PHP-MySQL Documentation

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Data Structure Index

2.1 Data Structures

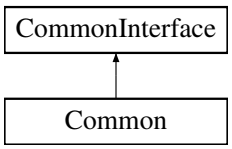
Here are the data structures with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 Common Class Reference

Inheritance diagram for Common:



Public Member Functions

- Common ($debug)
- connect ($db)
- executeQuery ($sql, $filename)

Data Fields

- **$conn**
- **$debug**
- **$db** ="database.cse.tamu.edu"
- **$dbname** ="jgwesterfield-TamuDriver"
- **$user** ="jgwesterfield"
- **$pass** ="Whoop19!"

3.1.1 Detailed Description

Common is responsible for actually connecting to the MySQL database. The Common class object must be created and the connect() function run before any functions on the database can be performed.

3.1.2 Member Function Documentation

3.1.2.1 Common (*$debug*)

Common constructor.

Parameters

$debug	The constructor for the Common class

Implements CommonInterface.

3.1.2.2 connect (*$db*)

Parameters

$db	Uses the $db class member functions to actually make the connection to the database

Implements CommonInterface.

3.1.2.3 executeQuery (*$sql, $filename*)

Parameters

$sql	
$filename	

Returns

mixed

Uses the sql statement that is passed in and executes the query

Example Code:
$sql = "SELECT COUNT(WalkerNumber) FROM WalkerData";
$rs = $this->COMMON->executeQuery($sql, $_SERVER["SCRIPT_NAME"]);
$row = $rs->fetch(PDO::FETCH_ASSOC);
return (int)$row['COUNT(WalkerNumber)'];

Implements CommonInterface.

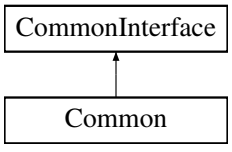
The documentation for this class was generated from the following file:

- PHP Server Code/CommonMethods.php

3.2 CommonInterface Interface Reference

Interface to for the CommonMethods.php file. Include this file in your code instead of the CommonMethods.php file.

Inheritance diagram for CommonInterface:



Public Member Functions

- Common ($debug)
- connect ($db)
- executeQuery ($sql, $filename)

3.2.1 Detailed Description

Interface to for the CommonMethods.php file. Include this file in your code instead of the CommonMethods.php file.

3.2.2 Member Function Documentation

3.2.2.1 Common (*$debug*)

Common constructor.

Parameters

$debug	The constructor for the Common class

Implemented in Common.

3.2.2.2 connect (*$db*)

Parameters

$db	Uses the $db class member functions to actually make the connection to the database

Implemented in Common.

3.2.2.3 executeQuery (*$sql, $filename*)

Parameters

$sql	
$filename	

Returns

mixed

Uses the sql statement that is passed in and executes the query

Example Code:
$sql = "SELECT COUNT(WalkerNumber) FROM WalkerData";

```
$rs = $this->COMMON->executeQuery($sql, $_SERVER["SCRIPT_NAME"]);
$row = $rs->fetch(PDO::FETCH_ASSOC);
return (int)$row['COUNT(WalkerNumber)'];
```

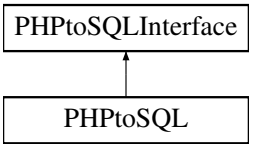
Implemented in Common.

The documentation for this interface was generated from the following file:

- PHP Server Code/CommonInterface.php

3.3 PHPtoSQL Class Reference

Inheritance diagram for PHPtoSQL:



Public Member Functions

- __construct ($db)
- __destruct ()
- getNumCarsThisWeek ($locationChoice)
- getTrafficByYear ($year, $locationChoice)
- getTrafficByMonth ($year, $month, $locationChoice)
- getTrafficByDay ($year, $month, $day, $locationChoice)
- getTrafficTimeRange ($year1, $month1, $day1, $year2, $month2, $day2, $locationChoice)
- getListOfLocations ()
- GetMinimumCountInHour ($countByHour, $dateByHour)
- GetMedianCountInHour ($countByHour, $dateByHour)
- GetMaximumCountInHour ($countByHour, $dateByHour)
- GetAverageCountInHour ($countByHour)
- ExecuteQuery ($Time1, $Time2, $COMMON)

  A function to execute a query of correct MySQL syntax for the specified instance of the database within a certain time range.
- ConvertTime ($Time)
- PrintResults ($results)
- GroupResultsByHour ($results)
- ReformatDate ($date)
- **GetDatesOfPastWeek** ()
- **GetTrafficCountForDates** ($results, $dates)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 __construct (*$db*)

DBAPI constructor.

Parameters

$db	Mostly sets up the dates in this object. Also sets the timezone to our timezone.

Implements PHPtoSQLInterface.

3.3.1.2 __destruct ( )

Class destructor

Implements PHPtoSQLInterface.

3.3.2 Member Function Documentation

3.3.2.1 ConvertTime (*$Time*)

Parameters

$Time	- timestamp of the format 'YYYY-MM-DDTHH:MM'.

Returns

string - which is of the form 'YYYY-MM-DD HH:MM:00'.

Converts a time of a certain format to that of another.

Implements PHPtoSQLInterface.

3.3.2.2 ExecuteQuery (*$Time1, $Time2, $COMMON*)

A function to execute a query of correct MySQL syntax for the specified instance of the database within a certain time range.

Parameters

$Time1	- First timestamp of the format 'YYYY-MM-DD HH:MM:00'
$Time2	- Second timestamp of the format 'YYYY-MM-DD HH:MM:00'.
$COMMON	- common instance that connects to the database.

Returns

array - all of the resulting data (in rows) returned from the query.

Usage:
$Time1 = $_POST["Time1"];
$Time2 = $_POST["Time2"];

```
// Convert input times into the correct Timestamp format
$time1Format = ConvertTime($Time1);
$time2Format = ConvertTime($Time2);
// Create a new Common instance to connect to the database
$debug = false;
$COMMON = new Common($debug);
// Execute query and fetch the results
$results = ExecuteQuery($time1Format, $time2Format, $COMMON);
```

Implements PHPtoSQLInterface.

3.3.2.3 GetAverageCountInHour (*$countByHour*)

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.

Returns

integer - Returns the average number of people recorded in an hour of the specific time range.

Finds average number of people counted during an hour in a specific time range.

Implements PHPtoSQLInterface.

3.3.2.4 getListOfLocations ( )

Returns

array of strings

Will return the names of all different locations (parking lots) that are being recorded in the database

Implements PHPtoSQLInterface.

3.3.2.5 GetMaximumCountInHour (*$countByHour, $dateByHour*)

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.
$dateByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.

Returns

2 arrays - Returns an array containing (1) the maximum count within an hour and (2) the corresponding time for which the count was recorded.

Finds the maximum count for an hour in a specific time range.

Implements PHPtoSQLInterface.

3.3.2.6 GetMedianCountInHour (*$countByHour, $dateByHour*)

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.
$dateByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.

Returns

array - Returns an array containing (1) the median count within an hour and (2) the corresponding time for which the count was recorded.

Finds the median count for an hour in a specific time range.

Implements PHPtoSQLInterface.

3.3.2.7 GetMinimumCountInHour (*$countByHour, $dateByHour*)

Parameters

$countByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.
$dateByHour	- the array of counts by hour returned from GroupResultsByHour.

Returns

Returns an array containing (1) the minimum count within an hour and (2) the corresponding time for which the count was recorded.

Finds the minimum count for an hour in a specific time range.

Implements PHPtoSQLInterface.

3.3.2.8 getNumCarsThisWeek (*$locationChoice*)

Returns

int

Gets the number of cars in the last week starting from today Usage: $numCars = getNumCarsThisWeek("lot35"); for Lot 35

Implements PHPtoSQLInterface.

3.3.2.9 getTrafficByDay (*$year, $month, $day, $locationChoice*)

Parameters

$year	
$month	
$day	
$locationChoice	

Returns

int array - length 24 array with the traffic number for each our of the day

Gets the traffic for each hour and returns it in a 24 element array.

Usage: < $dayArray = getTrafficByDay(2018, 2, 15, "lot35");> for Febraury 15, 2018 for Lot 35 output the times to see if I overshot how many times to iterate

Implements PHPtoSQLInterface.

3.3.2.10 getTrafficByMonth (*$year, $month, $locationChoice*)

Parameters

$year	
$month	
$locationChoice	

Returns

int array - array with the traffic number for each day of the specified month

Gets the traffic for each day during the specified month of the specified year

Usage: $monthArray = getTrafficByMonth(2018, 2, "lot35"); // for February 2018 for Lot 35

Implements PHPtoSQLInterface.

3.3.2.11 getTrafficByYear (*$year, $locationChoice*)

Parameters

$year	
$locationChoice	- string for the lot you want to put in

Returns

int array - a length 12 array where each element is the number of cars that passed through for that month. If there were 8 cars that passed through in April, the element at index 3 would be 8

Gives the traffic for each month in an array for the specified year passed in.

Usage: $yearArray = getTrafficByYear(2018, "lot35"); for 2018 for Lot 35

Implements PHPtoSQLInterface.

3.3.2.12 getTrafficTimeRange (*$year1, $month1, $day1, $year2, $month2, $day2, $locationChoice*)

Parameters

$year1	
$month1	
$day1	
$year2	
$month2	
$day2	
$locationChoice	

Returns

integer - the number of cars in the time range

Takes in a date range (start and end date) and counts the number of cars in the given range

Usage: $trafficInRange = getTrafficTimeRange(2018, 2, 15, 2018, 2, 28, "lot35"); for number of cars between 2/15/2018 and 2/28/2018 for Lot 35

Implements PHPtoSQLInterface.

3.3.2.13 GroupResultsByHour (*$results*)

Parameters

$results	- the results returned from a query, specifically from the function, ExecuteQuery().

Returns

array - Returns an array containing (1) an array of the dates within the results, grouped by hour, and (2) an array containing the counts for each corresponding hour of the dates array.

Groups the results of a query by hour and counts the number of results within each hour of the time range.

Implements PHPtoSQLInterface.

3.3.2.14 PrintResults (*$results*)

Parameters

$results	- the results returned from a query, specifically from the function, ExecuteQuery()

Returns

void - returns echo's html code to return rows with an entry number and the timestamp for it

PrintResults is a function used to print the results of a query.

Implements PHPtoSQLInterface.

3.3.2.15 ReformatDate (*$date*)

Parameters

$date	- A date in format "YYYY-MM-DD HH" where HH ranges from 00 to 23.

Returns

string - The date in format "YYYY-MM-DD HH:00" where HH ranges from 00 to 12 AM/PM.

Reformats a date into a more readable format.

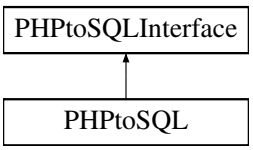
Implements PHPtoSQLInterface.

The documentation for this class was generated from the following file:

- PHP Server Code/PHPtoSQL.php

3.4 PHPtoSQLInterface Interface Reference

Inheritance diagram for PHPtoSQLInterface:



Public Member Functions

- __construct ($db)
- __destruct ()
- getNumCarsThisWeek ($locationChoice)

  Gets the number of cars in the last week starting from today.
- getTrafficByYear ($year, $locationChoice)

  Gives the traffic for each month in an array for the specified year passed in.
- getTrafficByMonth ($year, $month, $locationChoice)

  Gets the traffic for each day during the specified month of the specified year.
- getTrafficByDay ($year, $month, $day, $locationChoice)

  Gets the traffic for each hour and returns it in a 24 element array.
- getTrafficTimeRange ($year1, $month1, $day1, $year2, $month2, $day2, $locationChoice)

  Takes in a date range (start and end date) and counts the number of cars in the given range.
- getListOfLocations ()

  Will return the names of all different locations (parking lots) that are being recorded in the database.
- GetMinimumCountInHour ($countByHour, $dateByHour)

  Finds the minimum count for an hour in a specific time range.
- GetMedianCountInHour ($countByHour, $dateByHour)

  Finds the median count for an hour in a specific time range.
- GetMaximumCountInHour ($countByHour, $dateByHour)

Finds the maximum count for an hour in a specific time range.

- GetAverageCountInHour ($countByHour)

  Finds average number of people counted during an hour in a specific time range.

- ExecuteQuery ($Time1, $Time2, $COMMON)

  A function to execute a query of correct MySQL syntax for the specified instance of the database within a certain time range.

- ConvertTime ($Time)

  Converts a time of a certain format to that of another.

- PrintResults ($results)

  PrintResults is a function used to print the results of a query.

- GroupResultsByHour ($results)

  Groups the results of a query by hour and counts the number of results within each hour of the time range.

- ReformatDate ($date)

  Reformats a date into a more readable format.

3.4.1 Detailed Description

Created by PhpStorm. User: Jabroni Date: 4/22/18 Time: 2:58 PM

3.4.2 Constructor & Destructor Documentation

3.4.2.1 __construct (*$db*)

DBAPI constructor.

Parameters

$db	Mostly sets up the dates in this object. Also sets the timezone to our timezone.

Implemented in PHPtoSQL.

3.4.2.2 __destruct ( )

Class destructor

Implemented in PHPtoSQL.

3.4.3 Member Function Documentation

3.4.3.1 ConvertTime (*$Time*)

Converts a time of a certain format to that of another.

Parameters

$Time	- timestamp of the format 'YYYY-MM-DDTHH:MM'.

Returns

string - which is of the form 'YYYY-MM-DD HH:MM:00'.

Implemented in PHPtoSQL.

3.4.3.2 ExecuteQuery (*$Time1, $Time2, $COMMON*)

A function to execute a query of correct MySQL syntax for the specified instance of the database within a certain time range.

Parameters

$Time1	- First timestamp of the format 'YYYY-MM-DD HH:MM:00'
$Time2	- Second timestamp of the format 'YYYY-MM-DD HH:MM:00'.
$COMMON	- common instance that connects to the database.

Returns

array - all of the resulting data (in rows) returned from the query.

Usage:
$Time1 = $_POST["Time1"];
$Time2 = $_POST["Time2"];
// Convert input times into the correct Timestamp format
$time1Format = ConvertTime($Time1);
$time2Format = ConvertTime($Time2);
// Create a new Common instance to connect to the database
$debug = false;
$COMMON = new Common($debug);
// Execute query and fetch the results
$results = ExecuteQuery($time1Format, $time2Format, $COMMON);

Implemented in PHPtoSQL.

3.4.3.3 GetAverageCountInHour (*$countByHour*)

Finds average number of people counted during an hour in a specific time range.

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.

Returns

integer - Returns the average number of people recorded in an hour of the specific time range.

Implemented in PHPtoSQL.

3.4.3.4 getListOfLocations ( )

Will return the names of all different locations (parking lots) that are being recorded in the database.

Returns

array of strings

Usage:
$listLocations[] = getListOfLocations();

Implemented in PHPtoSQL.

3.4.3.5 GetMaximumCountInHour (*$countByHour, $dateByHour*)

Finds the maximum count for an hour in a specific time range.

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.
$dateByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.

Returns

2 arrays - Returns an array containing (1) the maximum count within an hour and (2) the corresponding time for which the count was recorded.

Implemented in PHPtoSQL.

3.4.3.6 GetMedianCountInHour (*$countByHour, $dateByHour*)

Finds the median count for an hour in a specific time range.

Parameters

$countByHour	- the array of counts by hour returned from GroupResultsByHour.
$dateByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.

Returns

array - Returns an array containing (1) the median count within an hour and (2) the corresponding time for which the count was recorded.

Implemented in PHPtoSQL.

3.4.3.7 GetMinimumCountInHour (*$countByHour, $dateByHour*)

Finds the minimum count for an hour in a specific time range.

Parameters

$countByHour	- the corresponding array of dates sorted by hour returned from GroupResultsByHour.
$dateByHour	- the array of counts by hour returned from GroupResultsByHour.

Returns

Returns an array containing (1) the minimum count within an hour and (2) the corresponding time for which the count was recorded.

Implemented in PHPtoSQL.

3.4.3.8 getNumCarsThisWeek (*$locationChoice*)

Gets the number of cars in the last week starting from today.

Returns

int

Usage:
$numCars = getNumCarsThisWeek("lot35"); for Lot 35

Implemented in PHPtoSQL.

3.4.3.9 getTrafficByDay (*$year, $month, $day, $locationChoice*)

Gets the traffic for each hour and returns it in a 24 element array.

Parameters

$year	
$month	
$day	
$locationChoice	

Returns

int array - length 24 array with the traffic number for each our of the day

Usage:
< $dayArray = getTrafficByDay(2018, 2, 15, "lot35");> for Febraury 15, 2018 for Lot 35

Implemented in PHPtoSQL.

3.4.3.10 getTrafficByMonth (*$year, $month, $locationChoice*)

Gets the traffic for each day during the specified month of the specified year.

Parameters

$year	
$month	
$locationChoice	

Returns

int array - array with the traffic number for each day of the specified month

Usage:
$monthArray = getTrafficByMonth(2018, 2, "lot35"); // for February 2018 for Lot 35

Implemented in PHPtoSQL.

3.4.3.11 getTrafficByYear (*$year, $locationChoice*)

Gives the traffic for each month in an array for the specified year passed in.

Parameters

$year	
$locationChoice	- string for the lot you want to put in

Returns

int array - a length 12 array where each element is the number of cars that passed through for that month. If there were 8 cars that passed through in April, the element at index 3 would be 8

Usage:
$yearArray = getTrafficByYear(2018, "lot35"); for 2018 for Lot 35

Implemented in PHPtoSQL.

3.4.3.12 getTrafficTimeRange (*$year1, $month1, $day1, $year2, $month2, $day2, $locationChoice*)

Takes in a date range (start and end date) and counts the number of cars in the given range.

Parameters

$year1	
$month1	
$day1	
$year2	
$month2	
$day2	
$locationChoice	

Returns

integer - the number of cars in the time range

Usage:
$trafficInRange = getTrafficTimeRange(2018, 2, 15, 2018, 2, 28, "lot35");
for number of cars between 2/15/2018 and 2/28/2018 for Lot 35

Implemented in PHPtoSQL.

3.4.3.13 GroupResultsByHour (*$results*)

Groups the results of a query by hour and counts the number of results within each hour of the time range.

Parameters

$results	- the results returned from a query, specifically from the function, ExecuteQuery().

Returns

array - Returns an array containing (1) an array of the dates within the results, grouped by hour, and (2) an array containing the counts for each corresponding hour of the dates array.

Implemented in PHPtoSQL.

3.4.3.14 PrintResults (*$results*)

PrintResults is a function used to print the results of a query.

Parameters

$results	- the results returned from a query, specifically from the function, ExecuteQuery()

Returns

void - returns echo's html code to return rows with an entry number and the timestamp for it

Implemented in PHPtoSQL.

3.4.3.15 ReformatDate (*$date*)

Reformats a date into a more readable format.

Parameters

$date	- A date in format "YYYY-MM-DD HH" where HH ranges from 00 to 23.

Returns

string - The date in format "YYYY-MM-DD HH:00" where HH ranges from 00 to 12 AM/PM.

Implemented in PHPtoSQL.

The documentation for this interface was generated from the following file:

- PHP Server Code/PHPtoSQLInterface.php

Index