

ChiliProject - Feature # 662: Project path formatting

Status:	Open	Priority:	Normal
Author:	Jeff Mitchell	Category:	
Created:	2011-10-16	Assignee:	
Updated:	2011-10-17	Due date:	
Remote issue URL:			
Affected version:			
Description:			
This is recapping a discussion between myself, Ben Cooksley and Holger Just in IRC.			
Ben has written a patch that has the full project path in the URL...you can see it in action at http://projects.kde.org ; for instance, https://projects.kde.org/projects/extragear/network/rekonq/repository			
This is functionality I'd find useful in my two installs at work, and I'm willing to bet that others would too. In prior discussions with Ben, the view of the CP devs was that it was too difficult to figure out how to not have conflicting paths. For instance, is <code>/foo/bar/repository</code> the project "repository" as a subproject of "bar", or the actual repository module?			
A blacklist was not deemed a good idea as plugins or updates to CP could cause the need for more values in the blacklist, which could then conflict with existing projects.			
Also not deemed a good idea were prepending or appending a module's part of the URL with some character to indicate that it's a module; or, putting the project path last.			
However, I believe that an acceptable solution would be to separate out the components of the path using colons. Colons are natural separators (and are familiar to coders) and I think that they don't look too bad. Another benefit of using this approach is that it's also backwards-compatible; new subprojects can have a slug using colons as separators, and existing subprojects can keep their existing slugs but also be allowed to be referred to via colons.			
In this scheme, the URL above becomes https://projects.kde.org/projects/extragear:network:rekonq/repository . To me, it's quite easy to see the project path in that URL, and aesthetically it's not too bad either.			

History

2011-10-17 05:19 am - Felix Schäfer

One big problem with trying to put only "part" of the path/name of a project into the DB: how do you handle the integrated authentication? If you call the repo just `@foo@`, how do you know it's the project `@foo@` under `@bar@` or under `@baz@`, or maybe even the top-level one? So you'd have to call the repos `@bar:foo@`, `@baz:foo@` and `@foo@`, and as you'd have to put the full "name"/path everywhere you want to reference the project anyway, so you can go on an save that exact name in the DB.

Now, if your only irk is to have some sort of way of knowing what parent projects a project has based on its parents, have a look at "`@ensure_project_hierarchy@`" (on GitHub): https://github.com/thegcat/chiliproject_ensure_project_hierarchy, the plugin ensures that the identifier of a subproject of `@bar@` must begin with `@bar_@`, effectively implementing a "path" of sorts, except the separator is `@_@` instead of `@:@`.