PHP Training

Episode 4

by Andrei Kojusko, 2018

Lesson contents

- Composer
- Webserver
- $_GET, $_POST, $_COOKIE, $_SERVER, headers
- $_SESSION
- HTML Forms
- Templates
- MVC, DAO, ORM, DTO
- SOLID, DRY, KISS
- REST

Composer

```
// composer.json

{
    "name": "akojusko/webapp",
    "license": "mit",
    "type": "project",
    "description": "Homework web app",
    "autoload": {
        "psr-4": { "Webapp\\": "src/" }
    },
    "require": {
        "php": ">=7.2"
    }
}
```

```
>composer install
```

Creates:

- vendors
- autoload
- composer.lock

>composer require ***vendor/bundle***

Bundles:

- Packagist
- Custom repositories

>composer update [***vendor/bundle***]

```
>composer update --no-dev
```

Webserver

- Apache
- Nginx / phpfpm
- Built-in

```
>php -S localhost:8000
```

Hosts:
Windows => Windows\System32\drivers\etc\hosts
Linux => /etc/hosts

```
127.0.0.1    localhost
```

Links

https://getcomposer.org/

https://packagist.org/

http://php.net/manual/en/features.commandline.webserver.php

$_GET, $_POST, $_FILES

http://localhost/index.php?name=Andrei

$_GET['name'] === 'Andrei'

http://localhost/index.php?id=1&name=Andrei

$_GET['id'] === '1'
$_GET['name'] === 'Andrei'

http://localhost/index.php?ids[]=1&ids[]=2&ids[]=3

$_GET['ids'] === ['1', '2', '3']

$_POST:

Content-type:

- application/x-www-form-urlencoded
- multipart/form-data

$_FILES:

```
[name] => facepalm.jpg
[type] => image/jpeg
[tmp_name] => /tmp/phpn3FmFr
[error] => 0
[size] => 15476
```

$_COOKIE, $_SERVER, headers

```
bool setcookie (
        string $name,
        string $value = '',
        int $expire = 0,
        ...
)
```

- Part of headers
- Not immediately available

```
setcookie('TestCookie', $value, time()+3600);

$_COOKIE['TestCookie']
```

```
header('Location: http://www.example.com/');

header('HTTP/1.0 404 Not Found');
```

Important $_SERVER values:

- REMOTE_ADDR
- REQUEST_URI
- REQUEST_METHOD
- PATH_INFO

$_SESSION

Start session:
session_start()

Set session variable:
$_SESSION['varname'] = 55;

Get session variable:
echo $_SESSION['varname'];

Clear the session:
unset($_SESSION['varname']);
session_unset(); // clears all variables

- Uses a cookie
- Has a limited lifetime
- Locks session file for writing

Links

http://php.net/manual/en/features.file-upload.php

http://php.net/manual/en/reserved.variables.php

http://php.net/manual/en/function.setcookie.php

http://php.net/manual/en/function.header.php

http://php.net/manual/en/ref.session.php

HTML Forms

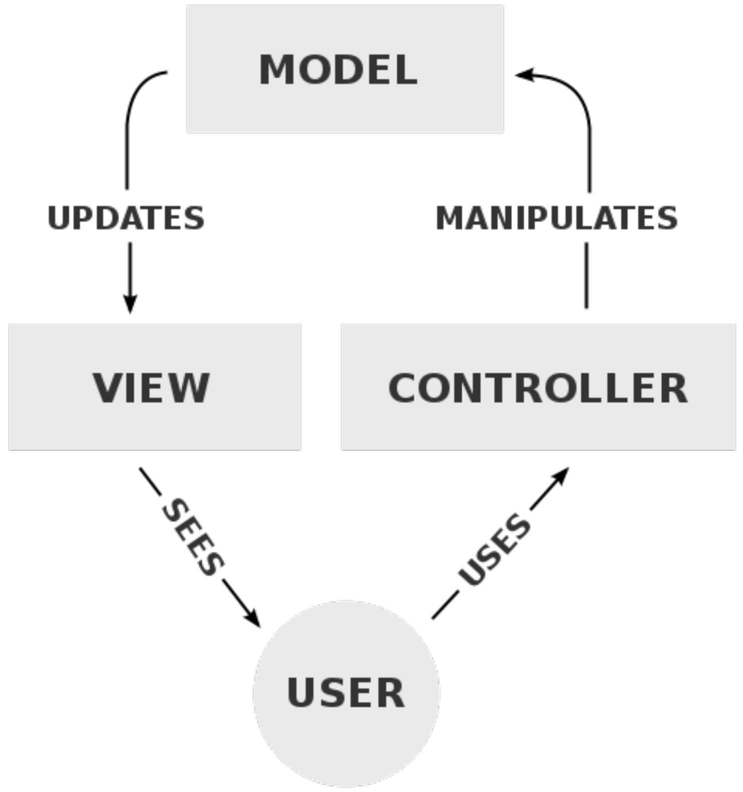
```
<form action="foo.php" method="post" enctype="multipart/form-data">
    Name:  <input type="text" name="username" /><br />
    Email: <input type="text" name="email" /><br />
    <input type="submit" name="submit" value="Submit me!" />
</form>
```

Templates

```
<div>

    <h1>Welcome!</h1>

    <div>Hello {{ name }}!</div>

    <div>Today is {{ date }}</div>

</div>
```

- prepare values
- load template
- replace the placeholder
- return / render

MVC, DAO, ORM, DTO



DAO: Data Access Object
Application <-> DAO <-> Database

ORM: Object-relational mapping
Object <-> ORM <-> Database

DTO: Data transfer object
An object that carries data between processes (ex. webservice calls)

SOLID, DRY, KISS

Single responsibility principle
Open/closed principle
Liskov substitution principle
Interface segregation principle
Dependency inversion principle

Don't **R**epeat **Y**ourself

Keep **I**t **S**imple, **S**tupid

Links

https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

https://en.wikipedia.org/wiki/Data_access_object

https://en.wikipedia.org/wiki/Object-relational_mapping

https://en.wikipedia.org/wiki/SOLID_(object-oriented_design)

https://en.wikipedia.org/wiki/Don%27t_repeat_yourself

https://en.wikipedia.org/wiki/KISS_principle

Homework

username/web/index.php
- use templates
- use MVC and services

Routes:
/ or empty:
- if user is not logged in, then show login form
- if user is authenticated, then show welcome message with user name

/login: action of the login form, redirects to /
- if login failed then a failure message should be displayed on the login form
- use sqlite DB with predefined user(s)

- Use OOP
- Use separate files
- Use composer with autoloading
- Use Symfony 2 Code Style
- Use correct file/folder names