ITMD 462/562
Web Site Application Development

Lecture 10

Fall 2015 – November 4, 2015

Tonight’s Agenda

- Finish Database discussion and demos

PDO Connecting

- Create a database handle for specific connection string.

```
$dbh = new PDO("pgsql:dbname=no_database;host=localhost", "username", "password" );
```

- Should use a try catch

-
  ```
  try {
      # SQLite Database
      $DBH = new
  PDO("sqlite:my/database/path/database.db");
  } catch(PDOException $e) {
      echo $e->getMessage();
  }
  ```
- Close the connection by setting handle to null, `$DBH = null;`

PDO Exec & Query

- PDO::exec()
- Execute an SQL statement and return the number of affected rows
- PDO::query()
- Executes an SQL statement, returning a result set as a PDOStatement object or false on failure
- You can iterate over the PDOStatement Object to get the rows

- Better to use prepared statements if you can, especially for user entered data.

Exec

- /* Delete all rows from the FRUIT table */
  `$count = $dbh->exec("DELETE FROM fruit WHERE colour = 'red'");`

- http://php.net/manual/en/pdo.exec.php

Query

-

```
$sql = "SELECT animal_id FROM users";

/*** run the query ***/
$result = $dbh->query($sql);
```

http://php.net/manual/en/pdo.query.php

Prepared Statements

- Pre-compilied SQL statement that accepts 0 or more parameters
- Prepares the SQL for execution
- Helps prevent SQL injection by calling the PDO::quote() method internally
- PDO accepts two kinds of parameter markers.
  named - :name
  question mark - ?
  Choose one or the other. Cannot mix.
- Can repeate the statement with different values
- http://php.net/manual/en/pdo.prepared-statements.php

Prepared Statements

- /*** prepare the SQL statement ***/
  $stmt = $dbh->prepare("SELECT * FROM animals WHERE animal_id = :animal_id AND animal_name = :animal_name");
- /*** bind the paramaters ***/
  $stmt->bindParam(':animal_id', $animal_id, PDO::PARAM_INT);
  $stmt->bindParam(':animal_name', $animal_name, PDO::PARAM_STR, 5);
- /*** execute the prepared statement ***/
  $stmt->execute();
- Fetch the results on the statement object
  - $stmt->fetch();
  - Should set the fetch mode first with $stmt-> setFetchMode(PDO::FETCH_ASSOC);

Examples

- Let’s look through a couple web examples
- http://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059
- http://www.phpro.org/tutorials/Introduction-to-PHP-PDO.html

- PHP PDO Section in docs
- http://php.net/manual/en/book.pdo.php

Assignments

Reading

- If you have not taken those courses you need to read/watch one or more of the following tutorials
  - https://www.codeschool.com/courses/try-sql
  - http://www.w3schools.com/sql/default.asp
  - http://www.tutorialspoint.com/sql/
- Read PHP PDO Tutorial
  - http://code.tutsplus.com/tutorials/php-database-access-are-you-doing-it-correctly--net-25338
- Attempt to create a SQLite Database and MySQL database using the database management applications we discussed in class.
- Good PHP resource -> http://wiki.hashphp.org/Main_Page

Assignments

- Assignment 2 is up and due this Sunday Nov 8 11:59 Chicago time.
- It should be very easy so there will be **NO EXTENSIONS**.
- Basically convert from serialized data files to SQLite PDO database for storage.
- Graduate students need to have very basic authentication