

昵称：孤竹君

园龄：6年11个月

粉丝：35

关注：0

+加关注

< 2016年3月 >						
日	一	二	三	四	五	六
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

随笔分类(106)

ACE(11)

ATL/WTL(6)

C++(14)

chromium(10)

COM(2)

IE Program(7)

Miscellaneous(4)

Net Program(2)

Shell(2)

SQL Server(2)

VC++(29)

WebRTC(6)

代码封装(5)

设计模式(2)

心情日记(4)

随笔档案(85)

2015年4月 (2)

2015年3月 (4)

2014年9月 (1)

2014年8月 (1)

2014年6月 (4)

2013年11月 (1)

2013年5月 (1)

2012年11月 (1)

2010年11月 (1)

2010年10月 (1)

2010年5月 (1)

2010年4月 (1)

2010年3月 (1)

2009年12月 (2)

2009年11月 (2)

2009年10月 (2)

2009年8月 (6)

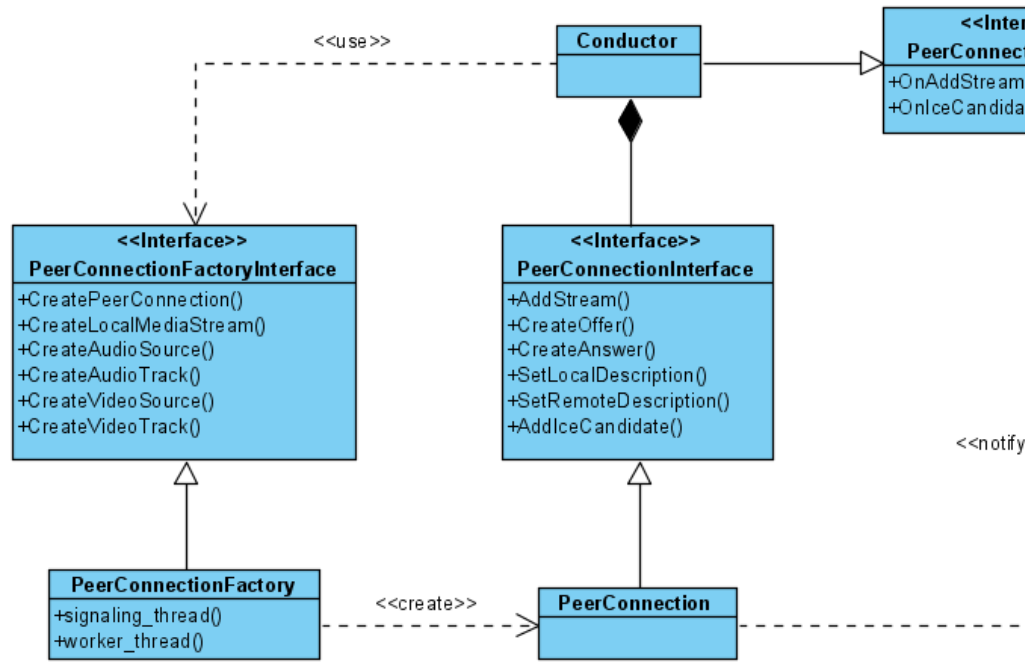
2009年7月 (8)

WebRTC手记之框架与接口

转载请注明出处：<http://www.cnblogs.com/fangkm/p/4370492.html>

上一篇文章简单地介绍了下WebRTC的协议流程，这一篇就开始介绍框架与接口。

一提到框架，本能地不知道从什么地方入手了。曾经直接从Chromium项目对WebRTC的源码的集成方面入手，后来发现这个步子迈的太大了，看的越多，概念越混乱，看了半个月感觉也没啥沉淀。还是从WebRTC提供的示例工程peerconnection_client入手比较轻便。先抛开音视频流的构建和渲染流程，示例工程核心的代码结构如下：



从面向对象的视角来看，WebRTC的设计还是非常棒的，真正地做到了接口编程的概念，对WebRTC功能的使用都通过接口来进行，这样最大程度上保证了WebRTC模块的可定制性，这样就可以让WebRTC更多地回归到描述协议的本质。如果WebRTC对这些接口的实现不能满足你的业务需求，理论上你可以提供自己的实现逻辑。本图中的PeerConnectionFactoryInterface和PeerConnectionInterface没有这种定制的代表性，因为重新提供它们的实现逻辑的需求场景基本上不存在（即便不用重写，但也支持参数的定制，具体请参见CreatePeerConnectionFactory的重载方法）。但是音视频相关的接口定制的场景就很普遍了，比如Chromium浏览器集成WebRTC，但是音视频采集需要走Chromium自己的音视频模块，所以Chromium对WebRTC音视频的采集接口重新做了实现适配，以后有机会肯定非常乐意分享下Chromium源码对WebRTC的集成，不过那也是在WebRTC熟悉完之后的工作了。

图中Conductor是该示例工程提供的核心业务类，整个WebRTC的使用都浓缩在这个类中。Conductor通过CreatePeerConnectionFactory方法创建了一个PeerConnectionFactoryInterface接口的实现对象，通过这个接口，可以创建关键的PeerConnectionInterface接口，PeerConnectionInterface接口是WebRTC的协议核心。此外，PeerConnectionFactoryInterface接口还提供了创建本地音视频流的功能接口，这个部分稍后再述。根据图中PeerConnectionInterface接口的成员方法可以看出，WebRTC通信流程的交互接口基本上都在这里了，给Conductor的回调通知是通过PeerConnectionObserver接口来完成。具体的交互流程请参见上一篇博文。

口供AudioTracks使用，但是从源码中理解，这是个伪概念，因为没有提供一个类似于VideoCapturer的AudioCapturer接口，这里没有音频的采集逻辑，实际上WebRTC的音频采集接口使用的是AudioDeviceModule，在创建PeerConnectionFactory的时候可以由外界定制，如果没有，则内部创建AudioDeviceModuleImpl来实现此接口完成音频设备的采集工作。可能是功力不够，反正我是不太理解音频采集和视频采集这种设计的不对称性。如果也封装一个AudioCapturer接口的概念，这样可定制性是不是可以更高。

构建媒体流的过程基本上就是构建Video Track和Audio Track，并将其添加到Media Stream里。在peerconnection_client工程中，Conductor依赖DeviceManagerInterface接口的CreateVideoCapturer方法创建一个当前可用的视频设备采集对象VideoCapturer，将它作为视频采集源中的数据来源（通过挂接VideoCapturer的SignalVideoFrame信号来接收视频数据），此外MainWnd还创建了一个内部类VideoRenderer从VideoRendererInterface接口派生，并将其添加到Video Track中， VideoRenderer的实现就是将接收到的视频帧数据渲染到窗口上。

下一篇开始分析WebRTC原生的音视频本地采集模块。

分类: [WebRTC](#)

好文要顶

关注我

收藏该文

孤竹君

关注 - 0

粉丝 - 35

+加关注

40

(请您对文章做出评价)

« 上一篇: [WebRTC手记之初探](#)
» 下一篇: [WebRTC手记之本地视频采集](#)

posted @ 2015-03-27 00:09 孤竹君 阅读(2886) 评论(9) 编辑 收藏

评论列表

- #1楼 2015-03-27 10:43 KiliU

如果不下载chromium的源代码，webRTC的代码只有100M吗？
能问下您学习webRTC是为了啥？感觉这东西做起来只能定制自己的视频通话了

支持(0) 反对(0)
- #2楼[楼主] 2015-03-27 11:32 孤竹君

@KiliU
纯的webrtc代码也差不多200M吧，这东西就是个宝库，可学习的东西多着呢。

支持(0) 反对(0)
- #3楼 2015-07-31 10:56 narutoinfo

楼主，UML图是自己画的，还是逆向的？

支持(0) 反对(0)
- #4楼[楼主] 2015-07-31 14:12 孤竹君

@narutoinfo
好吧，我承认画的很挫。

支持(0) 反对(0)
- #5楼 2015-07-31 15:30 narutoinfo

@孤竹君
囧，没那个意思，只是自己在找逆向的工具

支持(0) 反对(0)
- #6楼 2015-09-24 12:58 clareying

请问能发份源码出来吗

支持(0) 反对(0)