

## Redmine - Feature #30486

### Convert all jQuery to ECMA Script

2019-01-23 21:52 - Anonymous

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Code cleanup/refactoring	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			
<b>Description</b>			
<p>I am going to just create this ticket, it's sounds quite general, but I would like to see how many will agree with me here, so in the future, it could be branched up into separate issues.</p> <ol style="list-style-type: none"><li>1. I think java script in Redmine is inconsistent, many things are written in bare JS and many other things are written in bare jQuery.</li><li>2. ECMA Script brings new functions for easy DOM manipulation, please see You Dont Need jQuery by nefe[[<a href="https://github.com/nefe/You-Dont-Need-jQuery">https://github.com/nefe/You-Dont-Need-jQuery</a>]]</li><li>3. I think the set of front end functions deserves a good rethinking, in order to redesign most of them with re usability in mind as much is possible, so plugins developers could reuse them when necessary, the set of functions will however need to be documented.</li></ol> <p>For example, quite a versatile HTTP request function I made for my WebGL plugin:</p> <pre>function httpRequest(method, url, body, callback) {   let request = new XMLHttpRequest();   request.open(method, url, true);   request.setRequestHeader('Content-type', 'application/json');   request.onload = function () {     if (request.status &lt; 200    request.status &gt; 299) callback("Error: Status " + request.status + " on resource " + url);     else callback(null, request.responseText);   }   request.send(body); }</pre> <p>etc etc :)</p>			

#### History

##### #1 - 2019-02-10 05:17 - Enziin System

In the future, I think Redmine need convert to API only at the backend and React/Vue/Angular in the frontend. Rails 6 use webpacker and it does not support assets pipeline.

##### #2 - 2019-02-10 17:53 - Anonymous

Wouldn't make a lot of sense to just swap a library with a framework, or would it?

I think, judging by the pattern the vanilla JS has been evolving in recent years, it's safe to say that it will continue to get influenced by React, Vue and Angular, meaning things will get easier even in vanilla JS. IMHO, Redmine should not use front-end frameworks or libraries at all and encourage new contributors and developers to instead think for themselves more than just use some lame solutions premade for them. I believe a zillion of front-end JS frameworks today is the reason why many till this day thinks of a front-end engineer in JS as of someone who is a noob exactly for that reason. XD

I actually share a lot with what Adrian Holovaty had to say on the situation: <https://youtu.be/k7n2xnOiWl8>.

If front-end frameworks and libs are absolutely needed by someone, then they can include it with their plugin and write an appropriate code which will use their desirable framework or a lib, but for the core, encouraging vanilla seems like the best longer term solution since now.

And besides, things are already in there written in a bare vanilla JS, e.g. **jstoolbar.js**