

What is your project idea?

Implement [rails/jquery-ujs](#) using native JavaScript with other improvements

The goal of this project is to

1. Rewrite UJS using native JavaScript, i.e., without jQuery.
2. Add new features to UJS. Currently I have planned
 - a. Asynchronous file upload.
 - b. User-defined rails.confirm with asynchronous support.
 - c. Better organization of the source code, modularized into multiple files.
 - d. Making UJS available in NPM package.

Why we should choose you for this project?

First, I believe I'm fully capable of this project. I have not only been using Ruby on Rails in many projects for a long time, but also done a lot of JavaScript development with various kind of libraries including jQuery, Angular, React and Vue.js. I'm familiar to web development and web developers, knowing how a good library should be.

Second, I understand the idea behind UJS and appreciate it very much. UJS is designed to be unobtrusive, delicate and graceful. By "unobtrusive", developers write minimum codes, mostly no JavaScript, to make everything work. And it will do nothing if developers write nothing. "Delicate" means UJS is very small and only contains the very common features, so people can use it without any worry. By "graceful", users never encounter a failure whatever happens. If someone is using an unsupported browser, UJS just behave as if it never exists, and the user can still use most functionality.

After I got to know more about web components and Angular, I found that UJS is similar to Angular's directive. They both enhance a DOM element by adding attributes to it. The difference is that Angular is a heavy framework while UJS is very lightweight. Is it possible for UJS to become a common directive library?

At last, I'll have plenty of time during summer. I have no exam and no course this summer and I'll graduate at June. I don't have any other plan this summer.

Why do you think this idea is worth doing?

This project will help to reduce the ever increasing page bloat, remove a gem from the Ruby load path and simplify our release process so we could bundle the library with Rails. Users of Webpack and Browserify will also benefit from the project.

Rewrite UJS using native JavaScript

There are two main reasons that UJS should be rewritten in native JavaScript. First, JavaScript support in modern browsers has improved a lot in last five years with

standardized APIs such as `document.querySelector`, making it possible to eliminate the dependency on jQuery. Second, more and more modern websites use MVC frameworks like AngularJS instead of jQuery, which means adding UJS would introduce a large library unnecessarily.

Asynchronous file upload

[Currently UJS doesn't support async file upload](#). In the past this feature would require using some complex solutions such as `iframe` or `flash`. However, thanks to the new Web APIs, we can now use [FormData](#) to achieve this.

User-defined `rails.confirm` with asynchronous support

Currently `rails.confirm` is configurable, but only in synchronous way, which mostly limits the usage to `window.confirm`. Changing this function into asynchronous way makes it possible to use custom confirm box. Note this task is not as trivial as it seems to be because most `rails.confirms` are called in a event-handling context, which means we need to return `false` first, then fire the event again if needed.

Better organization of the source code, modularized into multiple files.

Currently `rails.js` is a single file with about 550 lines. It should have a better organization by splitting it into multiple modules.

Making UJS available in NPM package.

UJS is developed for the Ruby on Rails framework, but is not strictly tied to any specific backend. Making it available in NPM package would benefit Webpack users and Browserify users a lot.

What are your plans for the summer?

I have no exam, no course and no other plan for the summer and I'll devote at most 40 hours a week to this project during the summer.

However, Peking University's summer starts at around early June. So I may not have enough time the first few weeks.

What is your expected timeline?

Deadline	Outcome
May 23, 2016	A brief report that shows how much current UJS depends on jQuery with answers to the questions below.
May 30, 2016	Work out solutions to replace jQuery with native JavaScript.
June 20, 2016	The migration should be finished with all existing tests passed.

July 4, 2016	Support for asynchronous file upload
July 18, 2016	User-defined rails.confirm with asynchronous support.
August 1, 2016	Better organization of the source code.
August 15, 2016	UJS should be available in NPM package.

Do you have plans for if you finish sooner than expected? How about slower than expected?

It's likely that I will come up with new ideas of how to improve UJS during my working on it so I won't worry if I finish sooner than expected.

I'll try to make sure that the project won't be finished later than expected. But in case it happen, I will first make sure the main purpose of this project is achieved, i.e., implement current functionality with native JavaScript first, then add other features.

How do you expect to accomplish your project idea? (do you have a plan? need help planning?)

1. Research and sort out how much current UJS depends on jQuery. Determine the following questions
 - a. What features in jQuery does UJS use?
 - b. For each of these features, what are the alternative solutions? What browsers do they support?
 - c. What browsers should we support? i.e., To which version of IE should we support?
2. Rewrite UJS in JavaScript. This could be done in following two ways.
 - a. If the feature is already available in modern browsers' APIs, just replace them with native JavaScript API calls, and make some small changes when necessary. These may include
 - i. `document.querySelector`
 - ii. `$.ajax()`
 - iii. Event system / custom events
 - b. If the feature is not universally available in all supported browsers or behaves differently in different browsers, just write a small wrapper and try to provide same APIs as jQuery does. These may include
 - i. `form.serialize()`
 - ii. Data storage on elements
3. Add new features to UJS, currently including
 - a. Asynchronous file upload.
 - b. User-defined rails.confirm with asynchronous support.
 - c. Better organization of the source code, modularized into multiple files.
 - d. Making UJS available in NPM package.