

## Redmine - Feature #35415

### Unified plugin api esp. regarding patches

2021-06-14 08:41 - Ludwig Meysel

<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> Plugin API	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>Resolution:</b>	
<b>Description</b>	
<p>Many plugins use different ways to include own lib-code. A common topic I came across is applying patches and consuming hooks. Sometimes the files are require'd having something like</p> <pre>unless FoobarController.included_modules.include? MyPlugin::Patches::FooControllerPatch   FooController.send(:include, MyPlugin::Patches::FooControllerPatch) end</pre> <p>in the bottom.</p> <p>Sometimes the require is wrapped in Reloader's to_prepare-callback (which seems quite useless to me), other plugins have some kind of apply_patch-method implemented, which does the same as the code above in a more generic manner (IMO the most elegant way to go).</p> <p>While all these techniques basically work, it often invites to mess up with the auto reloader. When doing some tweaks in the codebase (or even develop my own plugin), I either have to disable foreign plugins or painfully restart the dev-server with every change, when autoloading is not properly implemented. Otherwise I face the often reported problem of "Cannot autoload Constant ..." -Error.</p> <p>My suggestion is to provide a more unified API for applying patches and consuming hooks.</p> <p>I could imagine a simple approach e.g. of an apply_patch function, which forwards the params to an array, which then is processed internally by the ActiveSupport::Reloader.to_prepare callback.</p> <p>Another (more opinionated) approach is to grab all patches from plugins/.../lib/patches/*.rb, after a plugin was registered, which also was applicable to the hooks.</p> <p>Anyways, I think there should be some mechanism to provide plugins safely, which do not mess up with the autoloader. This could also catch compatibility-stuff I have also often seen like Rails.version &lt; '5.1' ? ActionController::Callbacks : ActiveSupport::Reloader.to_prepare.</p>	