Pentalog

PHP Training

Episode 5

by Andrei Kojusko, 2018

Lesson contents

- REST
- cURL
- Escaping HTML and URLs
- Hashing
- Crypting
- More on Closures
- Singletons
- Dependency injection
- Service container

REST

Representational **S**tate **T**ransfer

- Client-server architecture
- Statelessness
- Cacheability
- Layered system
- Uniform interface

Uses:

- HTTP methods
- HTTP response codes
- JSON

Examples:

GET /articles?author=akojusko
HTTP 200 OK
Array with all articles where author is “akojusko”

GET /articles/15
HTTP 200 OK
Article with ID 15

POST /articles
HTTP 201 Created
New article is saved to database

Links

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

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods

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

https://developer.github.com/v3/guides/getting-started/

cURL

```
$options = [
    CURLOPT_POST => 1,
    CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_POSTFIELDS => http_build_query($postData)
];

$ch = curl_init();
curl_setopt_array($ch, $options);
$result = curl_exec($ch);
if ($result === false) {
    throw new \Exception(curl_error($ch));
}

curl_close($ch);
```

Links

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

http://php.net/manual/en/function.curl-setopt.php

Escaping HTML and URLs

```
echo nl2br("foo isn't\n bar");

// foo isn't<br /> bar
```

```
echo htmlentities("A 'quote' is <b>bold</b>");

// A 'quote' is &lt;b&gt;bold&lt;/b&gt;
```

```
echo '<a href="/url?foo='.urlencode($userinput).'">';
```

```
$query_string =
        'foo='.urlencode($foo).
        '&bar='.urlencode($bar);

echo '<a href="/url?'.htmlentities($query_string).'">';
```

Links

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

http://php.net/manual/en/function.nl2br.php

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

Hashing

```
echo hash('ripemd160', 'The quick brown fox jumped over the lazy dog.');

// ec457d0a974c48d5685a7efa03d137dc8bbde7e3
```

```
$str = 'apple';

if (md5($str) === '1f3870be274f6c49b3e31a0c6728957f') {

    echo "Would you like a green or red apple?";

}
```

Crypting

```php
$options = [
    'cost' => 12,
];

echo password_hash(
        "rasmuslerdorf",
        PASSWORD_BCRYPT,
        $options
);
```

```
/*
$2y$11$q5MkhSBtlsJcNEVsYh64a.aCluzHnGog7TQAKVm
QwO9C8xb.t89F.
*/
```

```php
if (password_verify($plain, $hash)) {

    echo 'Password is valid!';

} else {

    echo 'Invalid password.';

}
```

- PASSWORD_BCRYPT
- PASSWORD_ARGON2I

Links

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

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

http://php.net/manual/en/function.password-hash.php

http://php.net/manual/en/function.password-verify.php

More on Closures

```
$userId = 15;

array_filter(function ($arr) use ($userId) {
  return $var->getUserId() === $userId;
});
```

```
$this->userId = 15;

array_filter(function ($arr) {
  return $var->getUserId() === $this->userId;
});
```

```
$tax = 0.25;
$total = 0.00;

$callback =
  function ($quantity, $product) use ($tax, &$total) {
    $pricePerItem = 5;
    $total += ($pricePerItem * $quantity)
            * ($tax + 1.0);
  };

array_walk($this->products, $callback);
```

Singletons

```
class Singleton {
  private static $instance = null;

  private function __construct() {
    // Initialisation code
  }

  private function __clone() {}

  public static function getInstance(): self
  {
    if (self::$instance === null) {
      self::$instance = new self();
    }

    return self::$instance;
  }
}
```

Dependency injection

```
interface GeolocationService {
    public function
      getCoordinatesFromAddress($addr);
}

class GoogleMaps implements GeolocationService
{
      //…
}

class OpenStreetMap implements GeolocationService
{
      //…
}
```

```
class StoreService {
    private $geolocationService;

    public function __construct(
      GeolocationService $geolocationService
    ) {
        $this->geolocationService =
            $geolocationService;
    }

    public function getAddrCoordinates($addr) {
        return
            $this
            ->geolocationService
            ->getCoordinatesFromAddress($addr);
    }
}
```

Service container

```
class Kernel {
    private $container = [];

    private function add(string $service)  {
        // ... instantiate the service
        $this->container[$service] = $instance;
        return $instance;
    }
    public function get(string $service)  {
        if (!array_key_exists($service, $this->container)) {
            $this->add($service);
        }
        return $this->container[$service];
    }
```

Service container - instantiation

```php
$class = new \ReflectionClass($service);
$constructor = $class->getConstructor();

if ($constructor !== null) {
    $arguments = [];
    foreach ($constructor->getParameters() as $parameter) {
        $paramType = $parameter->getClass();
        $arguments[] = $this->get($paramType->getName());
    }
    $instance = new $service(...$arguments);
} else {
    $instance = new $service;
}
```

Homework

1. Implement crypted passwords in Webapp.
2. After login, check if user has a GitHub ID with the same login. If yes, then display a link to his profile.
   - Use cURL & GitHub API
3. Refactor the code: implement service container with dependency injection.