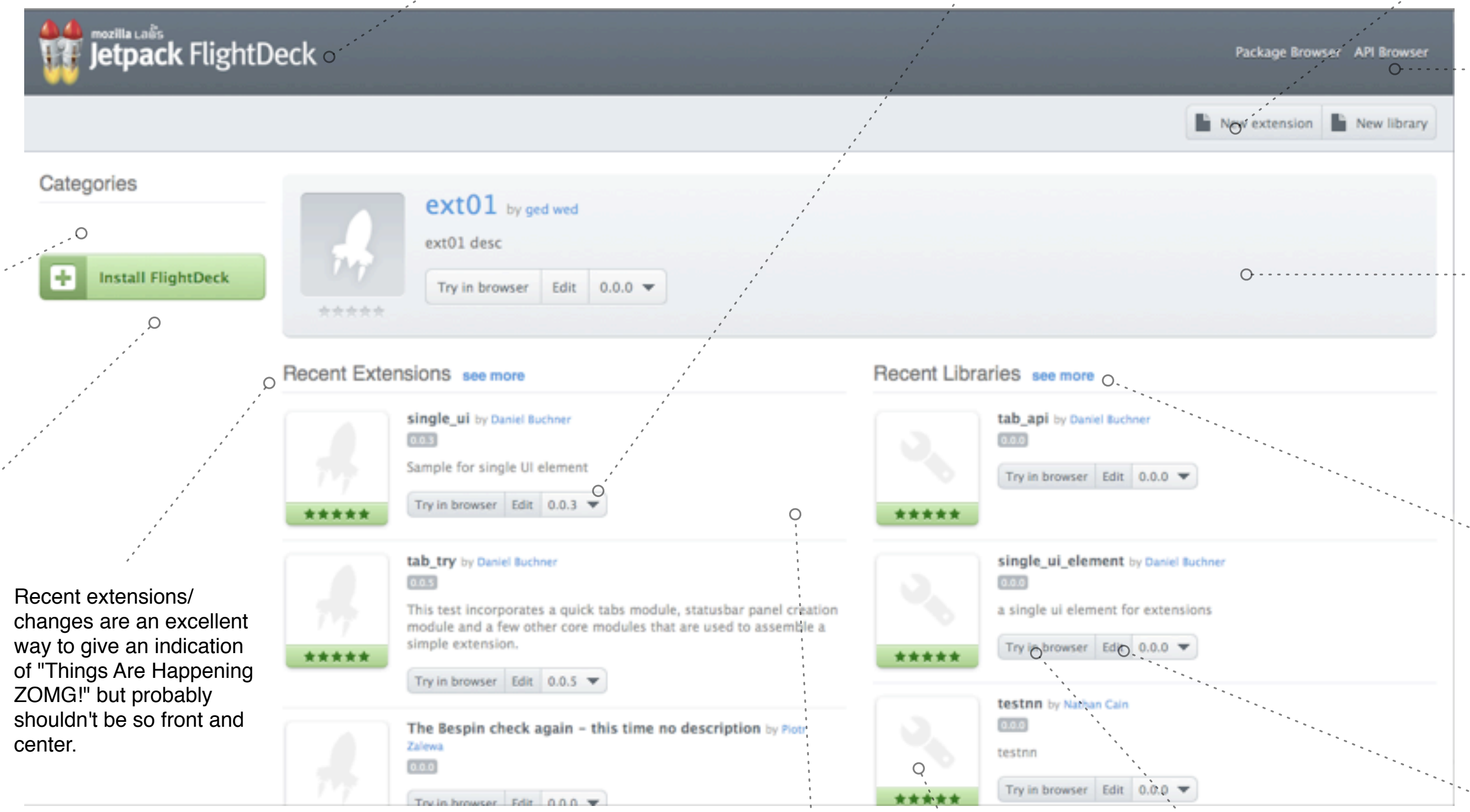
Jetpack Builder UX Review

Aza Raskin 15 April 2010

Front Page


mozilla Labs
Jetpack FlightDeck
Package Browser
API Browser
New extension
New library
Categories
Install FlightDeck
ext01 by ged wed
ext01 desc
Try in browser
Edit
0.0.0
Recent Extensions see more
single_ui by Daniel Buchner
0.0.3
Sample for single UI element
tab_try by Daniel Buchner
0.0.5
This test incorporates a quick tabs module, statusbar panel creation module and a few other core modules that are used to assemble a simple extension.
The Bespin check again – this time no description by Piotr Zalewa
Recent Libraries see more
tab_api by Daniel Buchner
single_ui_element by Daniel Buchner
a single ui element for extensions
testnn by Nathan Cain
testnn

Should be Firefox Add-On Builder

It doesn't make sense to include a version picker as such primary piece of UI. It's a rare use case to go history diving. Much more useful are # of installs, reported errors, comments, # of other add-ons the author has written, how were they rated, etc. Something that yields more information about why I should be interested in this add-on/ library.

This is the point of this page, why are they hidden out of the way? The win scenario is to have the dev make a new add-on/library.

It makes sense to see these only when you are inside the main builder interface.

It is nice to have a featured extension, but it should be called out explicitly as such. Sitting here is is just a bit confusing without explanation.

Why is the install button in Categories? Why does categories exist here?

Not sure that we should show the install button expect (1) during the tutorials and step-by-step walkthroughs, (2) When the user tries to install one of the add-ons prompt them to first install

Recent extensions/ changes are an excellent way to give an indication of "Things Are Happening ZOMG!" but probably shouldn't be so front and center.

Having a see more link is good, but should appear at the end of the list, where it will fall in the path of a reader who wants to see more.

I think this will create confusion: am I editing this actual version? Won't I disrupt the original authors work? While *I* know that isn't the case, it isn't clear from this wording. Instead, we should go with "View Source" and then inside of View Source page give the option of Forking or Duplicate & Edit.

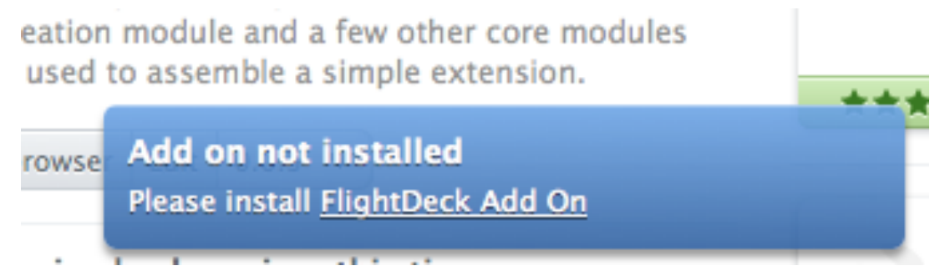
The goal of the front page is to provide a launch spot for the developer. It needs to communicate a whole bunch of context to get the user to the right spot. In particular:

- What is the Firefox Add-On Builder? (with appropriate links to downloadable the Firefox Add-On Builder SDK)
- A entry way into tutorials, walk-throughs, and intros [build your first add-on!]
- A large link to build an add-on, and slightly smaller link to build a library

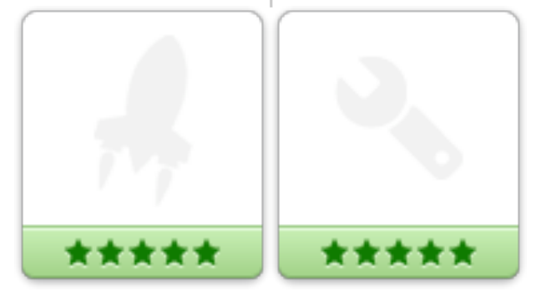
Random thought on making sure all libraries and add-ons have icons: if the author doesn't provide an icon, allow visitors to supply one (which can then be voted on, etc.)

Overall, the graphic style and layout is pleasing and airy. Very well done!

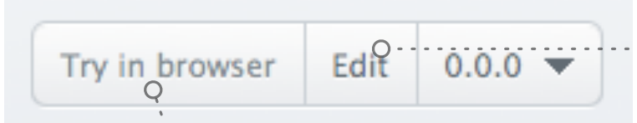
It is totally awesome to have an instant-try button! I wonder if "Try in browser" are the right words. We should probably use words that emphasize the instant nature of getting to try it out. Perhaps "Try Instantly" or "Test Now"... more thought is needed :)


eation module and a few other core modules used to assemble a simple extension.
Add on not installed
Please install FlightDeck Add On

This message should only be displayed when you do something that required the add-on being installed! Also, the text should be more useful: "To insta-try this add-on you first need to install the Jetpack Builder add-on" or some such.



The rating widget looks great. I'm assuming that actual rating things has not yet been implemented?

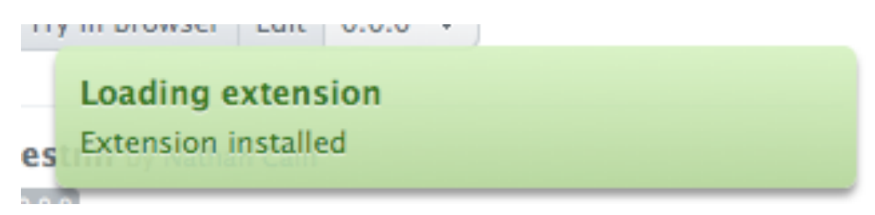


The buttons are missing a hover state and a click state. You don't get good visual indication that the buttons are clickable or clicked.

http://flightdeck.zalewa.info/extension/ext01/v_0.0.0/create_xpi/

Nitpick: isn't this URL a little bit odd. I would imagine it should be /v_0.0.0/ try_now or something that indicates what it will do from a user perspective instead of create_xpi (which I'd use for when the user tries to download the extension)

"Edit" here should actually be written as "View Source". Besides being more friendly and less confusable with destructive actions, we also do not have to go through a login page. The more we allow without require a user to go through the credential phase the better — we'll get more users and with a higher engagement.



It is excellent that we get a message that the something is going on and that the extension has installed correctly.

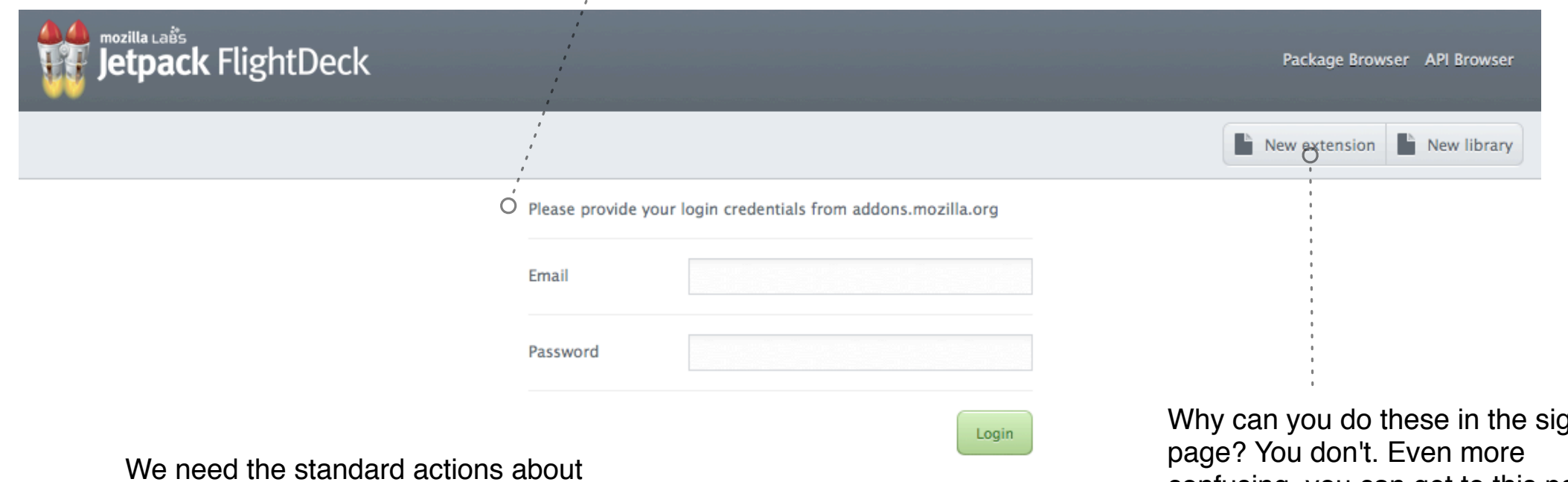
That said, there are a couple of issues:

(1) There needs to be *immediate* feedback after clicking the button. Right now it takes seconds for this message to appear — and far away from my foveal vision, which is centered on the button I just clicked. We need indication on or very near the button that we doing stuff.

(2) "Loading extension/Extension installed" these are saying two different things. We should provide more real-time feedback on what's going on, either a spinner with text below it, or a indicator progress bar.

(3) If at all possible, we should be doing the minimal amount of static analysis so that we can tell the user in this pop-up a list of the types of features provided (i.e., "Adds a simple feature and preference.")

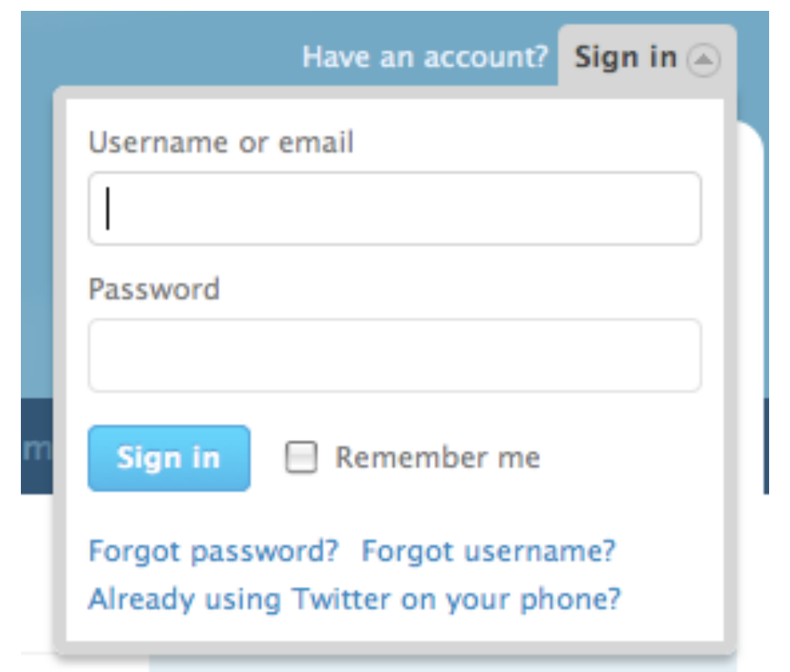
Login Page



Should read "Log in to the Firefox Add-On Builder with your addons.mozilla.org account. Don't have an account? Get one."

We need the standard actions about forgetting email/password/etc.

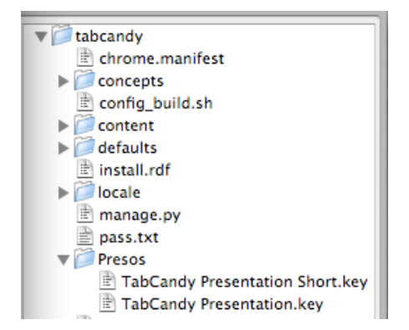
Why can you do these in the signup page? You don't. Even more confusing, you can get to this page by having clicked one of these links!



Logins should happen in-vivo. That is inline to the page, and from the action that required logging in. Twitter has a good implementation of an in-vivo login.

The goal is to interrupt the users workflow as little as possible and to retain their context as much as possible (which is why in-place sign in is good).

Because Jetpack-based add-ons can contain many resources, we are going to need a more rich resource/file/folder manager.

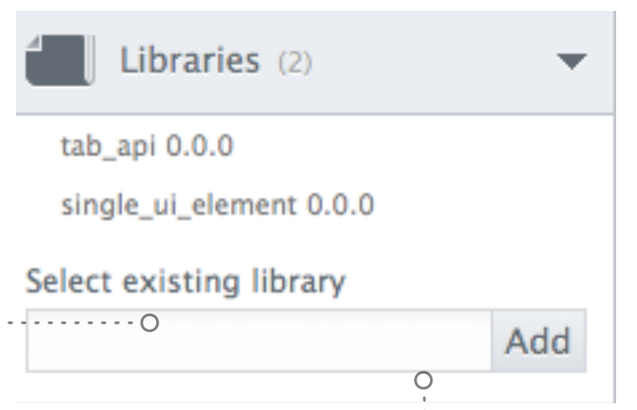


I'd take inspiration from the drawers often shown in OSX code editors (the screenshot is from TextMate).

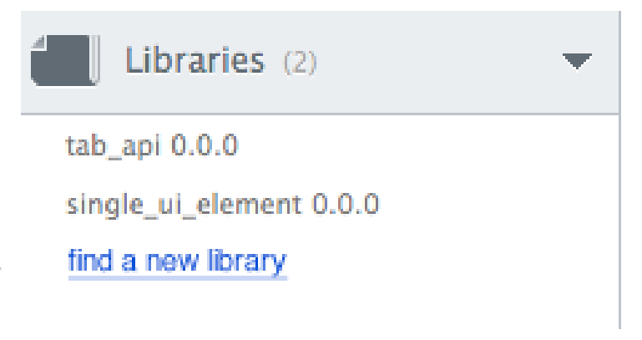
For now, let's have two folders that handle the basics.

- ▶ Meta
  - Manifest
  - Description
  - Version Description
- ▶ Resources
  - main.js
  - New File

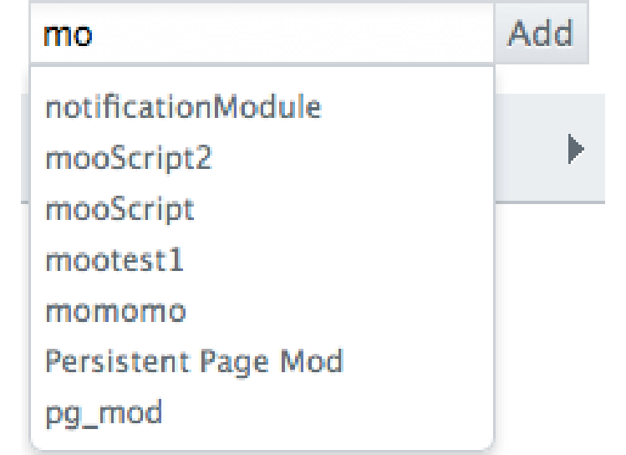
Should say, "Included Libraries". At the bottom of the opened view, should

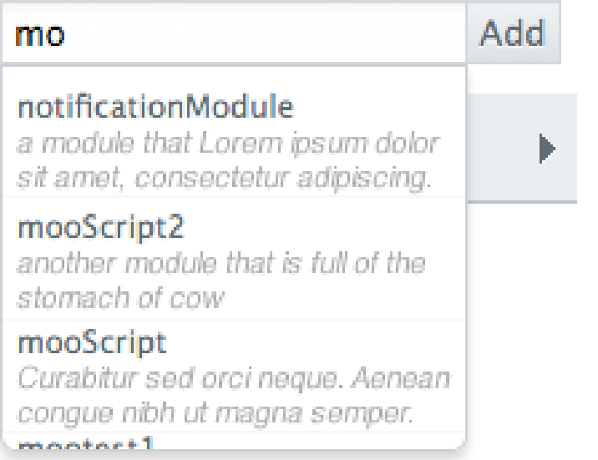


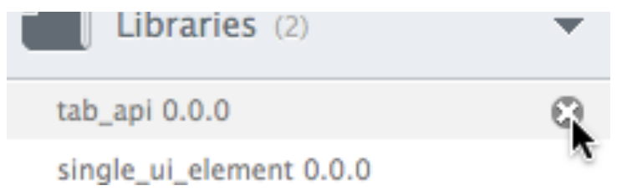
This should be hidden behind a progressive disclosure link. It's confusing to see a box here when this is for currently used libraries.



The live autocomplete of existing libraries is fantastic! Just a name, however, is probably not enough information for a developer to be able to easily decide which library is apropos. Let's add in the first two lines of the library's description.



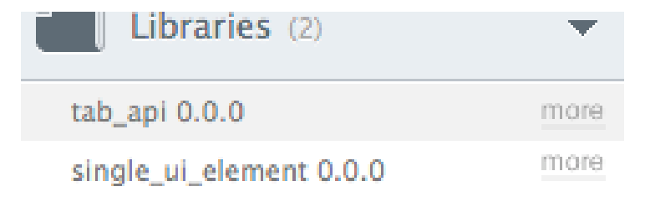

mo
Add
notificationModule
a module that Lorem ipsum dolor sit amet, consectetur adipiscing.
mooScript2
another module that is full of the stomach of cow
mooScript
Curabitur sed orci neque. Aenean congue nibh ut magna semper.



Why are there close buttons on these library items? We should get rid of them.

here is a strange coding breakage of try going on here: is the library added if I am using a "require" statement, and click the close button here, does the require statement get automatically deleted?

It is useful to provide affordances for finding new libraries that you want to use (and then automatically add the require statement at the top of the file) because finding a new library requires more knowledge than the inverse.



The Editor

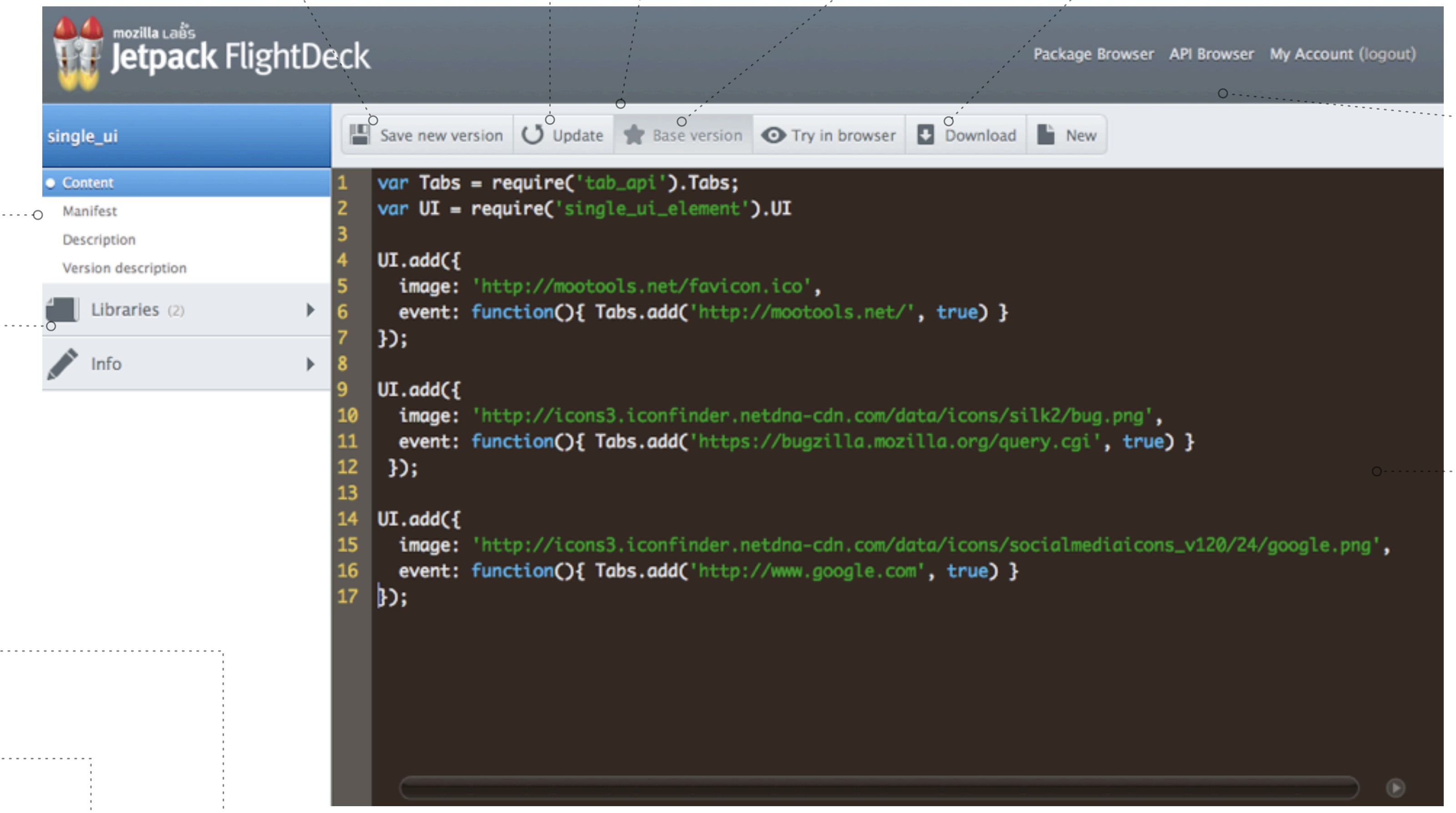
Is the point of this to mark a new revision or to fork the project?

Why "Update" — That implies that the currently running test extension will be reloaded with the new code. I think what we mean here is "Save".

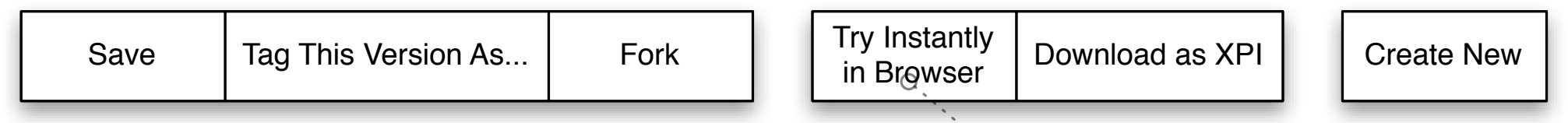
We need to separate these into distinct action groups in a logical fashion — visual grouping is important for a user to know where to go for an action.

What does "base version" mean?

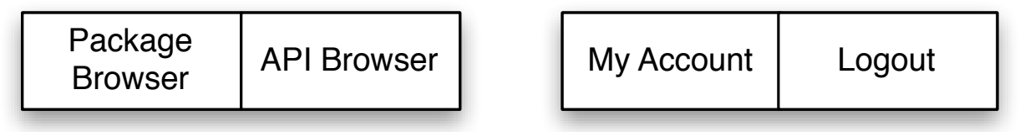
Download as XPI is probably a better wording.



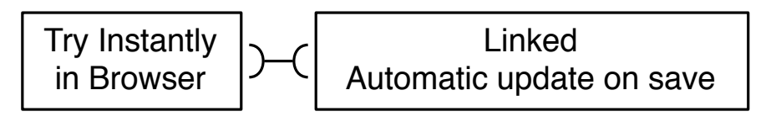
```
var Tabs = require('tab_api').Tabs;
var UI = require('single_ui_element').UI

UI.add({
  image: 'http://mootools.net/favicon.ico',
  event: function(){ Tabs.add('http://mootools.net/', true) }
});

UI.add({
  image: 'http://icons3.iconfinder.netdna-cdn.com/data/icons/silk2/bug.png',
  event: function(){ Tabs.add('https://bugzilla.mozilla.org/query.cgi', true) }
 });

UI.add({
  image: 'http://icons3.iconfinder.netdna-cdn.com/data/icons/socialmediaicons_v120/24/google.png',
  event: function(){ Tabs.add('http://www.google.com', true) }
});
```



Package Browser and API Browser should be visually separated from "My Account" and "Logout".



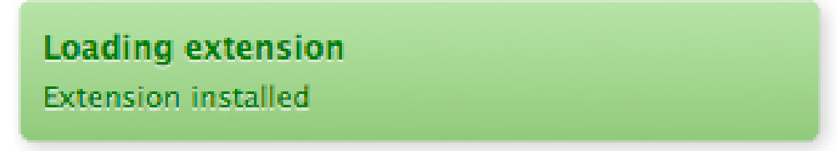
This should be a multi-state button. After you click it, it should "link" the version being shown with the browser with the save last save. Thus "Trying in browser" will only have to be clicked once. **Closing or refreshing the window, of course, removes the extension.**



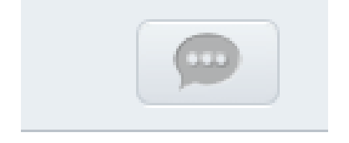
On hover, text switches to say "Click to Uninstall"

The current version of Bespin still feels sluggish (especially scrolling). If by time of launch, we are having performance issues, I recommend using a more mature platform for the time being. Performance will be key in creating a great user experience. If it feels slow, developers will eschew it.

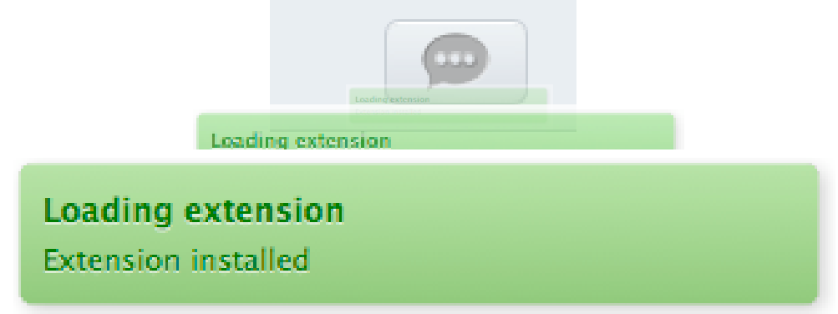
Messages & Notifications



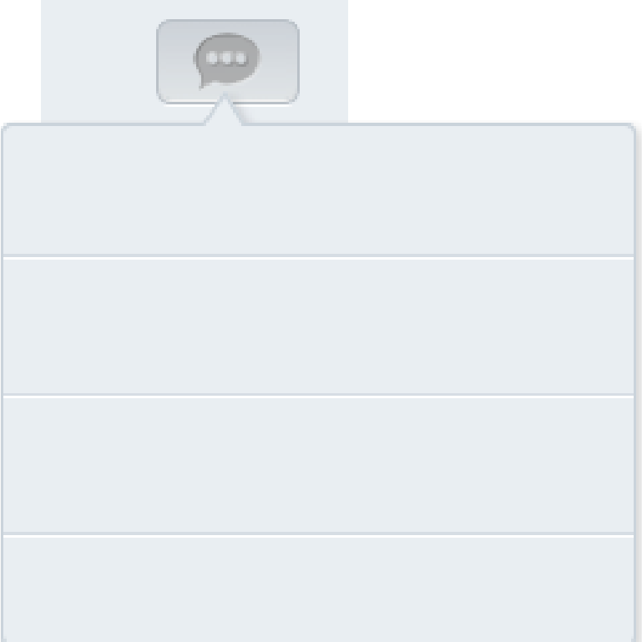
The unobtrusive notifications are fantastic. We can improve the copy of most of the messages, but that can be an iterative process. The main thing missing with the messages is a place for users to find old messages — right no they sometimes go away just when you are getting around to reading them but before you actually do. It is a unsettling experience. We need a place for messages to go after the disappear.



We should add a messages button in the action bar along the top of the editing area.



When a message goes away, it swiftly animates into the messages button. It can still appear using a fast "pop" or slide out. It should appear near the top of the screen, near the message button.



Clicking on the button gives a log of the last n messages. If clicking on the original message caused an action, clicking on the logged message causes the same action.

We have the option of providing an in-depth message inside the message logs that fully explains what the message was all about. For instance, the "Extension Installed" message can explain how the add-on becomes installed and is auto-linked every time it is saved, it gets reloaded, etc.