Finding way around chromium's source code

Asked 13 years, 3 months ago Modified 13 years, 2 months ago

Viewed 6k times

28

I just checked out chromium's source, but I desperately need to learn how to navigate around this monster.

How would I search for parts of the code that implement behavior/features I'm interested in?

Let's say I want to see what happens after a URL is entered into the address bar. How do I find that piece of code?

Or, that I want to see what happens when, while parsing HTML, a certain tag is reached.

I have before me a huge amount of source code, and no skill of navigating around it. How do I learn that skill?

google-chrome chromium

Share

Improve this question

Follow

edited Sep 9, 2011 at 15:26

asked Sep 9, 2011 at 13:02

ntl0ve
1,986 ● 4 ● 19 ● 25

5 As a starting point, read chromium.org/developers/how-tos/… - that should give you a better overview of what you're seeing in that monstrous src/ directory. (I didn't really think this would be a good *answer*, as it is merely a tip..) – maligree Sep 9, 2011 at 13:15

3 Also have a ready of How do you find your way around a new codebase – Justin Sep 9, 2011 at 15:43

2 grep is your friend. or find main and trace. Navigating large code bases is not trivial. – Chris Sep 13, 2011 at 21:36

2 Answers

Sorted by: Highest score (default)

13 ✓

My recommendation for sort of diving in is to take a look at the source for http://code.google.com/p/chromiumembedded/.

It's sort of the condensed version of Chrome and if you look at the files it specifically uses, either ones included in its source tree, or files included therein from the Chromium repo at large. The Chromium code base is a huge amount of stuff, most of which isn't actually in the browser. There's a ton of pulled in code from third party repos which are then boiled down in the build process or Chromium's implementation is located somewhere else in the tree, there's a lot of side projects that (while interesting and an awesome resource for a wide breadth of stuff) will prevent you from achieving your goal of specifically honing in on the browser implementation and how that fits together.

CEF is great because you can see someone who's already done the process of pulling all that stuff together to build a project very specifically scoped at the browser view and nothing else. You can see which parts are primarily derived from webkit easily, you can see where the crossover comes in with Google's implementations, and you can see pretty easily how V8 gets tossed into the mix.

I do say "easily" in relative terms because we're still talking a huge amount of code overall. CEF will put you smack in the center of the requirements, but that stuff is still pulling in the massive amount of various things from the rest of the tree. Compiling it takes me about an hour on a really good computer with 12 gigs of ram and 8 cores, and the generated files take up like 6-10 gigs depending.

At the very least, there's not going to be any sort of quick jump into the shallow end to pick something here or there piecemeal. Browsers are incredibly complex pieces of engineering necessarily, because they have to subsume such a huge amount of individual pieces of functionality and then combine them into a shared context. You may find the one thing you're looking for, but you'll find that it's part of a class library that likely is composed of dozens or hundreds of files, which in turn relies on a hundred more of these libraries to handle each task, so to really take something away you'll have to commit time to taking in a lot more than any given piece of information.

Edit: oh also as your specific example.

src is root
http://src.chromium.org/viewvc/chrome/trunk/src

/chrome
http://src.chromium.org/viewvc/chrome/trunk/src/chrome

The "chrome" tree largely contains the direct implementations (a lot of stuff isn't in there though, most of it even, but that's the starting point). This has overlap with chromeos (chromeos is kind of chromium browser taken to a crazy extreme)

/chrome/browser
http://src.chromium.org/viewvc/chrome/trunk/src/chrome/browser/

Is getting you to close to where you want to be. You start to see specific references to things that you can match to the browser, like the tabs and whatnot (ignoring the giant elephant of the actual browser implementation itself which is what takes up the majority of the mindspace in all this stuff)

/chrome/browser/ui
http://src.chromium.org/viewvc/chrome/trunk/src/chrome/browser/ui/

Brings you to where most of the ui code is for the browser. It can be confusing when there's crossover or when stuff migrates, like there's a "ui" in the root src directory which has some crossover.

And finally http://src.chromium.org/viewvc/chrome/trunk/src/chrome/browser/ui/omnibox/

Which has a surprisingly small amount of code in it. But this is what you find a lot. The code here is an implementation of a number of classes that are built up elsewhere. For non-webview gui component you'll find them mostly pointing back to the root "ui" and the native widgets stuff there, which is where the bulk of the actual event handling code is if I remember correctly.

Share Improve this answer Follow

edited Oct 12, 2011 at 8:49

answered Oct 12, 2011 at 8:31

user748221

Thank. You. Seriously, great answer -- thanks a lot. This, coupled with what I learned from reading the docs pointed to by maligree in his comment sounds like a blast. – ntl0ve Oct 26, 2011 at 8:44

It was hard fought knowledge. Google doesn't promote the kind of approachable environment that is common in web related development contexts. This was commented on recently in a post by a Google engineer (accidentally made public). The quality of engineering is phenomenal in most of their work, very much so with Chrome. Essentially it's high enough quality code that the people they expect to work on it can efficiently make use of it without much documentation. The point though is their target audience is not one that requires learning. It's a part of their company's culture culture – user748221 Oct 27, 2011 at 10:41

and a function of having a very, very high level baseline expectation of the people approaching the code. But it does mean there's a wealth of incredibly high quality open source code that is kind of like dark matter. It's there but not approachable without coming in with a high level of expertise or a lot of time to dump into bubbling through it, so only benefits a handful of people who are able to pour a bunch of time in with little initial direction. – user748221 Oct 27, 2011 at 10:42

Why there are two ui folders? There should be a reason why they separated it. – Ameen Dec 9, 2015 at 11:18

@benvie Great answer and thanks. Any ideas on what part of the source trunk would deal with rendering overlay elements in the HTML5 video player (elements that overlay the video canvas but are not part of the video stream)? I want to create my own overlay elements that are not part of the controls (chrome) that border the video canvas, but are elements overlay the video canvas itself, possibly ina semi-transparent manner. – Robert Oschler Mar 21, 2017 at 10:03

▲

1

▼

You can try this... it may actually lead somewhere too :-)

http://aaronboodman-com-v1.blogspot.com/2010/10/wherein-i-help-you-get-good-job.html

Reading through the dev forums may help too...

http://groups.google.com/a/chromium.org/group/chromium-dev/topics

Also, this section has a lot of useful documents, such as style guides, etc.

http://dev.chromium.org/developers/contributing-code

Last, but not least, IRC is your friend...

http://dev.chromium.org/developers/irc

Share Improve this answer Follow

edited Sep 14, 2011 at 3:31

answered Sep 14, 2011 at 3:25

Homer6
15.1k ● 11 ● 64 ● 82