Firefox for iOS Coding Exercise

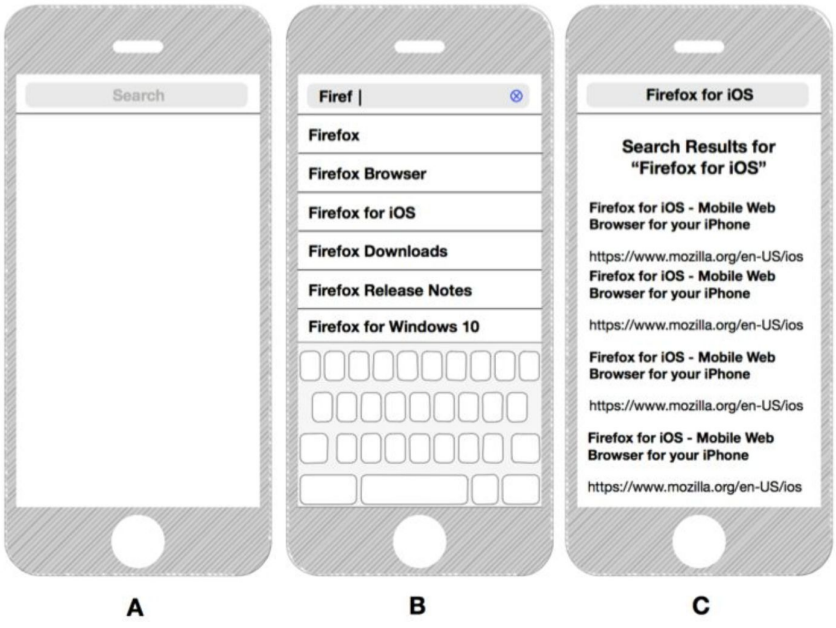
Thank you for taking time to interview for a position at Mozilla. To assess your coding skills, we've designed a coding exercise to be completed on your own as part of the interview process.

The goal of this exercise is to give you the opportunity to demonstrate your coding skills without the pressure of someone watching over you while you work. We expect that this exercise will take a couple of hours to complete, but if you find it's taking you longer, please stop and we will just discuss the parts you completed. Instead of "grading" you on this exercise, we will use it as a starting point for discussion during your next video screen, so you will have ample opportunity to explain your decisions.

Exercise Details

We would like you to build a very simple browser. Not a fully fledged one like Firefox, but instead a minimal application that can only use very simple *searches* and *search suggestions* as a browsing starting point.

The following three screens show the complete flow of the application.



A) Empty State

When the application is started it will show just a search field at the top of the screen with a placeholder text that says “Search”.

B Entering a Search Query and displaying Search Suggestions

When the user taps in the search field, it becomes an editable text field and the user can start entering a search query.

While the user types, execute a HTTP request to the Bing Search Suggestions API and show the returned Search Suggestions below the search field.

The Bing Search Suggestions API is a very basic JSON API. It has a main endpoint that looks like this: https://api.bing.com/osjson.aspx?query=firefox

Which returns a JSON array with the first item being the original query and the second item being another array listing all the suggestions. For example, for firefox, it will return the following: `["firefox",["firefox","firefox download","firefox for windows 10","firefox download windows 7","firefox windows 8","firefox free download","firefox browser","firefox 64 bit","firefox.com","firefox mozilla","firefox update","firefox download for windows 10"]]`

Tapping the little x on right will clear the current value of the search field and reset the search suggestions.

Note that there is no Cancel button. This is a shortcoming of the UI to simplify things.

C Display Search Results for the chosen Search Suggestion

When the user taps on a Search Suggestion, the following happens:

- The search suggestions and keyboard are hidden so that the webview is fully exposed
- The search field will now display the chosen search suggestion
- A Bing search with the chosen Search Suggestion is loaded in the web view

For example, if the user chose the “Firefox for iOS” search suggestion, you would load

https://www.bing.com/search?q=Firefox%20for%20iOS in the web view.

When the user taps the Go (or Enter) button on the keyboard, the same thing will happen, except that not a Search Suggestion is used but instead simply the search text that the user was typing into the search field.

Result Delivery

You can send us a .zip file that contains the project.

Implementation Notes & Hints

If you have questions, please contact us. Feel free to ask us anything to clarify the exercise.

Xcode & Simulator vs Device

Please use the latest production Xcode. We will only run the application in the latest non-beta Simulator. Don't worry about running on a real device.

External Libraries

Please do not use any external libraries and just use what UIKit, Core Foundation and the other iOS frameworks have to offer.

ObjectiveC or Swift

You can use either ObjectiveC or Swift. But do know that we love Swift.

UI Polish

For this exercise we are more interested in code quality than UI polish. Prefer functionality over being pixel perfect or using animations and transitions.

Browser Navigation

You can keep the embedded web view very simple. The WKWebView have all kinds of hooks for navigation and progress reporting. It is OK to not use those. If you feel adventurous and have time, feel free to add some additional browsing UI to the application. But this is not a requirement.

Error Handling

Making network requests, parsing JSON and loading pages in a web view can all result in errors. Don’t go too fancy with reporting. If you display errors in a simple alert that is great. Please do not ignore errors.

Storyboards & Auto Layout

Do whatever you are comfortable with. Either build the UI in Xcode’s Interface Builder or build up the UI programmatically in your code. There is no preference for this exercise.

Device Orientation & Size Classes

You can simplify the application by just supporting iPhone in portrait mode. Or, if you want to go a step further, support all orientations by simply scaling the UI. No special UI for iPad or landscape is asked for. Keep it simple.