<?php require\_once('config.php'); ?>

<!DOCTYPE html>

<html>

<body>

<h1>Database Tester (PDO)</h1> <?php

try {

$pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS); //创建一个新的PDO对象

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION); //设置属性

$sql = "select \* from Artists order by LastName";//从artist表中选择并按Lastname排列

$result = $pdo->query($sql);//执行命令得到result

while ($row = $result->fetch()) {

echo $row['ArtistID'] . " - " . $row['LastName'] . "<br/>";

}//遍历

$pdo = null;

}catch (PDOException $e) {

die( $e->getMessage() );

}//抛出错误

?>

</body>

</html>

<?php require\_once('config.php'); ?>

<!DOCTYPE html>

<html>

<body>

<h1>Database Tester (mysqli)</h1> Genre:

<select>

<?php

$connection = mysqli\_connect(DBHOST, DBUSER, DBPASS, DBNAME); //连接数据库

if ( mysqli\_connect\_errno() ) {

die( mysqli\_connect\_error() );

}//如果连接不成功，即报错

$sql = "select \* from Genres order by GenreName";

if ($result = mysqli\_query($connection, $sql)) {

// loop through the data

while($row = mysqli\_fetch\_assoc($result)) {

echo '<option value="' . $row['GenreID'] . '">';

echo $row['GenreName'];

echo "</option>";

}//获取该行信息，加入Option选项

// release the memory used by the result set

mysqli\_free\_result($result);

}

// close the database connection

mysqli\_close($connection);

?>

</select>

</body>

</html>

Exercise 8

function outputPaintings() {//右端照片

try {

if (isset($\_GET['id']) && $\_GET['id'] > 0) {//选取符合条件的id

$pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS); //创建新的PDO对象

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = 'select \* from Paintings where ArtistId=' . $\_GET['id']; //从painting中寻找artistid

$result = $pdo->query($sql);

while ($row = $result->fetch()) {//遍历图片

outputSinglePainting($row);

}

$pdo = null;

}

}catch (PDOException $e) {

die( $e->getMessage() );//报错提醒

}

}

function outputSinglePainting($row) {

echo '<div class="item">';

echo '<div class="image">';

echo '<img src="images/art/works/square-medium/' .$row['ImageFileName'] .'.jpg">'; //添加照片

echo '</div>';

echo '<div class="content">';

echo '<h4 class="header">';

echo $row['Title'];

echo '</h4>';

echo '<p class="description">';

echo $row['Excerpt'];

echo '</p>';

echo '</div>'; // end class=content

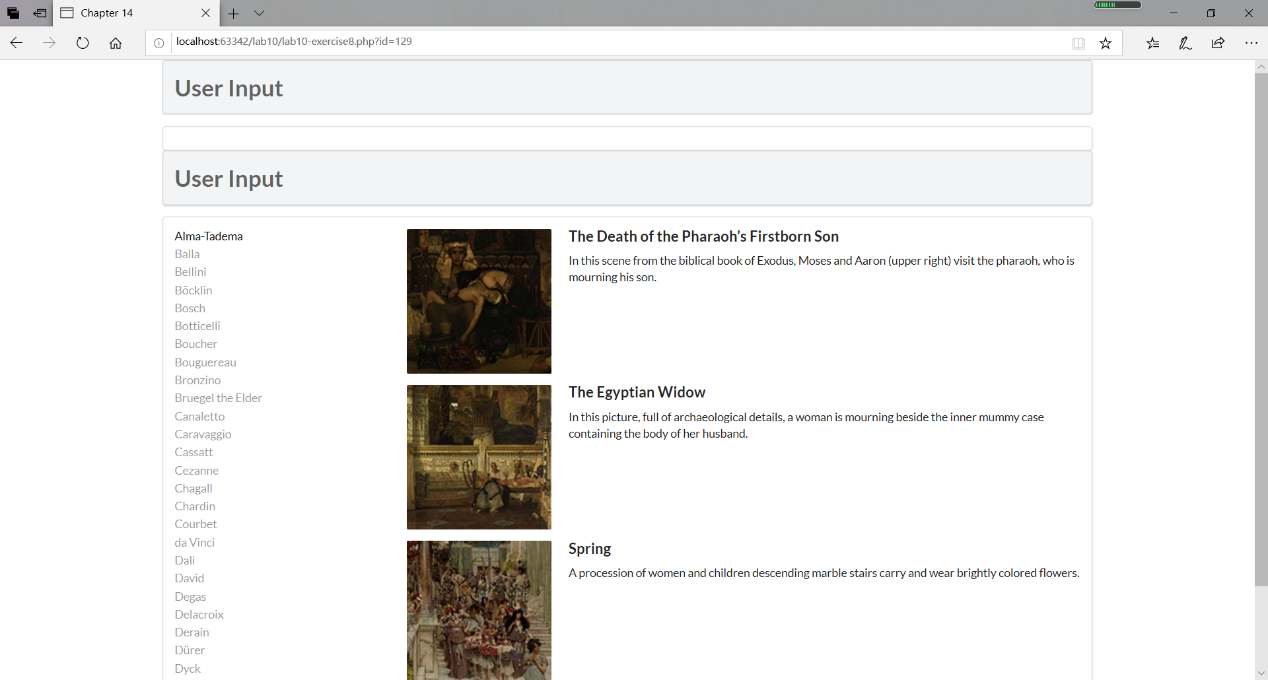
echo '</div>'; // end class=item

}

这段github上没有……从phpstorm原文件复制下来去了样式，可读性有点差

function outputArtists() {  
 try {  
 $pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS);  
 $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  
 $sql = "select \* from Artists order by LastName limit 0,30";  
 $result = $pdo->query($sql);  
 while ($row = $result->fetch()) {  
 echo '<a href="' . $\_SERVER["SCRIPT\_NAME"] . '?id=' . $row['ArtistID'] . '" class="';  
 if (isset($\_GET['id']) && $\_GET['id'] == $row['ArtistID']) echo 'active ';  
 echo 'item">';  
 echo $row['LastName'] . '</a>';  
 }  
 $pdo = null;  
 }  
 catch (PDOException $e) {  
 die( $e->getMessage() );  
 }  
}

输出页面左端艺术家的内容，选取30个，循环内返回artistid,，并加入active，最后是报错信息。



1、ResultSet  executeQuery（String sql）throws SQLException：专用于查询。

2、int  executeUpdate（String sql）throws SQLException：执行DDL、DML语句，前者返回0，后者返回受影响行数。

3、boolean execute（String sql）throws SQLException：可执行任何SQL 语句。如果执行后第一个结果为ResultSet（即执行了查询语句），则返回true；如果执行了DDL、DML语句，则返回false。返回结果为true，则随后可通过该Statement对象的getResultSet()方法获取结果集对象（ResultSet类型），返回结果为false，则可通过Statement对象的getUpdateCount（）方法获得受影响的行数。

——>PrepareStatement：为Statement的子接口，可预编译SQL 语句，常用语执行多条结构相同，仅值不同的SQL 语句，见下例。同样具有Statement对象常用的三个方法，但用法不同，因为已经预编译了SQL 月，所以无需再在方法中写sql语句，只需setString方法设置参数值即可，如代码中。

使用PreparedStatement比使用Statement多了三个好处：

1、PreparedStatement预编译SQL语句，性能更好，执行更快。

2、PreparedStatement无须“拼接”SQL 语句，编程更简单。

3、PreparedStatement可以防止SQL 注入（如将输入的true当成直接量，导致判断直接通过，从而降低了安全性），安全性更好。