

Firefox Input v3.0

Prepared by: Aakash Desai

October 20, 2010

Summary

Problem

Currently, Firefox Major Release users can only offer feedback with Mozilla via support.mozilla.com and Report a Broken Website. Both alternatives only give us insight into specific use cases such as “What are users having the most trouble with?” and “Which websites are users having issues with?”. In the latter option, Report a Broken Website is not triaged at all and does a poor job as a reporting and clustering mechanism. Meanwhile, input.mozilla.com has done an good job in garnering user feedback about our Betas relating less to specific problems and more to general issues. We’d like to incorporate a feedback mechanism that allows Firefox Major Releases users to offer feedback in a better and more general product-bettering-focused dashboard of information.

Goals

- 1. Create submissions pages on Firefox releases that allow users to submit broken site feedback, ideas and their experience ratings on specific aspects of the browser**
- 2. Uplift the generic feedback page to incorporate the “Report a Broken Site”, “Rate your Experience” and “Give Firefox an Idea” submission pages**
- 3. Create a new Input dashboard for Firefox releases and add it as another flavor of the general Input platform**

New Features

#	Feature	Priority
3.0.1	Rate Your Experience Submission	P1
3.0.2	Report a Broken Site Submission	P1
3.0.3	Idea Box Submission	P1
3.0.4	Input Flavors	P1

3.0.1 - Rate Your Experience

Submission Form (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/5082404746/>

Notable Features

1. User Agent Sniffing: To get better constructive feedback from the user, we only want Beta users on the latest Firefox Beta to submit information to us. So, the following actions should be performed to safeguard against improper submissions.
 - a. Latest Firefox Release: Direct the user to the correct submission page
 - b. Firefox Release older than 4.0.x: Direct the user to a page with a download link to get the latest Firefox release build
 - c. Firefox Releases: Direct the user to the correct submission page
 - d. Non-Firefox Browsers: Direct the user to a page with a download link to get the latest Firefox release build
2. Submission: A HTTP POST, with the following fields, should be sent to the production database. The user should then be directed to a "Thank You" page
 - a. Sentiment: A new sentiment category, called "rate", should be submitted.

- b. Rating Input: The user should be able to give a positive or negative piece of feedback by clicking “happy” or “sad”.
 - c. Aspect Type: The message should send an aspect category (i.e. either “speed”, “stability”, “compatibility” or “features”).
 - d. Build Id: The user-agent will be a part of the form header which will be parsed by Input and added as an element to the entire feedback row entry.
3. Thank You Page Follow-Up: On submission of the feedback, the user should be re-directed to the “Thank You” page.

Trends Dashboard (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/5082404808/>

Notable Remarks

1. Controls
 - a. When: The user should be offered two button options: “7D”, or seven days, or “Release” where each refer to the number of feedback received within a within the given time period. When checked, the page should update with four updated “Aspect” graphs that relate to the set of feedback for the selected time period.
 - b. Platform: A grouping of checkboxes where each is assigned to a specific platform seen on our database of feedback. When checked, the page should update with four updated “Aspect” graphs that relate to the set of feedback for the selected platform.
 - c. Locale: A grouping of checkboxes were each is assigned to a specific locale seen on our database of feedback. When checked, the page should update with four update “Aspect” graphs that relate to the set of feedback for the selected locale.
2. Feedback v. Time Line Graphs per Aspect Type
 - a. Graph: We’ll be using a DHTML-based AJAX widget called Timeplot. This is the same widget used in Input 2.0, but whose controls are tied to the LHS sidebar controls of the dashboard.
 - i. Time Axis (X): The period of time (notated either per hour or by day) for which feedback was received. 24 hours should be selected by default.
 - ii. Submission Axis (Y): Number of feedback received notated by every 100 messages.
 - iii. Mouse Hover Effects:
 1. Cursor inside the Graph: Once a user hovers over an hourly/daily point on the graph, the messages received in between that hour/date and the previous hour/date should be shown in a box colored the same as its line. An example of this can be found on AMO’s statistics dashboard for AbBlockPlus.
 2. Mouse outside the Graph: The on hover effects should disappear from view on the graph

3.0.3 - Report a Broken Site

Submission Form (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/5081810611/>

Notable Features

1. User Agent Sniffing: To get better constructive feedback from the user, we only want Beta users on the latest Firefox Beta to submit information to us. So, the following actions should be performed to safeguard against improper submissions.
 - b. Latest Firefox Release: Direct the user to the correct submission page
 - c. Firefox Release older than 4.0.x: Direct the user to a page with a download link to get the latest Firefox release build
 - d. Firefox Releases: Direct the user to the correct submission page
 - e. Non-Firefox Browsers: Direct the user to a page with a download link to get the latest Firefox release build
3. Submission: A HTTP POST, with the following fields, should be sent to the production database. The user should then be directed to a "Thank You" page
 - a. Sentiment: A new sentiment category, called "brokensite", should be submitted.
 - b. Text Input: The user should be able to give mostly free-form feedback (profanity is blocked). There will be a 120 character max limit just like on the happy/sad forms.
 - c. Build Id: The user-agent will be a part of the form header which will be parsed by Input and added as an element to the entire feedback row entry.
 - d. URL Input: We want to grab the website users have issues on when they submit broken site feedback. So, to do this, the submission form needs to offer a required text field for users to give us that URL.
 - e. Constraints: The normal client/server side constraints we put on the URL and comment field in the "sad" form should apply here as well.

Broken Sites Dashboard (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/5095120090/>

Notable Remarks

1. Controls
 - a. When: The user should be offered three button options: “7D”, or seven days, “30D”, or 30 days, or “Release” where each refer to the time period in which to siphon or show feedback. For example, if a user selects “7D”, the total set of per-domain-siphoned clustered messages categorized under “brokensite” and received over the past 7 days (168 hours) should be shown.
 - b. Platform: A formulated group of checkboxes where each is assigned to a specific platform seen on the feedback received over the specified release. When checked, the page should update with a new Broken Site and Uncommon Broken Site List that relates only to the selected platform.
2. Broken Sites List: This list should be created and outputted using the same siphoning and clustering approach already offered in “site” for the beta dashboard.
 - a. Domain Link: A hyperlink for the top-level domain for which the feedback was submitted. When clicked, the user should be sent to a list of all clustered messages for the domain.
 - b. Messages Link: A hyperlink for the messages sent in under the respective domain. When clicked, the user should be sent to a list of all messages for the domain.
 - c. Clustered Broken Site Comments: A list of the top three clustered messages submitted for the specific domain. A “xx similar” hyperlink should be offered on the same line. On click of “xx similar”, the user should be shown a page of comments listed for the specific clustered message.
3. Uncommon Broken Sites Snippet: A list of hyperlinks of the first ten domains that could not be clustered.
 - a. Domain Link: Hyperlinks that directs the user to a search result page with the URL and message clicked on.
 - b. More Sites Link: A hyperlink that directs the user to a results page of all unclustered sites and their messages.

Broken Sites Search Results (Webpage)

Notable Remarks

1. Controls
 - a. When: After a search, the “When” option highlighted should modify the search results to only show domains and clustered messages relating to that period of time.
 - b. Platform: After a search, the “Platform” option selected should modify the search results to only show domains and clustered messages relating to that OS platform.
2. Broken Sites List: The list should be populated on domains that match the search term submitted. The same siphoning and clustering approach already offered in “site” for the beta dashboard.
3. Uncommon Broken Sites Snippet: A list of hyperlinks of the first ten domains that match the search term submitted.
 - a. Domain Links: Hyperlinks that direct the user to a search result page with the URL and message clicked on that match the search term submitted.
 - b. More Sites Links: A hyperlink that directs the user to a results page of all unclustered sites and related messages that match the search term submitted.

Uncommon Broken Sites List (Webpage)

Notable Remarks

1. Controls:
 - a. When: The user should be offered three button options: “7D”, or seven days, “30D”, or 30 days, or “Release” where each refer to the time period in which to siphon or show feedback. For example, if a user selects “7D”, the total set of un-clusterable sites and messages categorized under “brokensite” and received over the past 7 days (168 hours) should be shown.
 - b. Platform: A formulated group of checkboxes where each is assigned to a specific platform seen on the feedback received over the specified release. When checked, the page should update with a new Uncommon Broken Site List that relates only to the selected platform.
2. Uncommon Broken Site List: A list of all unclustered sites and their messages. Within the list, neither the domain link nor the message should be linkable.

3.0.3 - Idea Box

Submission Form (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/4923931517/>

Notable Features

1. User Agent Sniffing: To get better constructive feedback from the user, we only want Beta users on the latest Firefox Beta to submit information to us. So, the following actions should be performed to safeguard against improper submissions.
 - a. Latest Firefox Release: Direct the user to the correct submission page
 - b. Firefox Release older than 4.0.x: Direct the user to a page with a download link to get the latest Firefox release build
 - c. Firefox Releases: Direct the user to the correct submission page
 - d. Non-Firefox Browsers: Direct the user to a page with a download link to opt-in to our Beta Program
3. Text Counter: A user will be able to submit 250 characters (double the max limit of happy/sad comments). Instead of only applying a hard count, we'll implement a soft count of 200 characters (i.e. about 33 words) for people to comment on. To do this, a countdown-ing (from 200) counter should be shown in green (Hex #0AC92B). As the soft-200 is reached, the next 25 characters should be colored orange (Hex #FF4500) and then the last 25, red (Hex #8B0000).
4. Submission: A HTTP POST, with the following fields, should be sent to the production database. The user should then be directed to a "Thank You" page
 - a. Sentiment: A new sentiment category, called "Idea", should be submitted.
 - b. Text Input: The user should be able to give mostly free-form feedback (profanity is blocked). Unlike the 120 character max limit on happy/sad feedback, we will provide a max of 250 characters to users to submit their thoughts.
 - c. Build Id: The sniffed user-agent should be added to a separate, but hidden field on the form and submitted with the rest of the fields.

Dashboard (Webpage)

Mock: <http://www.flickr.com/photos/aakashhdesai/5081810665/>

Notable Remarks

1. Controls
 - a. When: The user should be offered three button options: “7D”, or seven days, “30D”, or 30 days, or “Release” where each refer to the time period in which to siphon or show feedback. For example, if a user selects “7D”, the total set of clustered messages categorized under “idea” and received over the past 7 days (168 hours) should be shown.
 - b. Platform: A formulated group of checkboxes where each is assigned to a specific platform seen on the feedback received over the specified release. When checked, the page should update with a new Idea List that relates only to the selected platform.
 - c. Locale: A formulated group of checkboxes where each is assigned to a specific locale seen on the feedback received over a specified release. When checked, the page should update with a new Idea List that relates only to the selected platform.
2. Idea List: This list should be created using the same clustering approach already offered in “themes” for the beta dashboard and outputted in the same format.
 - a. Clustered Idea: A message should be the main clustered “idea” where other messages were found to be similar.
 - b. Similar Message Link: A hyperlink that links to a list of all similar messages found in the clustering process. The order of the list should be determined by similarity to the top-level clustered idea where the first message in the list is the most similar to the clustered idea and the last is the least similar.
 - c. List of Similar Messages: A list of the top 5 similar messages found compared to the clustered idea.

3.0.4 - Input Flavors

Sub-Domain and Directory Structure

Notable Remarks

1. Desktop: When a user goes to input.mozilla.com, they should be directed to the major release (i.e. “/release”) in the input.mozilla.com domain. Current User agent sniffing rules should still stay the same.
 - a. Beta: The main dashboard should be located at the URL: input.mozilla.com/beta. From there, theme’ and sites’ URL’s should be located in directories one step lower. For example, input.mozilla.com/beta/themes and input.mozilla.com/beta/sites respectively.
 - b. Releases: The main dashboard should load by default at the URL: input.mozilla.com/. From there, ideas’ and brokensites’ URL’s should be located in directories one step lower. For example, input.mozilla.com/ideas and input.mozilla.com/brokensites respectively.
2. Mobile: It should have the sub-domain m.input.mozilla.com and with the following directory structure. Current User agent sniffing rules should still stay the same.
 - a. Beta: The main dashboard should be located at the URL: m.input.mozilla.com/beta. From there, stats’ and trends’ URL’s should be located in directories one step lower. For example, m.input.mozilla.com/beta/stats and m.input.mozilla.com/beta/trends respectively.
 - b. Releases: The main dashboard should load by default at the URL: m.input.mozilla.com/. From there, stats’ and trends’ URL’s should be located in directories one step lower. For example, m.input.mozilla.com/stats and m.input.mozilla.com/trends respectively.

Dashboard (Webpage)

Notable Remarks

1. Dashboard links: There should be a set of two button links at the bottom of the page, within a footer, that allows the

Other Notables

Bug 605539 - Make our database dumps downloadable

Bug 583669 - Show cluster reports of previous betas