Zend_Test_PHPUnit_Database - Benjamin Eberlei

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: Zend_Test_PHPUnit_Database Component Proposal

Proposed Component Name	Zend_Test_PHPUnit_Database
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/Zend_Test_PHPUnit_Database
Proposers	Benjamin Eberlei
Zend Liaison	Matthew Weier O'Phinney
Revision	1.0 - 5 February 2009: Initial Draft. (wiki revision: 10)

Table of Contents

1. Overview

The PHPUnit Database Extension is very PDO centric and may hinder people using Zend_Db to effectivly write Database related tests. This proposal is for a subcomponent in the Zend_Test namespace that implementes the necessary interfaces of the PHPUnit_Extensions_Database to make this tool work with Zend_Db_Adapter_Abstract connections.

It would also allow for a very simple integration of a Zend_Db_Table implementation as a dataset which would make testing against Zend_Db_Table and corresponding Zend_Db_TableRowset instances a bit more convenient.

Additionally the profiler does much work on counting queries and stuff, which could also be integrated as additional assertions.

2. References

- PHPUnit Database Extension
- Extension Authors blog with recent information

3. Component Requirements, Constraints, and Acceptance Criteria

- This component **MUST** allow using Zend_Db_Adapter_Abstract in conjunction with PHPUnits database extension.
- This component **MUST** allow using Zend_Db_Table implementations as DataSets to test against.
- This component **MUST** add new constraints for use with the Zend_Db_Profiler

4. Dependencies on Other Framework Components

- Zend_Db_Adapter_Abstract
- Zend_Db_Table
- PHPUnit

5. Theory of Operation

Operation with this component would be exactly the same as in the current PHPUnit Database Extension.

You seed the database with a default dataset and perform operations the database as you would in normal production enviroment. At any stage you can assert weater two datasets are equal, which proves test-success.

Step by step:

1. At each testrun, PHPUnit clears the database and refills it with the Seed Data you are giving to it.
2. You perform database operations.
3. You assert that the content of two datasets are the same, where a dataset is an abstract definition of rows in a table.

Datasets come from very different sources: Flat XML format, a more complex XML format, in the next version YAML, from other database tables and from CSV-Files.

PHPUnits Database Extension puts those into a common format and allows them to be comparable so that you can check for example weater the contents of a database table equal the contents in a given XML file after you performed some operations in the test-case.

6. Milestones / Tasks

- Milestone 1: Proposal, Community Review and Acceptance
- Milestone 2: Unit-Testing and Development
- Milestone 3: Documentation and Testing

7. Class Index

- Zend_Test_PHPUnit_DatabaseTestCase
- Zend_Test_PHPUnit_Database_Connection
- Zend_Test_PHPUnit_Database_DataSet_ZendDbTable
- Zend_Test_PHPUnit_Database_DataSet_RowSet
- Zend_Test_PHPUnit_Database_Metadata_Generic
- Zend_Test_PHPUnit_Database_Operation_*

8. Use Cases

9. Class Skeletons

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>