Google™ 09 I O

Dissecting a Google Chrome Extension

Aaron Boodman
May 2009

Administrativa

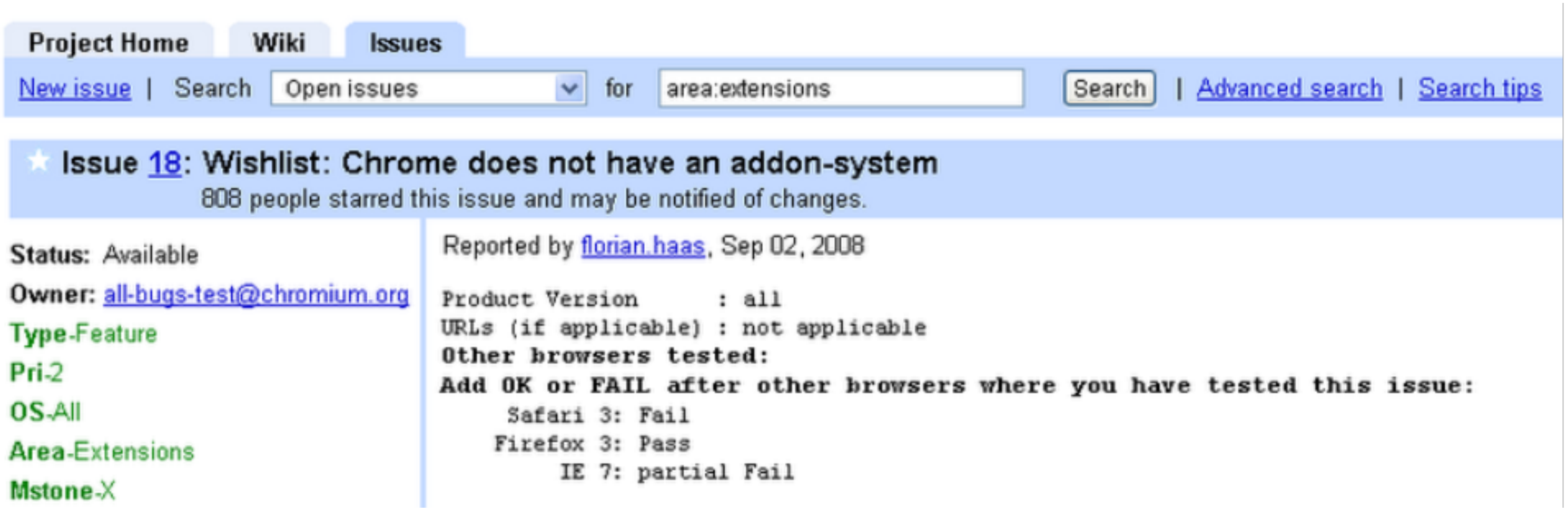
- Introductions
- Caveats
- Agenda
  - Three cool things about Google Chrome Extensions

But First: Why Extensions?

- Seriously, *Why*?

But First: Why Extensions?

- Seriously, *Why* ?
- You made us



Issue 18: Wishlist: Chrome does not have an addon-system

808 people starred this issue and may be notified of changes.

Status: Available
Owner: all-bugs-test@chromium.org
Type-Feature
Pri-2
OS-All
Area-Extensions
Mstone-X

Reported by florian.haas, Sep 02, 2008

```
Product Version      : all
URLs (if applicable) : not applicable
Other browsers tested:
Add OK or FAIL after other browsers where you have tested this issue:
     Safari 3: Fail
    Firefox 3: Pass
         IE 7: partial Fail
```

But First: Why Extensions?

- Seriously, *Why*?
- You made us

Some other good reasons:

- Keep Chrome minimal
- A customized browser for every user
- Prototype new feature ideas

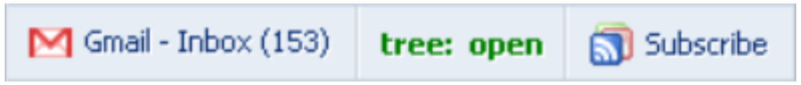
CT1: Extensions are Web Pages

HTML, CSS, and JavaScript

- Extensions are packages (zip files) containing HTML, CSS, and JavaScript
- Each piece of UI in an extension is a fully-functioning webpage
- Writing extensions is just like writing web pages. Use the same debugging tools, the same JavaScript libraries, and the same techniques.
- There's an easy, iterative development cycle
- Try it! Google: chrome extensions howto

We make them look good

```
<div id='button' class='toolstrip-button'>
  <img src='icon.png'>
  <span>Subscribe</span>
</div>
```



- But you can use all your CSS tricks, if you want
- ... Or pick up some new tricks for webkit-specific CSS extensions.

Cross-origin XMLHttpRequest

```
var req = new XMLHttpRequest();
req.open("GET", "http://www.google.com/reader/api/0/...",
        true);
req.onreadystatechange = function() {
  ...
};
req.send(false);
```

- Shared cookie jar with web content
- Extensions declare the origins they want access to in the manifest

HTML5 Local Storage

```
localStorage.setItem("foo", "bar");
console.log(localStorage.getItem("foo"));
```

- Reuse standard APIs, no separate settings API
- More coming all the time...

Browser APIs: Approach

- Narrow
- Webby

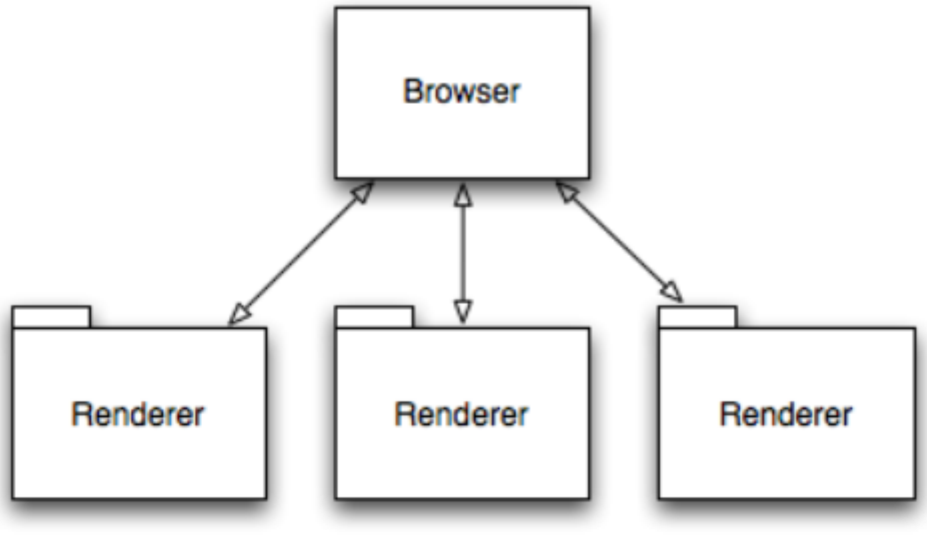
```
chrome.bookmarks.create({
  title: "Lovely green",
  url: "javascript:void(document.body.bgColor='green')"
});
```

Browser APIs: Which?

- Tabs and windows
- Bookmarks
- Downloads
- etc... (exact list TBD)

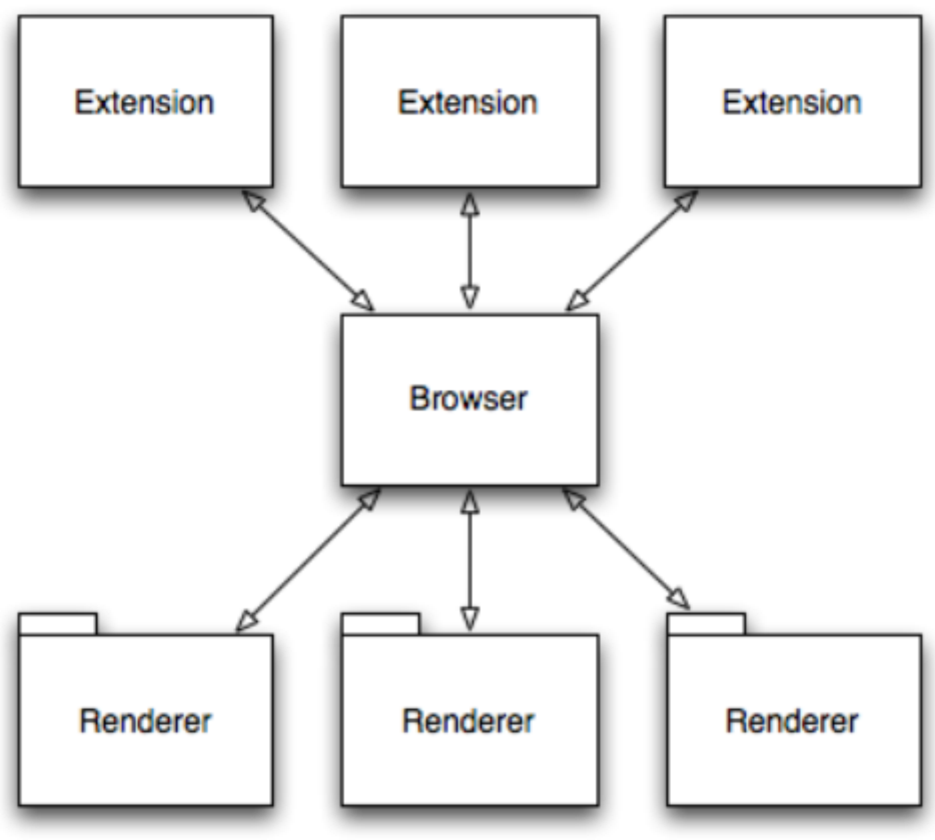
CT2: Extension Process Model

Chrome: A Multiprocess Browser



- One process for each tab and plugin
- Web pages and plugins can't crash browser
- Exploits in tabs are contained
- Better resource sharing

Extensions have their own processes, too.



- One process for each extension
- Extensions can't crash browser
- Exploits are contained
- Better resource sharing

Extensions are *multiple* web pages

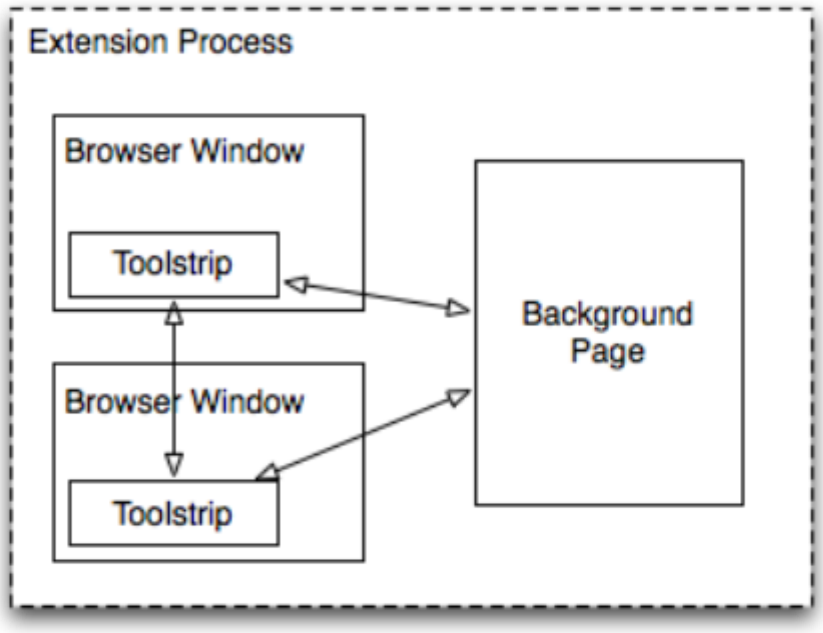
- Each toolstrip, sidebar, etc. is a web page.
- Each browser window gets its own set of widgets.

Extensions pages can communicate

- They're all in the same process, on the same thread.
- Communication is similar to inter-frame communication, or talking to a popup window.
- Direct function calls.

```
var total = 0;
chrome.extension.getToolstrips().forEach(function(toolstrip) {
  total += toolstrip.someFunction("foobar");
});
console.log("total is: " + total);
```

The background page ties it all together



- A single persistent context independent of windows.
- Majority of "application code" goes in background page, toolstrips and sidebars more like dumb views.

```
button.onclick = function() {
  div.innerHTML = chrome.getBackgroundPage().doSomethingHard();
}
```

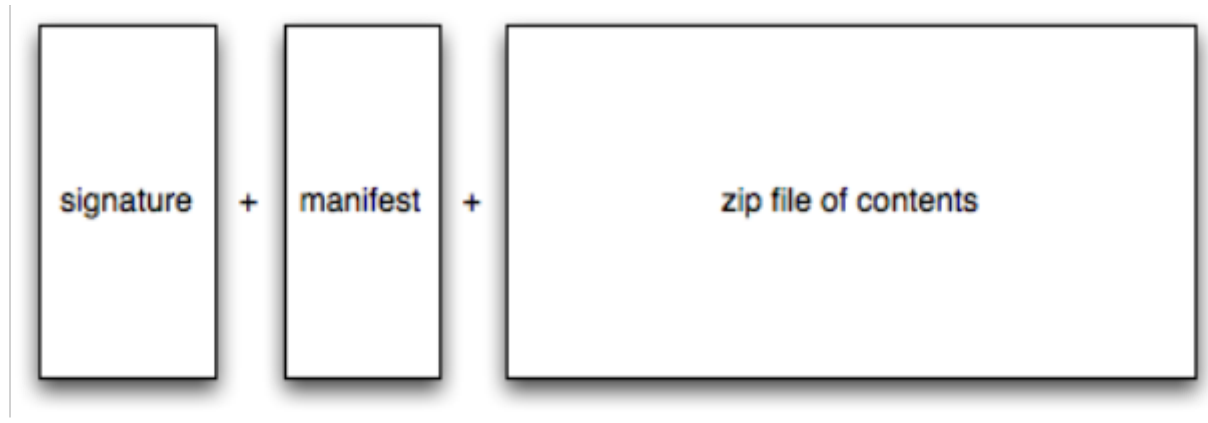
AJAX-Style, Asynchronous APIs

```
chromium.tabs.create(
  { url: "http://www.google.com/" },
  function(tab) {
    alert("Got tab with id: " + tab.id);
  }
);
```

- Multiprocess requires async APIs.
- The browser process becomes the "server".
- We use common AJAX patterns to make async programming easier.

CT3: Packaging and Distribution

CRX files



- Extensions are signed to prevent MITM attacks.
- Manifest is prepended to allow install UI to show up quickly.
- Don't worry about package details. Google will provide a service that does this, but the format is open.

Deployment, Installation

- Copy CRX to your server to deploy.
- Installation is instant
  - No restart!
- There will be a Google service to host your CRX files

Update

- Updates are automatic
  - no work required by users
  - users always have latest version
  - no restart prompt

- Forward compatible with future Chrome versions
- Google will provide an easy-to-use update service

Gallery

- There will be an extension gallery
- Nothing more on this quite yet :)

Get Started

Google: Chrome Extensions HOWTO

Email: chromium-discuss@googlegroups.com

Google™ 09 I O