



This repository Search

Pull requestsIssuesGist



+▼



ChenChenJoke / JokerChrome

Watch▼4

★Star41

🔗Fork12

<>Code

!Issues1

🔗Pull requests0

📖Wiki

📈Pulse

📊Graphs

The chrome produce and the "chrome://about"

28 commits

1 branch

0 releases

0 contributors

Branch: master▼

New pull request


New file


Upload files

Find file

HTTPS▼


https://github.com/ChenCh










Download ZIP

chenchenjoke last eitLatest commit 9d418a7 on 2 Jul 2015

Dismiss



File uploading is now available

You can now drag and drop files into your repositories. [Learn more](#)

images	'fad'	8 months ago
README.md	last eit	8 months ago
class = toe	fdsafdsa	8 months ago

README.md

作为一个前端工程师你了解你的小伙伴么--chrome

chrome的常用功能我就不说了，F12,element,network,timeline,resources,aduits,console这些我就默认大家都已经掌握了。

估计很多人不知道Chrome地址栏功能，我作为一个low到家的前端工程师，必须懂的。以下我要介绍的这些指令在Chrome地址栏输入即可

about:version – 显示当前版本 也可以是chrome-resource://about/

about:plugins – 显示已安装插件

about:histograms – 显示历史记录

chrome://history2 – 浏览历史 History2

about:dns – 显示DNS状态

about:cache, 重定向到 view-cache: 显示缓存页面

view-cache:stats – 缓存状态

about:stats – 显示状态

about:network – 很酷的网络工具

about:internets – 这应该算是一个彩蛋

chrome-resource://new-tab/ – 新标签页

about:memory – 可以查看内存和进程占用。也可以Shift+ESC，点击Statistics for nerds（傻瓜统计信息）

about:flags – Chrome高级设置

事实上你根本就无需记这些指令，你只需要记住一个万能的就行，那就是chrome://about/

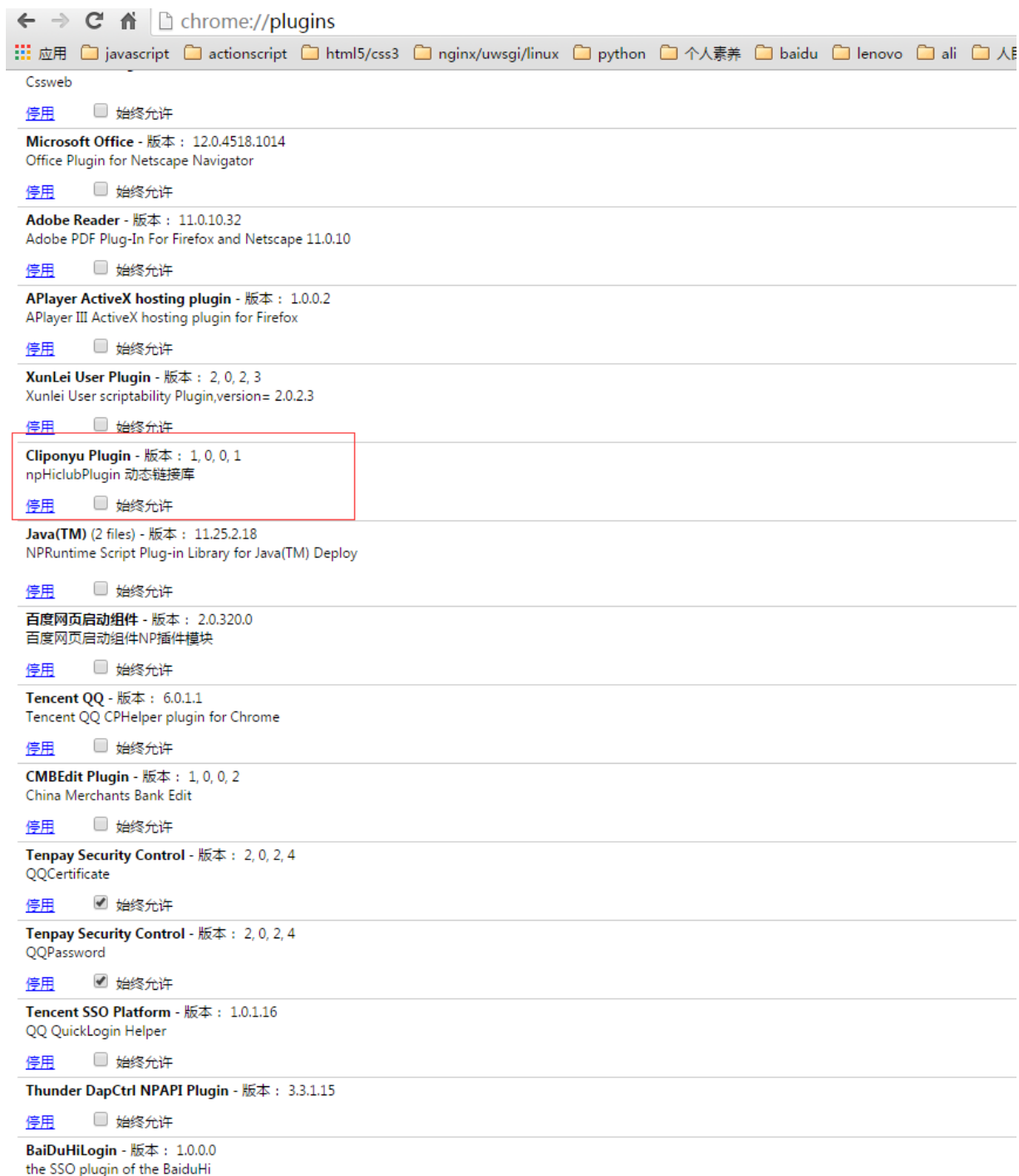
你在Chrome地址栏输入这个，就会出现如下截图，哈哈，万能吧！（还有好多，我只截了一部分哦）

List of Chrome URLs

- <chrome://accessibility>
- <chrome://appcache-internals>
- <chrome://apps>
- <chrome://blob-internals>
- <chrome://bookmarks>
- <chrome://cache>
- <chrome://chrome>
- <chrome://chrome-urls>
- <chrome://components>
- <chrome://conflicts>
- <chrome://crashes>
- <chrome://credits>
- <chrome://devices>
- <chrome://dns>
- <chrome://downloads>
- <chrome://extensions>
- <chrome://flags>
- <chrome://flash>
- <chrome://gcm-internals>
- <chrome://gpu>
- <chrome://help>
- <chrome://histograms>
- <chrome://history>
- <chrome://indexeddb-internals>
- <chrome://inspect>
- <chrome://invalidations>
- <chrome://ipc>
- <chrome://media-internals>
- <chrome://memory>
- <chrome://memory-internals>
- <chrome://nacl>
- <chrome://net-internals>
- <chrome://newtab>
- <chrome://omnibox>
- <chrome://password-manager-internals>
- <chrome://plugins>
- <chrome://policy>
- <chrome://predictors>
- <chrome://print>
- <chrome://profiler>
- <chrome://quota-internals>
- <chrome://serviceworker-internals>
- <chrome://settings>

插件管理,chrome://plugin

这个查看的就是我们的chrome里内置的插件，比如比较长用的有各种银行安全登录插件，迅雷的插件（点你点击一个，下载地址的时候会用迅雷打开下载这个资源，注意：不是下载完种子点击种子，这有本质的区别）。另外，我们再插件里面也看到了我们的直播助手。如下图：



就是靠它，我们才能通过浏览器的js引擎，调用np的动态库，然后客户端的同学，获取我们调用的np插件时候传入的参数来做一些比如挂起直播助手或微端的操作。

注意：chrome官方通在chrome42版本以及更高版本，默认关闭np插件，现在微端同学给出的临时解决反感，就是更改开关，但是这个不会是长久之计，不就之后估计chrome就要完全停用np插件了。通知如下图：

https://support.google.com/chrome/answer/6213033?hl=en

Known issues > Plugin issues

NPAPI plugins don't work on Chrome version 42 and higher

Plugins allow extra features to be added to browsers. For example, you can watch certain types of videos or play a game that's on a webpage.

NPAPI support has ended

In the past, many plugins were developed using an older system called NPAPI. Today fewer sites are using NPAPI plugins and they can sometimes cause security risks on websites.

To make browsing with Chrome safer, faster, and more stable, we're [ending support for NPAPI plugins](#) with Chrome version 42.

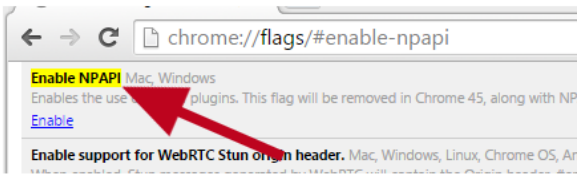
Which plugins are supported

Plugins that use a newer, more secure system called Pepper API (PPAPI) will continue to work, including those that ship with Chrome, like Adobe Flash and PDF Viewer. However, plugins that use NPAPI, including Silverlight, Java, and Unity, won't work.

How to temporarily enable NPAPI plugins

If you must use a NPAPI plugin, there's a temporary workaround that will work until Chrome version 45 is released later in 2015:

1. Open Chrome.
2. In the address bar at the top of the screen, type `chrome://flags/#enable-npapi`
3. In the window that opens, click the link that says **Enable** under the **Enable NPAPI** flag:



4. In the bottom left corner of the page, click the **Relaunch Now** button.

After the release of Chrome version 45, you'll need to use an alternate web browser to load content that requires a NPAPI plugin

那么我们的浏览器页面是如何调用我们的桌面程序的呢？

IE

我来解释一下他们的工作原理。先拿IE举例，首先，我们在页面上创建了一个名字叫做ActiveXObject的插件，大家都知道IE浏览器是可以通过

这个插件对磁盘文件进行操作（桌面程序对ActiveXObject开放了用户级别的权限）。知道我们的应用程序继承了一个windows系统的类，

并且声明自己是一个合法的插件，如下图：

```
14
15 //·CLiveShow
16 using namespace std;
17 class ATL_NO_VTABLE CLiveShow :
18     ... public CComObjectRootEx<CComSingleThreadModel>,
19     ... public CComCoClass<CLiveShow, &CLSID_LiveShow>,
20     ... public IDispatchImpl<CLiveShow, &IID_ILiveShow, &LIBID_HCPluginLib, /*wMajor=*/1, /*wMinor=*/0>,
21     ... public IObjectSafetyImpl<CLiveShow, INTERFACESAFE_FOR_UNTRUSTED_CALLER>
22 {
```

只要实现IObjectSafetyImpl这个接口就可以了（此方法只适用于IE9之前，在IE9之后需要实现另一个接口，并且参数也变多了）

所以我们可以直接通过ActiveXObject启动某一个可执行文件。因为我们的应用程序是被操作系统认为是安全的，可以被调用的。

首先我想问一下非IE浏览器能不能操作本地资源？

接下来我们说一下chrome,这个可能会比较复杂，首先我们的应用你程序要声明一个动态库

Block Header	英语(美国) (040904e4)
Comments	
CompanyName	
FileDescription	npHiclubPlugin 动态链接库
FileVersion	1, 0, 0, 1
InternalName	npHiclubPlugin
LegalCopyright	Copyright (C) 2013
LegalTrademarks	
MIMETYPE	application/npHiclubPlugin
OriginalFilename	npHiclubPlugin.dll
PrivateBuild	
ProductName	Cliponyu Plugin
ProductVersion	1, 0, 0, 1
SpecialBuild	

这个就是我们应用程序的动态库信息，他会被写入注册表，方便我们像用他的时候能够找到他。

然后我们安装完直播助手时候还会生成另外的文件两个dll,注册表里也包含这部分路径信息。

CliponyuPlugin.dll	2015/3/31 21:00	应用程序扩展	90 KB
CliponyuPlugin.pdb	2015/3/31 21:00	Program Debug...	3,267 KB
npCliponyuPlugin.dll	2015/3/31 21:00	应用程序扩展	80 KB
npCliponyuPlugin.pdb	2015/3/31 21:00	Program Debug...	1,771 KB

OK现在我们需要再chrome页面注册一个我们动态库里声明的插件了。

```
<!--主播npHiclubPlugin -for chrome-->
<embed width="0" height="0" type="application/npHiclubPlugin" id="npHiclubPlugin">
```

当页面初始化之后，我们在页面实例化了一个np插件。目前我们还不能调用我们的连麦，直播等等方法，我们先需要启动我们的动态库，我们调用了np插件里面我们动态库的启动指令，然后np插件跟着这条指令查找到我们应用程序的注册表，在注册表里，我们查找到了应用程序下的.dll的文件，然后dll文件开启一个线程，启动我们的.exe文件。然后dll自动销毁。

这样我们的直播助手就启动了起来（这里主要np插件大多数的调用桌面程序的方法都是这样，由于安全级别的问题，最开始浏览器是 没有吧直接启动某个可执行文件的权限放在np插件中，如果你安装了迅雷的话，可以查看一下迅雷是有两个插件来启动桌面程序。一个是np插件，另一个是通过extension的方式的迅雷迷你版）。至此我们的页面就可以跟桌面插件进行单项交互了。

地址预测chrome://predictors/

通过 Omnibox 优化与用户的交互，引入 Omnibox 是 Chrome 的一项创新，并不是简单地处理目标的 URL。除了记录之前访问过的页面 URL，它还与搜索引擎的整合，并且支持在历史记录中进行全文搜索(比如，直接输入页面名称)。

当用户输入时，Omnibox 自动发起一个行为，要么查找浏览记录中的 URL, 要么进行一次搜索。每一次发起的操作都会被加以评分，以统计它的性能。你可以在 Chrome 输入 chrome://predictors 来观察这些数据。由于图比较长，所以我切成了两个图让大家看的清楚些

b	http://icms.baidu.com:8080/		
b	http://tool.css-js.com/compressor.html?b		
b	http://www.baidu.com/		
b	http://www.baidu.com/#ie=UTF-8&wd=b		
b\	http://erp.baidu.com/		
ba	http://br.hao123.com/animalprofeta?from=predictor_logo_topic_br		
ba	http://dp.baidu.com/pms/other#cate=browser&productId=10&pageType=pc&pageId=777		
ba	http://erp.baidu.com/		
ba	http://icms.baidu.com:8080/		
ba	http://tool.css-js.com/compressor.html?b		
ba	http://www.baidu.com/		
ba	http://www.baidu.com/#ie=UTF-8&wd=ba		
bai	http://dp.baidu.com/pms/other#cate=browser&productId=10&pageType=pc&pageId=777		
bai	http://erp.baidu.com/		
bai	http://icms.baidu.com:8080/		
bai	http://www.baidu.com/		
baid	http://erp.baidu.com/		
baid	http://www.baidu.com/		

	0	771	0
	0	5	0
	3337	314	0.9139961654341276
	0	5	0
	0	1	0
	0	20	0
	0	453	0
	0	1522	0
	0	691	0
	0	3	0
	1513	12	0.9921311475409836
	0	5	0
	0	2	0
	0	12	0
	0	2	0
	11	1	0.9166666666666666
	0	2	0

Chrome 维护着一个历史记录，内容包括用户输入的前置文字，采用的行为，以命中的资数。在上面的列表，你可以看到，当输入**b**时，有 91% 的机会尝试打开 **baidu**。如果再补充一个 **a** (就是 **ba**)，打开 **www.baidu.com** 的可能性增加到 99.2%。

那么网络模块会做什么呢？上表中的黄色和绿色对于 **ResourceDispatcher** 非常重要。如果有一个一般可能性的页面(黄色)，**Chrome** 就是发起 **DNS** 预解析。如果有一个高可能性的页面(绿色)，**Chrome** 还会在 **DNS** 解析后发起 **TCP** 预连接。如果这两项都完成了，用户仍然继续录入，**Chrome** 就会在一个隐藏的页签进行预渲染(**pre-render**)。

相对的，如果输入的前置文字找不到合适的匹配项目，**Chrome** 会向搜索引擎服务器发起 **DNS** 预解析和 **TCP** 预连，以获取相似的搜索结果。

平均而言用户从填写查询内容到评估给出的建议需要花费数百毫秒。此时 **Chrome** 可以在后台进行预解析，预连接，甚至进行预渲染。再当用户准备按下回车键时，大量的网络延迟已经被提前处理掉了。

智能地址栏,chrome://omnibox/

智能地址栏，在这里你可以查询你之前输入的地址统计情况，相当于**predictors**的详细查询版。我们可以看一下下图，下面是我查询**famliy.baidu.com**百度内网的信息。

chrome://omnibox

应用

javascript

actionscrip

html5/css3

nginx/uwsgi/linux

python

个人票券

baidu

lenovo

ali

人民大学

sever

google

mobile

杂项

html4.01

github

群晖企业网盘

node

angular

测试

护眼

android

其他书

Enter omnibox input text: family.baidu.com

提交

Input parameters:

Prevent inline autocomplete

In keyword mode

Current page context: new tab page with focus in omnibox

Display parameters:

Show incomplete results

Show all details

Show results per provider, not just merged results

cursor position = 16

elapsed time = 5ms

all providers done = false

host = family.baidu.com has is_typed_host = true

Combined results

Provider	Type	Relevance	Contents	Can Be Default	Starred	HWYT	Description	URL	Fill Into Edit	Inline Autocompletion	Del	Prev	Next	Done	Associated Keyword	Keyword	Dupli
HistoryURL	url-what-you-typed	1413	family.baidu.com	✓		✓	搜索引擎, 上内网	http://family.baidu.com/	family.baidu.com		✓	X	1		family.baidu.com		2
HistoryQuick	history-url	1399	family.baidu.com/portal/	✓		✓	搜索引擎, 上内网	http://family.baidu.com/portal/	family.baidu.com/portal/	/portal/	✓	X	1	✓			1
HistoryQuick	history-url	1314	family.baidu.com/login?serviceUrl=http%3A%2F%2Ffamily.baidu.com%2FserviceUrl=http%3A%2F%2Ffamily.baidu.com%2F	✓		X	搜索引擎, 上内网	http://family.baidu.com/login?serviceUrl=http%3A%2F%2Ffamily.baidu.com%2F	family.baidu.com/login?serviceUrl=http%3A%2F%2Ffamily.baidu.com%2F	/login?serviceUrl=http%3A%2F%2Ffamily.baidu.com%2F	✓	X	1	✓			0
Keyword	search-other-engine	1100	用 family.baidu.com 搜索“输入查询”	✓		X	family.baidu.com 搜索	family.baidu.com	family.baidu.com		X	X	9	✓		family.baidu.com	0
Bookmark	bookmark-title	1090	family.baidu.com/core/index.jsp?chc=1405280916	✓		✓	搜索引擎, 上内网	http://family.baidu.com/core/index.jsp?chc=1405280916	family.baidu.com/core/index.jsp?chc=1405280916	/core/index.jsp?chc=1405280916	X	X	1	✓			0
Bookmark	bookmark-title	1089	family.baidu.com/core/index.jsp?chc=1603971430	✓		✓	搜索引擎, 上内网	http://family.baidu.com/core/index.jsp?chc=1603971430	family.baidu.com/core/index.jsp?chc=1603971430	/core/index.jsp?chc=1603971430	X	X	1	✓			0

数据有好多，大家可以自己看看，我只说几个，下面的五个是相关度比较高的连接，其实这个表最后还有你的visit count,以及你last visit time。当你再url输入的family地址的时候suggestion可能就是从下面出来的，具体的竞争策略我暂时还不太清楚，应该停留时间长度，以及访问次数。都是纬度因子。具体去看一下chromium这个项目的源码吧。

DNS,chrome://dns

这地址下chrome给我们提供了,dns的预解析名单，名单里永远都只有10条数据依据访问的次数和停留的时间长度作为权重，但是确实有次数权重，不过没有罗列出来。这样我们就有了一个列表，一个我们经常访问或者停留时间比较长的dns,目前他没有清空策略，但是有竞争策略，每天打开的时候dns，时间是以当次打开的时间为准。（亲自测试）另外，在这里我们还可以看一下，某一个站点线面都引用了哪些外部资源，或者外部域的内容。

chrome://dns	应用	javascript	actionscrip	html5/css3	nginx/uwsgi/linux	python	个人票券	baidu	lenovo	ali	人民大学	sever	google	mobile	杂项	html4.01	github							
Future startups will prefetch DNS records for 10 hostnames																								
Host name	How long ago (HH:MM:SS)	Motivation																						
http://bj.bs.bae.baidu.com/	06:02:35	n/a																						
http://family.baidu.com/	06:02:35	n/a																						
http://hm.baidu.com/	06:02:35	n/a																						
http://img.baidu.com/	06:02:35	n/a																						
http://log.hm.baidu.com/	06:02:34	n/a																						
http://static.family.baidu.com/	06:02:35	n/a																						
http://uuap.baidu.com/	06:02:34	n/a																						
https://accounts.google.com/	06:02:38	n/a																						
https://clients2.google.com/	06:02:37	n/a																						
https://uuap.baidu.com/	06:02:34	n/a																						

Host for Page	Page Load Count	Subresource Navigations	Subresource PreConnects	Subresource PreResolves	Expected Connects	Subresource Spec
http://static.bshare.cn/	1	1	0	0	1.137	http://iclick.cm.admaster.com.cn/
		1	0	0	1.302	http://www.bznx.net/
		1	0	0	1.857	http://hm.baidu.com/
http://mobile.10jqka.com.cn/	1	1	0	0	2.127	http://log.hm.baidu.com/
		1	0	0	1.857	http://mobile.10jqka.com.cn/
		1	0	0	3.206	http://s.thsi.cn/
		1	0	0	1.857	http://stat.10jqka.com.cn/
		1	0	0	1.265	http://atanx.alicdn.com/
http://stockpage.10jqka.com.cn/	1	1	0	0	4.153	http://cbjslog.baidu.com/
		1	0	0	1.265	http://eclick.baidu.com/
		1	0	0	1.265	http://log.hm.baidu.com/
		1	0	0	1.265	http://pos.baidu.com/
		1	0	0	1.104	http://q.10jqka.com.cn/
		1	0	0	14.743	http://s.thsi.cn/
		1	0	0	3.992	http://stat.10jqka.com.cn/
		1	0	0	2.548	http://stockpage.10jqka.com.cn/
		1	0	0	10.892	http://u.thsi.cn/
		1	0	0	2.127	http://comment.10jqka.com.cn/
		1	0	0	1.857	http://eclick.baidu.com/
		1	0	0	2.937	http://i.thsi.cn/
		1	0	0	2.127	http://log.hm.baidu.com/

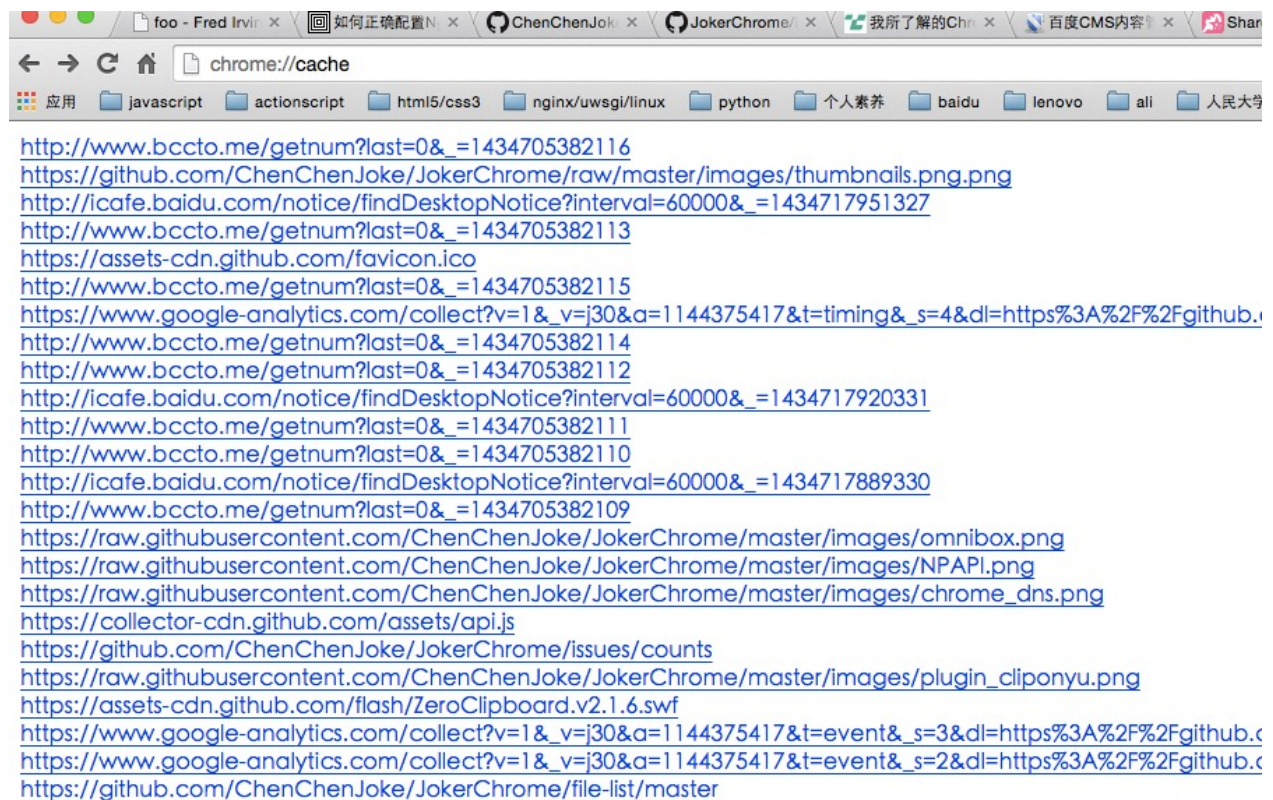
MARK 此处挖掘chrome深度优化（一种备份于本地磁盘(local disk)，另一种则存储于内存(in-memory)中）

浏览器缓存,chrome:cache

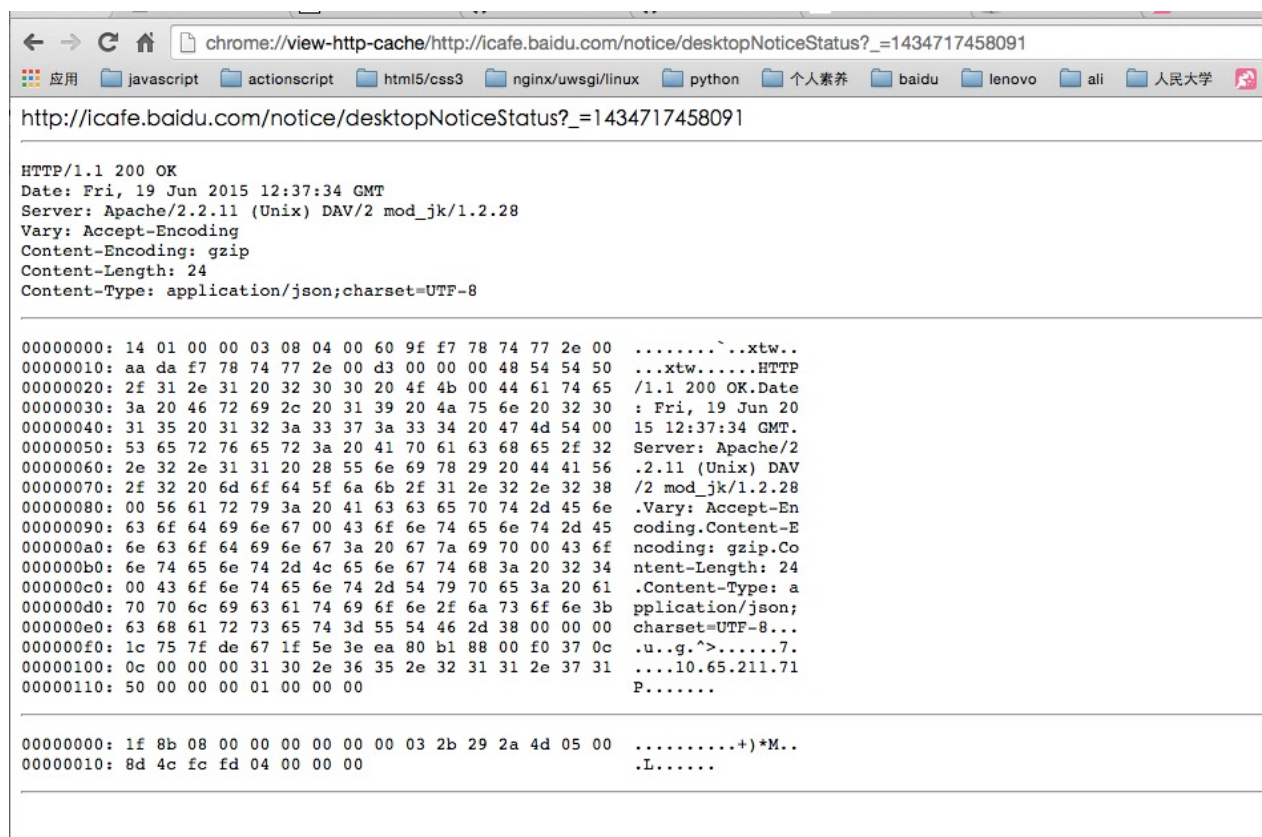
chrome://cache,下面我们输入网址可以先来看一下，chrome到底都缓存了什么东西。第一张图是，我们的缓存列表，里面缓存了我们访问了code为2XX的数据，这个2XX指令码指的http的code可不是前后端同学自定义的code.跳转类的3开头的不会被缓存（301永久跳转,302临时跳转）,4XX也不会,5XX的暂时还没试过。关于code码的解释，附上一个链接。

<http://justcoding.iteye.com/blog/833058>。这里解释的还可以。另外，如果像更深入了解的同学，可以看一下http协议权威指

南。也可以了解下restful思想，也就是Roy Fielding教授在参与定制http协议的时候主推的一种设计思想。也就是Http是应用层协议的原因之一（Apache基金会会有一个开源项目Sling，这个项目就是完全基于Restful思想设计的，他们用jcr进行存储，url作为被操作项，request method本身作为操作符）。



扯远了，下面我们看一下第二张图：



这个缓存的内容，并不是所谓的缓存为具体文件，而是请求本身的缓存，chrome把请求头部信息和请求体 本身都进行了缓存，大家可以看一下下面的信息。在我们实际上把这部分请求头存了下来。

```
HTTP/1.1 200 OK
Date: Mon, 22 Jun 2015 05:14:26 GMT
Via: 1.1 varnish
```



```
Age: 46371
X-Served-By: cache-lax1433-LAX
X-Cache: HIT
X-Cache-Hits: 4019
Content-Encoding: gzip
Content-Type: application/javascript
ETag: W/"53d7c80c-1f33"
Last-Modified: Tue, 29 Jul 2014 16:13:00 GMT
Server: GitHub.com
Content-Length: 2892
Accept-Ranges: bytes
```

其实看到这里的时候我突然有一个方法，chrome是如何获取缓存信息的，如果我们自己开发extension的时候如何获取缓存图片（不通过chrome内部机制）比如：ajax、直接导入、iframe如果我们读取chrome自带的缓存信息，是不是方面我们做一些用户信息收集类的操作。当然这部分获取信息，一定需要用户的授权。或者通过设置chrome的安全策略：Content-Security-Policy来指定加载资源

服务器端再返回的时候只要返回相应的头部信息就可以了。附上 Content-Security-Policy 指令说明，<http://www.2cto.com/Article/201307/230739.html>

net-internals网络请求，chrome://net-internals

查询socket

socket transport_socket_pool Idle占用当我们调试socket的时候。可以通过观察这个指示器来进行观察，个人感觉调试的时候socket意义不太大，但是如果了解socket建立连接，以及释放的过程还是不错的。点击响应的socket会进入到详细信息。

chrome://net-internals/#sockets

Sockets capturing events (2273)

Close idle sockets

Flush socket pools May break pages with active connections

View live sockets

Name	Handed Out	Idle	Connecting	Max	Max Per Group	Gene
transport_socket_pool	1	2	0	256	6	119
ssl_socket_pool	1	0	0	256	6	119
10.240.166.32:80 (http_proxy_socket_pool)	0	0	0	32	6	118
10.240.166.32:80 (http_proxy_socket_pool)->transport_socket_pool	0	0	0	32	6	118
10.240.166.32:80 (http_proxy_socket_pool)->ssl_socket_pool	0	0	0	32	6	118
10.240.166.32:80 (http_proxy_socket_pool)->ssl_socket_pool->transport_socket_pool	0	0	0	32	6	118
10.240.166.32:80 (socks_socket_pool)	0	0	0	32	6	118
10.240.166.32:80 (socks_socket_pool)->transport_socket_pool	0	0	0	32	6	118
10.240.166.32:80 (ssl_socket_pool_for_proxies)	0	0	0	32	6	118
agent.baidu.com:8128 (http_proxy_socket_pool)	0	0	0	32	6	118
agent.baidu.com:8128 (http_proxy_socket_pool)->transport_socket_pool	0	0	0	32	6	118
agent.baidu.com:8128 (http_proxy_socket_pool)->ssl_socket_pool	0	0	0	32	6	118
agent.baidu.com:8128 (http_proxy_socket_pool)->ssl_socket_pool->transport_socket_pool	0	0	0	32	6	118
agent.baidu.com:8128 (socks_socket_pool)	0	0	0	32	6	118
agent.baidu.com:8128 (socks_socket_pool)->transport_socket_pool	0	0	0	32	6	118
agent.baidu.com:8128 (ssl_socket_pool_for_proxies)	0	0	0	32	6	118

transport_socket_pool							
Name	Pending	Top Priority	Active	Idle	Connect Jobs	Backup Timer	Stalled
jialonghiclub-jp.hiclub.jp:8012	0	-	0	1	0	stopped	false
ssl/github.com:443	0	-	0	0	0	stopped	false
ssl/www.google-analytics.com:443	0	-	1	0	0	stopped	false
www.bccto.me:80	0	-	0	1	0	stopped	false

test测试

个人感觉最有用的功能之一，可以查看访问一个地址在使用不同代理（链路）的时候所消耗的时间（Pv4,Pv6还做了区分）如下图：



Input a URL which failed to load, and then click the button to run some tests for why it failed.

URL:

Started connection test suite suite on 2015-06-22 16:36:24.998

Result	Experiment	Error	Time (ms)
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Don't use any proxy		31
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Use system proxy settings		19
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Use Firefox's proxy settings		58
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Auto-detect proxy settings		33
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Don't use any proxy• Disable IPv6 host resolving		25
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Use system proxy settings• Disable IPv6 host resolving		14
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Use Firefox's proxy settings• Disable IPv6 host resolving		17
PASS	Fetch http://www.baidu.com/ <ul style="list-style-type: none">• Auto-detect proxy settings		15

另外当我们输入错误的网址的时候他还能解析出来我们访问错误网址时候返回的时间，以及错误原因。



Input a URL which failed to load, and then click the button to run some tests for why it failed.

URL:

Started connection test suite suite on 2015-06-22 16:48:44.691

Result	Experiment	Error	Time (ms)
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Don't use any proxy	ERR_NAME_NOT_RESOLVED (-105)	165
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Use system proxy settings	ERR_NAME_NOT_RESOLVED (-105)	4
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Use Firefox's proxy settings	ERR_NAME_NOT_RESOLVED (-105)	2
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Auto-detect proxy settings	ERR_NAME_NOT_RESOLVED (-105)	4
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Don't use any proxy• Disable IPv6 host resolving	ERR_NAME_NOT_RESOLVED (-105)	2
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Use system proxy settings• Disable IPv6 host resolving	ERR_NAME_NOT_RESOLVED (-105)	2
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Use Firefox's proxy settings• Disable IPv6 host resolving	ERR_NAME_NOT_RESOLVED (-105)	4
FAIL	Fetch http://www.baidu.com%60/ <ul style="list-style-type: none">• Auto-detect proxy settings• Disable IPv6 host resolving	ERR_NAME_NOT_RESOLVED (-105)	4

至于错误原因我就不一一分析了，这些愿意大家用搜索引擎都可以查到。

DNS到期

在这个选项卡中，能看到dns解析的过期时间，当然标记Expired的是已经过期的，另外我们还可以看到 一个域名到底对应了多少个ip，当dns服务器出现问题的时候我们可以直接那这些ip访问网站（如果他们对ip没有做处理， 或者不需要域名相关的操作的时候）。

chrome://net-internals/#dns

DNS

capturing events (268837)

Capacity: 1000

Current State

Active entries: 80

Expired entries: 298

Hostname	Family	Addresses	Expires
0.gravatar.com	IPv4	211.8.69.27	2015-06-22 17:50:11.395
1.gravatar.com	IPv4	124.40.7.245	2015-06-22 16:45:43.407 [Expired]
11.ur.bdimg.com	IPv4	180.76.22.49	2015-06-22 17:03:00.610 [Expired]
2.gravatar.com	IPv4	124.40.7.245	2015-06-22 16:45:43.407 [Expired]
a.alimama.cn	IPv4	202.108.250.250 202.108.250.240	2015-06-22 16:57:06.130 [Expired]
a.baidu.com	IPv4	220.181.57.146	2015-06-22 16:31:37.910 [Expired]
a.emarbox.com	IPv4	118.26.148.250 123.59.17.247 123.59.17.250 221.122.20.174 221.122.127.92 118.26.148.247	2015-06-22 17:04:01.085 [Expired]
a.tanx.com	IPv4	140.205.155.164	2015-06-22 17:10:18.450
a03.optimix.asia	IPv4	42.159.228.253	2015-06-22 16:57:02.397 [Expired]
accounts.google.com	IPv4	74.125.203.84	2015-06-22 16:59:23.543 [Expired]
acookie.ztcadx.com	IPv4	123.157.214.30	2015-06-22 17:04:27.581 [Expired]
adm.icast.cn	IPv4	122.225.81.41	2015-06-22 17:01:45.439 [Expired]
adx.adnxs.com	IPv4	68.67.128.111 68.67.129.43 68.67.129.120 68.67.129.178 68.67.129.185 68.67.128.9 68.67.128.26 68.67.128.74	2015-06-22 17:00:17.647 [Expired]
		54.192.211.2 54.192.211.4 54.230.210.219	

SPDY




SPDY协议是Google提出的基于传输控制协议(TCP)的应用层协议，通过压缩、多路复用和优先级来缩短加载时间。该协议是一种更加快速的内容传输协议。是http协议的增强 CDN供应商Akamai的首席产品架构师GuyPodjarny发现，在目前的实际环境中，SPDY并不快，它只是比HTTPS略快4.5%，比HTTP慢3.4%。SPDY需要客户端和服务端的支持，而服务器的优化对速度提高至关重要，现阶段除了Google外支持SPDY的第三方并不普遍。2012年7月，Facebook工程师道格·比弗表示，Facebook已经开始执行SPDY协议，并将在全球范围内部署。 后面是httpWatch对三个协议的各方面对比。 Raw HTTPS : SPDY/3.1 : HTTP/2 性能对比 <http://blog.httpwatch.com/2015/01/16/a-simple-performance-comparison-of-https-spdy-and-http2/>

thumbnails浏览器快照,chrome://thumbnails

主要是我们访问的页面的，页面快照，被照下来的图片都是data:image/png;base64这种形式的，所以没有地址，也不要想着复用了。

chrome://thumbnails

应用 javascript actionscript html5/css3 nginx/uwsgi/linux python 个人素养 baidu lenovo ali 中国人民大学 sever google mobile

- http://query.hao123.com/
- http://icms.baidu.com:8080/
- http://icms.baidu.com:8080/
- http://erp.baidu.com/
- http://erp.baidu.com:8001/bprouting/BpFlowRouting?appindex=true&t=0.10782439773902297
- http://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpFlowRouting%3Fappindex%3Dtrue%26t%3D0.1078243
- https://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpFlowRouting%3Fappindex%3Dtrue%26t%3D0.107824
- http://erp.baidu.com:8001/bprouting/BpFlowRouting?appindex=true&t=0.10782439773902297&ticket=ST-396058-naPtxG1MZycq45RRKBTM-uwap
- http://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpSecurityAuthentication%3Fappindex%3Dtrue&t=
- https://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpSecurityAuthentication%3Fappindex%3Dtrue&t=
- http://erp.baidu.com:8001/bprouting/BpSecurityAuthentication?appindex=true&ticket=ST-378960-Co3QOMVikW1i00nok6dmk-uwap
- http://erp.baidu.com:8001/portal/me.html?resAppId=10001&resType=EXT_PAGE&resId=1&t=1434006555411
- http://erp.baidu.com:8001/bprouting/BpFlowRouting?action=routing&resType=EXT_FUNCTION&resId=21&resAppId=10009
- http://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpFlowRouting%3Faction%3Drouting%26resType%3DEX
- https://uap.baidu.com/login?service=http%3A%2F%2Ferp.baidu.com%3A8001%2Fbprouting%2FBpFlowRouting%3Faction%3Drouting%26resType%3DEX
- http://erp.baidu.com:8001/bprouting/BpFlowRouting?action=routing&resType=EXT_FUNCTION&resId=21&resAppId=10009&ticket=ST-379026-wf5b
- http://fusion.baidu.com/faces/asset/apply.jsf?resAppId=10009&resType=EXT_FUNCTION&resId=21&t=1434006571281&ut=d530812a81eefdb10e0c
- http://uap.baidu.com/faces/asset/apply.jsf?jsessionid=8M9hV50LQhf9Vv1xV0n3S3Q1RTxQ0xtNxl0VBTVLZnNx0NznbyZ!-1044069384?resAppId=
- http://fusion.baidu.com/BAdfAuthentication?login=true&loginType=form&requestParams=faces%2Fasset%2Fapply.jsf%3Fut%3Dd530812a81eefdb10e0c
- http://fusion.baidu.com/faces/asset/apply.jsf?ut=d530812a81eefdb10e0c16084996de7&resType=EXT_FUNCTION&resId=21&t=1434006571281&res
- http://fusion.baidu.com/faces/asset/apply.jsf?ut=d530812a81eefdb10e0c16084996de7&resType=EXT_FUNCTION&resId=21&t=1434006571281&res
- http://fusion.baidu.com/faces/asset/apply.jsf?ut=d530812a81eefdb10e0c16084996de7&resType=EXT_FUNCTION&refreshTime=1434006571559&re
- http://erp.baidu.com/
- https://github.com/
- https://github.com/

memory-redirect浏览器内存消耗,chrome://memory-redirect

chrome的内存消耗，我们可以看一下，有哪些

About memory Measuring memory usage in a multi-process browser

Summary

Browser	Memory			
	Resident	Shared	Private	Virtual
Google Chrome 37.0.2062.94	421,060k	N/A k	N/A k	12,399,724k
Safari	18,836k	N/A k	N/A k	3,902,052k

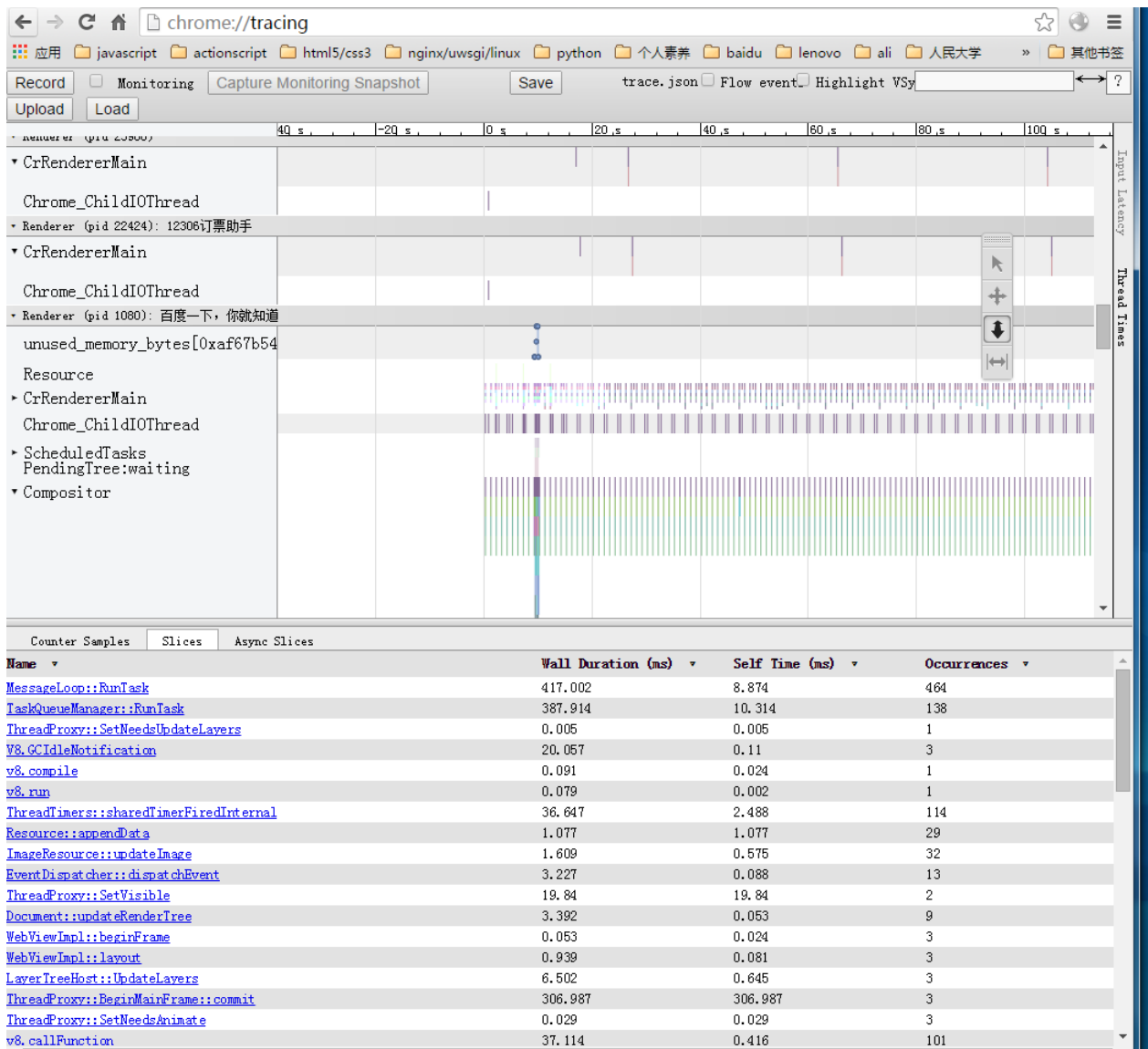
Note: Chrome includes memory used by plug-ins, other browsers may not.
(Bug: We seriously overcount our own memory usage: [Issue 25454](#).)

Processes

PID	Name	Memory			
		Resident	Shared	Private	Virtual
465	Browser	158,980k	N/A k	N/A k	2,101,044k
46923	Tab (Chrome)	N/A k	N/A k	N/A k	N/A k
	About Memory				
46801	Tab	N/A k	N/A k	N/A k	N/A k
	10分钟邮箱 - BccTo.ME				
46154	Tab	63,492k	N/A k	N/A k	1,189,552k
	iCafe+项目管理统一平台				
45617	Tab	N/A k	N/A k	N/A k	N/A k
	Developer Tools - http://jialongclub-jp.hiclub.jp:8060/liveroom/studio/index?roomid=104036				
45374	Tab	80,456k	N/A k	N/A k	1,554,344k
	Sharon E /MARVEL\				
43382	Tab	N/A k	N/A k	N/A k	N/A k
	JokerChrome/mages at master · ChenChenJoke/JokerChrome				
39142	Tab	18,196k	N/A k	N/A k	1,243,976k

追踪访问URL,chrome://tracing/,包括浏览器和渲染过程非常详细,

你可以选择自己感兴趣的页面，进行数据的收集以及分析，这里面写的非常详细，甚至详细到浏览器每一个方法所消耗的时间。

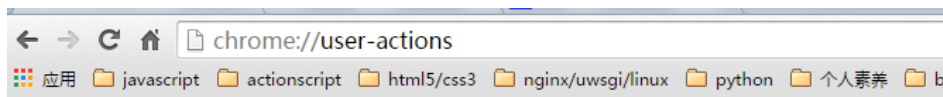


MARK详细最终他的使用、以及意义拓展预计一天;

tranc图一

用户行为分析，包括用户点击切换tab,chrome://user-actions

主要手机用户行为，当过切换tab，而且包括浏览器点击（有效或者无效）



Listening for user actions...

User Action	Timestamp (sec.)
OmniboxInputInProgress	1434088298.575
ActiveTabChanged	1434088394.79
ActiveTabChanged	1434088396.725
OmniboxInputInProgress	1434088396.733
ActiveTabChanged	1434088399.602
ActiveTabChanged	1434088401.303
OmniboxInputInProgress	1434088401.313
ActiveTabChanged	1434088405.26
NavEntryCommitted	1434088405.783
NavEntryCommitted	1434088406.15
ActiveTabChanged	1434088406.959
OmniboxInputInProgress	1434088406.979

后记：其实chrome里面有趣的小工具不止这些，有的人可能会说这些东西并没有什么卵用。有的人可能用到里面其中一个，或者其中几个。但是我始终感觉，作为一个前端工程师很有必要了解我们朝夕相处的伙伴。

