PDO – PHP Database Objects

Amsterdam, June 16th 2007

Dutch PHP Conference

Lukas Kahwe Smith (lsmith@optaros.com)

http://pooteeweet.org/files/dutchphpconf07/pdo_comments.pdf

optaros

About Myself

- Lukas Kahwe Smith
- Started using PHP in the late PHP3 days
- Joined PEAR sometime in 2002
- Began posting on internals@lists.php.net in 2003
- Started the unofficial official PHP todo list in 2005
- Makes his living as a consultant for Optaros
  - OSS consulting and system integration firm
  - We are always hiring
    - http://optaros.com/en/company/careers
- Can be reached
  - Via Email: lsmith@optaros.com
  - On IRC: “lsmith” on efnet, freenode, ircnet ..
  - In Person: Here or at Ultimate Frisbee tournaments

optaros

Why abstract your database interface?

- Support for multiple RDBMS
- Forward compatibility with new RDBMS versions
- Vendor Lock in
- Training costs
- Pushing your own preference to the client
- Higher level API

optaros

And now for the meat ..

- Actually Wez has so nice slides about PDO lets just use those
  - http://images.omniti.net/omniti.com/talks/furlong-pdo-long.pdf
- I will add a few more slides at the end about layers build on top of PDO
  - Actually I rarely use PDO directly at all
- Oh and did I mention we are also hiring?
  - http://optaros.com/en/company/careers

optaros

Types of abstraction layers?

- Database API Abstraction
  - Provide a common API
- Database SQL Abstraction
  - Provide a common API as well as methods to construct portable SQL
- ActiveRecord (and friends)
  - Wrap around schema to provide an OO interface
  - No need to write the standard day to day SQL
- ORM
  - Pure OO code that generates SQL
  - OO code defines SQL schema implicitly

optaros

Database SQL Abstraction and ORM Layers

- Raw SQL
  - `$sql = "SELECT * FROM Person WHERE name='Smith'";`
  - `$all_smiths = $db->getAll($sql);`
- Query-By-Example
  - `$user = new User();`
  - `$user->name = 'Smith';`
  - `$all_smiths = Query::execute($user);`
- Query-By-API
  - `$q = new Query();`
  - `$c = new EqCriteria('Name', 'Smith');`
  - `$q->Form('Person')->Where($c);`
  - `$all_smiths = $q->execute();`
- Query-By-Language
  - `$q = "FROM Person p WHERE p.name = 'Smith'"`
  - `$all_smiths = $conn->query($q);`

optaros

Database SQL Abstraction and ORM Layers

- ezc::Database
  - Raw SQL
  - Persistent Objects (via ezc:PersistentObject)
- Zend_DB
  - Raw SQL
  - Table Gateway Pattern (via Zend_DB_Table)
- Doctrine
  - Raw SQL, Query-By-Language and Query-By-API
- Propel 1.3
  - Raw SQL. Query-By-API
- MetaStorage
  - Raw SQL. Query-By-API (XML based meta programming language and code generator)