AGH INFORMATYKA Akademia Górniczo-Hutnicza w Krakowie

PHP web pages

Jacek Dajda [dajda@agh.edu.pl]
Room: 3.11

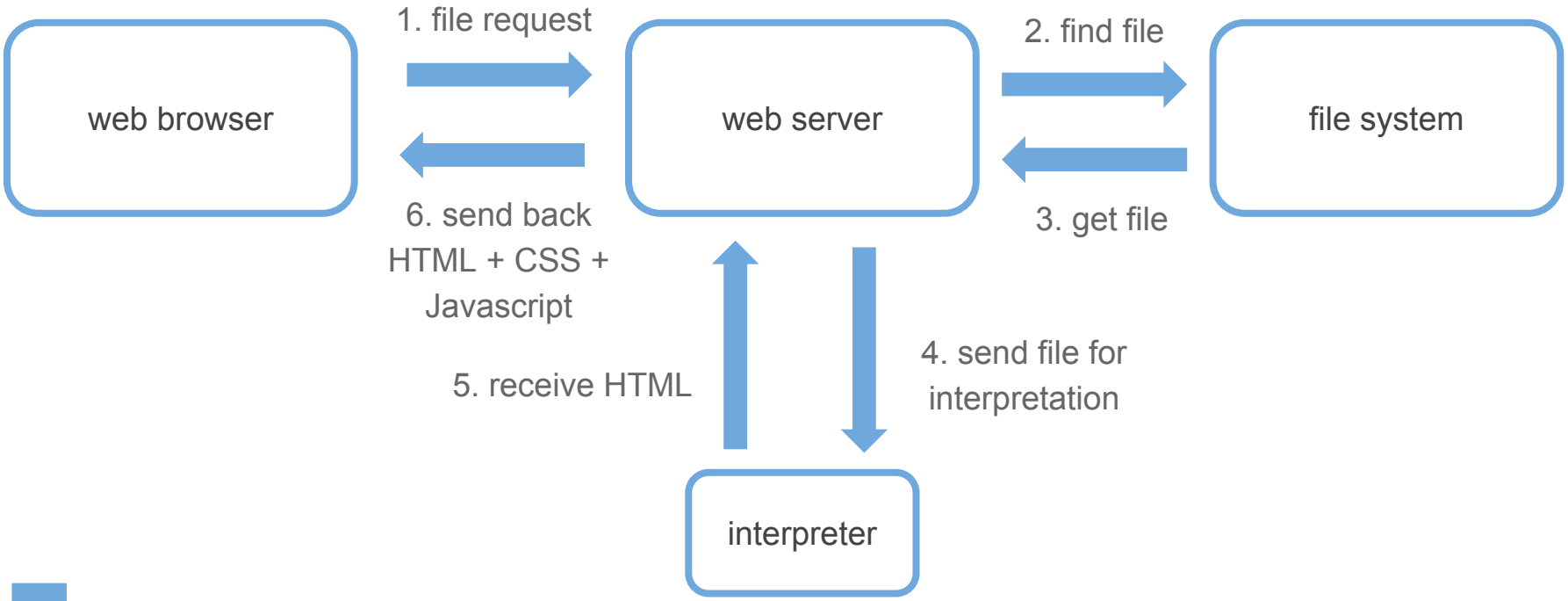
Agenda

- Embedding into HTML - reminder
- Sending data to webserver
- Implementing forms
- Handling files
- Working with databases

Embedding into HTML

Reminder

Web server programming



‘Hello world’ index.php file

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<?php
$name = "World";
echo "<h1>Hello, $name!</h1>";
?>
</body>
</html>
```

Output

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Running PHP program in web server

- You can install Apache with PHP and put your index.php file in home directory
- Since PHP Ver. 7.0 you can use built-in web server

file: test.php

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<?php
$name = "World";
echo "<h1>Hello, $name!</h1>";
?>
</body>
</html>
```

```
php -S localhost:8000 test.php
```

Include

- One of the most useful tools is to insert another php script from a file into the current php script.
- The command `include('filename');`
- This will import contents of a text file called filename and insert it at the include spot.
- The included text may be composed of XHTML, PHP or both. Any PHP in the included text must be inside the <?php tags

Include (index.php)

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<?php
include('header.html');
$name = "World";
echo "<h1>Hello, $name!</h1>";
include('footer.html');
?>
</body>
</html>
```

Include (header.html)

```
<header>
    <h2>Header</h2>
</header>
```

Include (footer.html)

```
<footer>
    <p>Footer</p>
</footer>
```

HTML Output

```
Header

Hello, World!

Footer
```

Nicer header and footer

```
<html>
<head>
<title>Hello World</title>
<style>

/* Style the header */
header {
    background-color: #666;
    padding: 30px;
    text-align: center;
    font-size: 35px;
    color: white;
}
```

```
/* Style the footer */
footer {
    background-color: #777;
    padding: 10px;
    text-align: center;
    color: white;
}

</style>
</head>
...
```

HTML Output

Header

Hello, World!

Footer

Include - other similar functions

- `require()` - works almost exactly like the include() function except that if the file inclusion attempt fails, require() will throw an error and stop further execution of your scripts whereas include() will simply throw a warning before continuing on as if nothing happened.
- Sometimes, depending on the code in them, including the same file multiple times can cause some issues. To prevent it, use these functions:
  - `require_once()`
  - `include_once()`

Sending data to webserver

HTTP protocol

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
- Works as request-response protocol
- Most popular/important HTTP Methods:
  - GET
  - POST

GET and POST

GET

- sending data through URL
- used for filtering data

`http://server/path/scriptfile?key1=value1&key2value2`

POST

- sending data using body of the HTTP request
- used for adding data to the website e.g. sending forms

`http://server/path/scriptfile`

GET and POST

- PHP provides $_GET and $_POST associative arrays to access all sent information using GET and POST methods, e.g.

http://server/path/scriptfile.php?key1=value1&key2value2

```
echo print_r($_GET); // Array ( [key1] => value1 [key2] => value2 )
echo $_GET['key1'];  // value1
```

$_GET in “Hello World” example

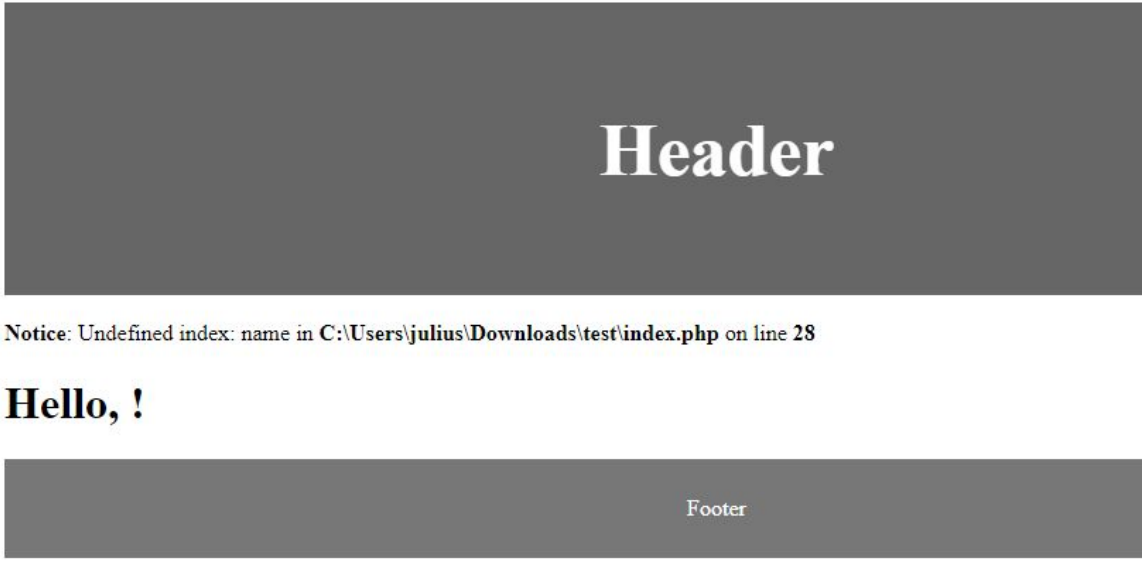
```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<?php
include('header.html');
$name = $_GET['name'];
echo "<h1>Hello, $name!</h1>";
include('footer.html');
?>
</body>
</html>
```

$_GET in “Hello World” example

- Run server: `php -S localhost:8000`
- Get the script with parameter:
  `http://localhost:8000/index.php?name=John`
- Result:

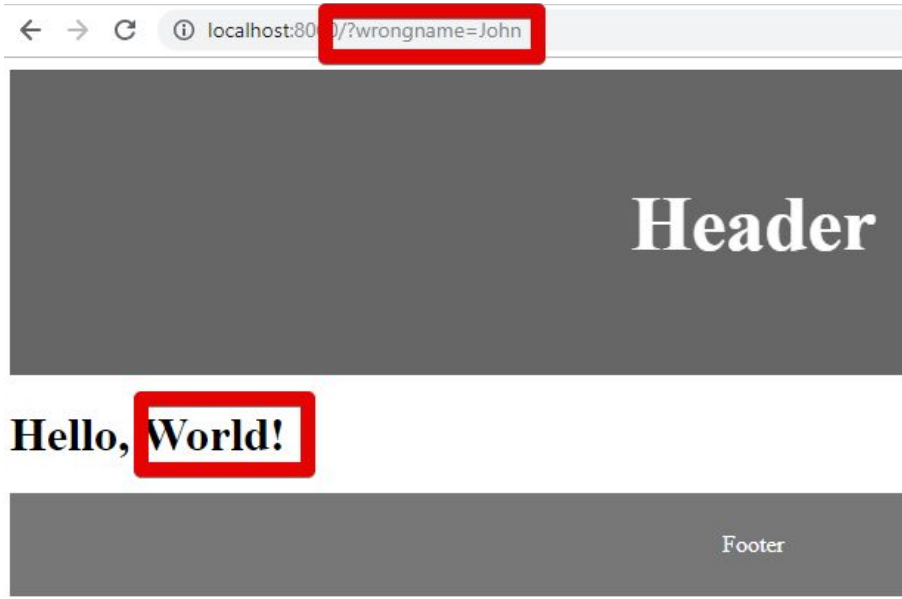
Handling not existing key

- Get the script with parameter:
  `http://localhost:8000/index.php?wrongname=John`
- Result:



Handling empty keys

```
$name = "World";
if (isset($_GET['name'])) {
    $name = $_GET['name'];
}
echo "<h1>Hello, $name!</h1>";
```



Implementing forms

Simplest form - using GET

```
<form action="index.php" method="get">
Name: <input type="text" name="name">
<input type="submit">
</form>
```

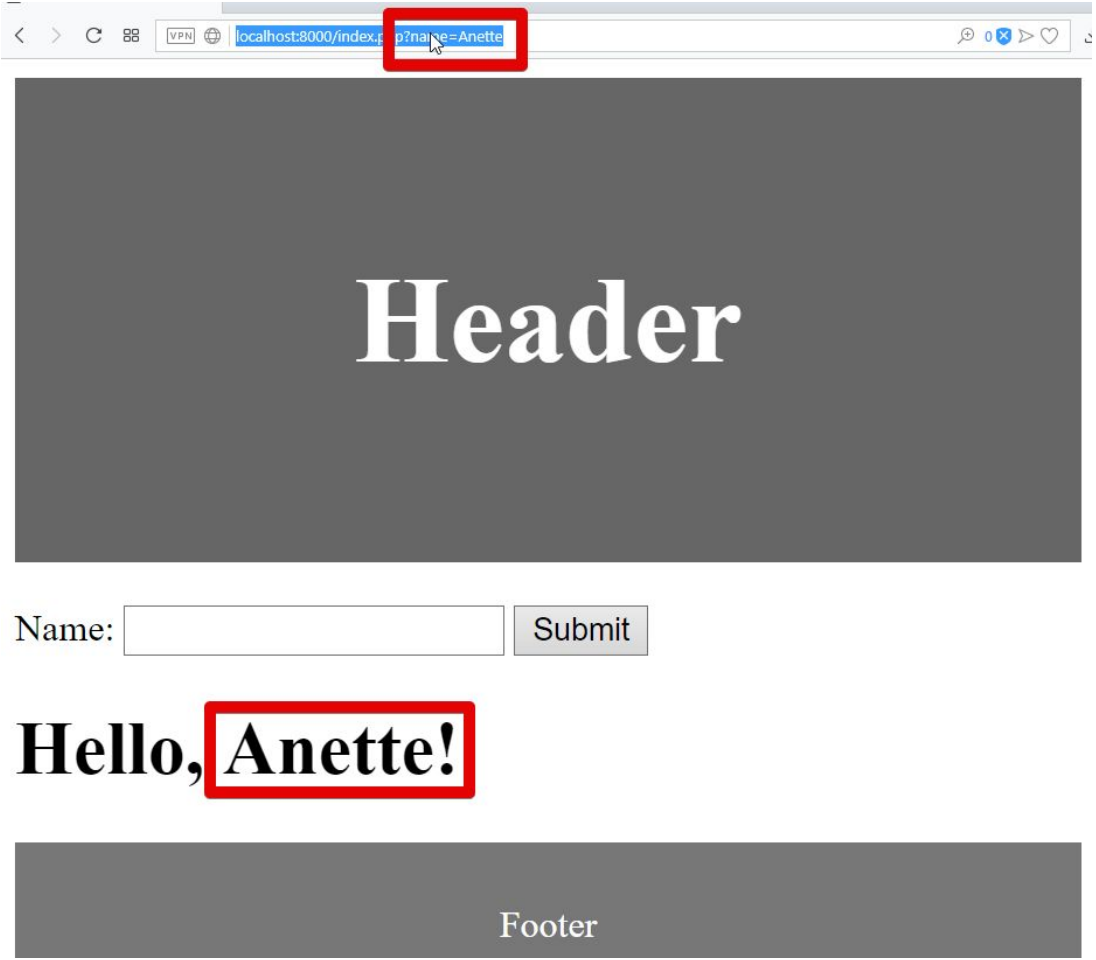
Name: Submit

Simplest form - using GET

file: index.php

```
<body>
<?php
include('header.html');
?>
<br>
<form action="index.php" method="get">
Name: <input type="text" name="name">
<input type="submit">
</form>
```

```
<?php
$name = "World";
if (isset($_GET['name'])) {
    $name = $_GET['name'];
}
echo "<h1>Hello, $name!</h1>";
include('footer.html');
?>
</body>
```


localhost:8000/index.php?name=Anette
Header
Name:
Submit
Hello, Anette!
Footer

Simplest form - using POST

```
<form action="index.php" method="post">
Name: <input type="text" name="name" />
<input type="submit" />
</form>
<?php
$name = "World";
if (isset($_POST['name'])) {
    $name = $_POST['name'];
}
echo "<h1>Hello, $name!</h1>";
```

Hello World

localhost:8000/index.php

Header

Name: Submit

Hello, Anette!

Footer

▼ General

Request URL: http://localhost:8000/index.php

Request Method: POST

Status Code: 200 OK

Remote Address: [::1]:8000

Referrer Policy: no-referrer-when-downgrade

▶ Response Headers (5)

▶ Request Headers (12)

▼ Form Data view source view URL encoded

name: Anette

Handling files

Function `fopen()`

6 modes of operation:

- r - opens the file for reading only. Places the file pointer at the beginning of the file.
- r+ - opens the file for reading and writing. Places the file pointer at the beginning of the file.
- w - opens the file for writing only. Places the file pointer at the beginning of the file and truncates the file to zero length. If files does not exist then it attempts to create a file.
- w+ - opens the file for reading and writing only. Places the file pointer at the beginning of the file and truncates the file to zero length. If files does not exist then it attempts to create a file.

Function `fopen()`

6 modes:

- a - opens the file for writing only. Places the file pointer at the end of the file. If files does not exist then it attempts to create a file.
- a+ - opens the file for reading and writing only. Places the file pointer at the end of the file. If files does not exist then it attempts to create a file.

File - basic functions

- Checking if end of file

```
feof($file)
```

- Reading one line from file

```
fgets($file)
```

File - basic functions

- Reading whole file contents

```
file_get_contents($filename)
```

- Write string to file

```
fwrite($file, $string_to_write)
```

- Deleting a file

```
unlink($file)
```

Reading file - example

```
// Output one line until end-of-file
while(!feof($file)) {
    echo fgets($myfile) . "<br>";
}
```

fopen() – reading example

```
$filename = "tmp.txt";
$file = fopen( $filename, "r" );

if( $file == false ) {
    echo ( "Error in opening file" );
    exit();
}
echo file_get_contents($file);
fclose( $file );
```

fopen() – writing example

```
$filename = "newfile.txt";
$file = fopen( $filename, "w" );

if( $file == false ) {
    echo ( "Error in opening new file" );
    exit();
}
fwrite( $file, "This is  a simple test\n" );
fclose( $file );
```

More advanced example

- We want to register names given in the above form in a single file
- Each name is saved in new line
- We want to display all names
- The new name should be in bold
- When you refresh your app the names are still there

Header

Name: Submit

Hello:

Footer

More advanced example

```
<form action="index.php" method="post">
Name: <input type="text" name="name">
<input type="submit">
</form>
<?php
echo "<h1>Hello: </h1>";
if (isset($_POST['name'])) {
    $name = $_POST['name'];
}
```

```
$filename = 'newfile.txt';
echo file_get_contents($filename);

$file = fopen( $filename, "a+" );
if( $file != false ) {
    echo "<b>$name</b><br>";
    fwrite( $file, "$name\n" );
    fclose( $file );
}
```

Working with DB

Relational databases

- Relational databases are one of the most popular DB engines for developing websites
- SQL (Structured Query Language) - standard language for accessing and manipulating databases.
- Popular relational database engines: SQLite, MySQL (MariaDB), PostgreSQL, MS SQL

SQL Quiz!

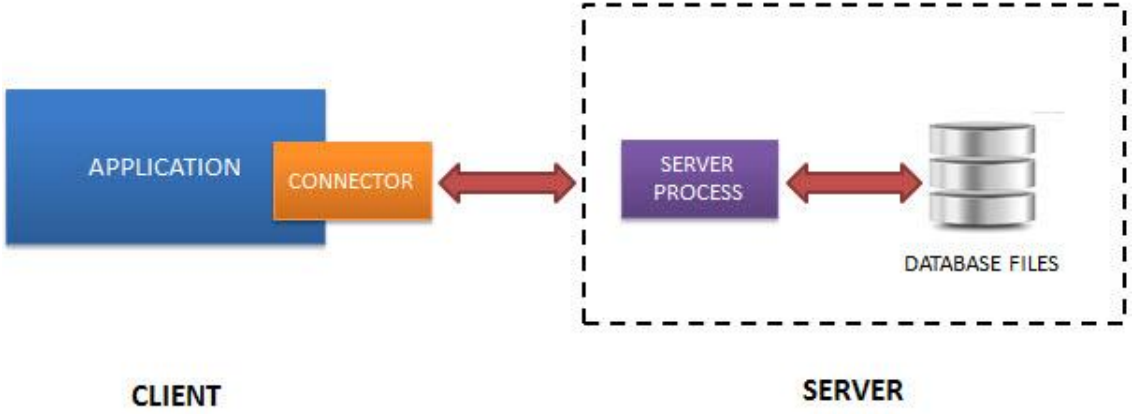
What is the result of the following commands?

```
select name, surname from customers;
select * from customers;
select * from customers where id = 1;
```

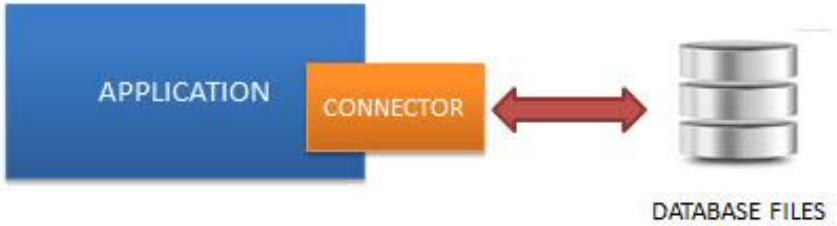
SQLite

- SQLite is a software library that provides a relational database management system.
- File-based, no server required
- Perfect for small applications
- Used by Android
- https://www.sqlite.org/index.html

What is file-based database engine



Working with standard database server



Working with file-based

Source: https://www.tutorialspoint.com/sqlite/

Example SQL table

```sql
CREATE TABLE `customer` (
    `id` INTEGER PRIMARY KEY AUTOINCREMENT,
    `name` TEXT,
    `surname` TEXT,
    `email` TEXT
)
```

Connecting to SQLite - example

```
class MyDB extends SQLite3 {
    function __construct() {
        $this->open('customers.db');
    }
}
$db = new MyDB();
if(!$db) {
    echo $db->lastErrorMsg();
    exit();
}
```

Quering SQLite

```
$sql = "select name, surname from customer where id = 3";
 $ret = $db->query($sql);
   while($row = $ret->fetchArray(SQLITE3_ASSOC) ) {
      echo "id = ". $row['id'] . ", ";
      echo "name = ". $row['name'] . ", ";
      echo "surname = ". $row['surname'] ."<br>";
   }
$db->close();
```

SQLite example: customers DB

Header

Name: Joh

Surname:

Email:

Submit

Customers

Footer

SQLite example: customers form

```
<form action="index.php" method="post">
Name: <input type="text" name="name"><br>
Surname: <input type="text" name="surname"><br>
Email: <input type="text" name="email"><br>
<input type="submit">
</form>
```

SQLite example: adding customers

```
if (isset($_POST['name']) && isset($_POST['surname']) &&
isset($_POST['email'])) {
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $email = $_POST['email'];
    $sql = "insert into 'customer' (name, surname, email)
            values ('$name', '$surname', '$email')";
    $ret = $db->query($sql);
}
```

SQLite example: listing customers

```
echo "<h2>Customers</h2>";
$sql = "select * from customer";
$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ) {
    echo "<b>id</b> = ". $row['id'] . ", ";
    echo "<b>name</b> = ". $row['name'] . ", ";
    echo "<b>surname</b> = ". $row['surname'] .", ";
    echo "<b>email</b> = ". $row['email'] . "<br>";
}
```

Active record - concept

- Active record is an approach to access data in a database
- A database table or view is wrapped into a class
- Every object is tied to a single row
- When an object is updated the row is also updated
- Table name and columns names are generated based on names of class and its attributes
- Example library: https://github.com/jpfuentes2/php-activerecord

Active record - concept

```
CREATE TABLE `customers`(
    `id` INTEGER PRIMARY
        KEY AUTOINCREMENT,
    `name` TEXT,
    `surname` TEXT,
    `email` TEXT
)
```

```
class Customer extends
ActivityRecord\Model {
    $id;
    $name;
    $surname;
    $email;
}
```

Active record - adding new object

```
$customer = new Customer();
$customer->name = "John";
$customer->surname = "Wick";
$customer->email = "some@email.com";
$customer->save(); // INSERT INTO `customers` (name, surname, email)
                      VALUES('John', 'Wick', 'some@email.com')
```

Active record - getting objects

```
$customer = Customer::find(1);
// or $customer = Customer::first();
echo $customer->name; // 'Johny'
echo $customer->surname; // 'English'

$customer = Customer::find_by_name('Johny');
$customer = Customer::find_by_name_and_surname('Johny','English');
$customer = Customer::find_by_name_or_email('Johny',
'johny@enlish.com');
```

Active record - updating objects

```
$customer = Customer::find(1);
echo $customer->name; // 'John'
$customer->surname = 'English';
$customer->save();
# UPDATE `customers` SET surname='English' WHERE id=1

$customer->name = 'Johny';
$customer->save();
# UPDATE `customers` SET name='Johny' WHERE id=1
```

Active record - deleting objects

```
$customer = Customer::find(1);
$customer->delete();
# DELETE FROM `customers` WHERE id=1
```