

Chamilo LMS - Feature #5023

Implement Doctrine for database access

20/06/2012 14:09 - Laurent Opprecht

Status:	New	Start date:	20/06/2012
Priority:	Normal	Due date:	
Assignee:		% Done:	10%
Category:		Estimated time:	0.00 hour
Target version:	3.0	Spent time:	0.00 hour
Complexity:	Challenging	SCRUM pts - complexity:	?
Description			
Related issues:			
Related to Chamilo LMS - Feature #4912: tool course description: allow for n...			New 14/06/2012

History

#1 - 20/06/2012 14:43 - Laurent Opprecht

I did a dry run:

- I removed the c_ from course table to have cleaner names for entities: c_announcement becomes announcement. One issue is that c_group clashes with group. This could be changed to CourseAnnouncement
- don't know where to put those so for the moment they are located under inc/entity - there are quite a bit of classes generated so it would not make sense to have them in inc/lib -
- I put metadata files under inc/entity/metadata and repositories files under inc/entity/repository to avoid having too many files under the same folder
- I kept on using "table" casing for property/method names i.e. get_id instead of getId
- there is a table called class and this obviously create an issue since this is a reserved word - would need to manually change the name of the class to something else - note that the table name can stay.
- Doctrine does not support auto assigned numbers for multiple keys. So for course tables (they have two keys: id, c_id) it is not possible to do an insert without setting the value for id. I went on doing (select max(id)+10 where ...) in a pre save method. But this is not 100% bullet proof, if somebody tries to make an insert at the same time in the same course this will fail. Another option would be to introduce a new type of entity generator but that would require some kind of modifications to the metadata factory at best - which could be done by creating our own. By the way I think that autoassign number for multiple keys work only for the ISAM engine.
- Beside a simple mapping one could add relationships to the object model. For example c_announcement would have a property called course of object course. With this scheme you don't provide get_c_id but replaces it by get_course. Doctrine does the work of lazy loading, optimization etc. If you want to set the value of c_id you use set_course instead and provide a real object course. Now I don't believe this is possible because c_id is part of the primary key.
- Other kind of relation ship mapping would need to be done manually - for example forum_category to forum_forum etc

#2 - 20/06/2012 14:48 - Laurent Opprecht

As a matter of fact I didn't do this by hand but hooked into the code generation engine of doctrine/symphony

#3 - 22/06/2012 16:03 - Laurent Opprecht

commit:9655bc9c2736
commit:6e4ed53dae3b
commit:cb1928f6841c
commit:2dbb801540a8

#4 - 22/06/2012 16:04 - Laurent Opprecht

- % Done changed from 0 to 10

#5 - 22/06/2012 16:05 - Laurent Opprecht

- Complexity changed from Normal to Challenging

#6 - 11/07/2012 07:31 - Yannick Warnier

- Target version set to 2.0

#7 - 31/01/2016 23:36 - Yannick Warnier

- Target version changed from 2.0 to 3.0