Subject: metadata vs. business logic
Posted by lauxa on Thu, 15 Jun 2006 21:27:09 GMT
View Forum Message <> Reply to Message

I am just getting started in php application development and I really like how you have set up your application framework. I especially like that your data array handles one or multiple records and without breaking it into $<field> variables as some reccommend.

I want to use a perl script to generate my table classes from the create sql script. I am considering making my meta-data definitions separate from any business logic and then using multiple inheritence to include them in the $<table> class so that if my table structures change I can regen the meta-data definitions with minimum impact. I just wanted to get your opinion on this and why you did not design it this way. Maybe changing table specifications is an issue that only affects inexperienced programmers like me? Thanks.

Subject: Re: metadata vs. business logic
Posted by AJM on Fri, 16 Jun 2006 09:12:20 GMT
View Forum Message <> Reply to Message

It is a good idea to keep table definitions separate from business logic so that you can incorporate any changes in your database structure very easily, but the Radicore framework already achieves this. When you use the dictionary export facility (see http://www.tonymarston.net/php-mysql/menuguide/dict_table(up d4)export.html) it can create two files - the table class for business logic and a separate structure file. It does not overwrite the class file if it already exists, thus preserving any business logic which you may have defined. Only the structure file is overwritten.