用户不接受更新,题外话了....混合应用既解决了代码的保密问题,同时解决了内容分发问题,还是很有其读到之处的,比如软件将功能和展示分别同步开发,只要功能完成地够快,指定好api,产品即可即时上线抢占市场,因为ui可以随时更新之后分发到用户的应用.

首先,我们需要搞明白qt和html具体是如何交互的,我们先从主从开始.

我们的设计原则是底层开放足够稳定强大的api,ui层调用或者组合这些api进行作业.ok,主即ui,从即底层,我们先来定一定一个html页面请求底层服务的方式(其实就是qt提供的元对象系统,只是我们让它更好玩一些)

我们在html的meta中加入一个自定义的字段

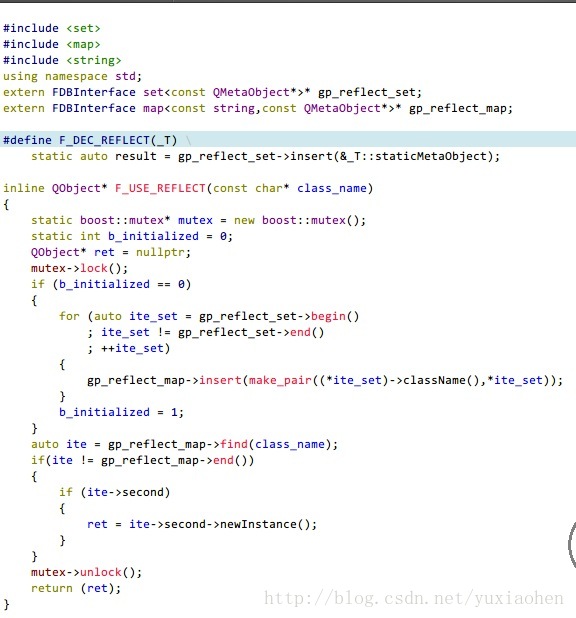


require\_core即是我们增加的meta字段,意思是请求的api的类名(这个类在底层一定是一个QObject的子类,且构造函数为Q\_INVOKABLE,开放的api即槽)

不如我们来看一看CAdbReactor好了



注意红线处,这是个宏,我们也看看



嗯...F\_DEC\_REFLECT和F\_USE\_REFLECT,想到什么了?是的,反射创建对象.

思绪很零散,没事,那就在零散一点吧,我们再来看点东西







ok,让我们试着把这些东西串在一起思考一下:

一个html的require\_core字段

一个WizardHTMLEE

一个反射

我们让页面载入的完成的信号连到WizardHTMLEE的PageLoadFinished槽,让WizardHTMLEE解析require\_core的字段,然后使用反射创建相应的api服务对象,在html页面中即可调用这些api了,当然,我们需要等待api服务对象创建完成



ok,这只是一个简单的模型,实际上我们可以让一个WizardHTMLEE支持多个webview和多个require\_core的能力(实际上我已经做了一个混合html和qml的反射引擎),当然,能看懂这篇文章的话我想这些都只是个体力活了

版权声明：本文虽然是博主原创文章，但可以随意转载，不过请注明出处，谢谢