

Middleware in PHP

Võ Duy Tuấn

CEO/Founder
Teamcrop.com





Vo Duy Tuan

CEO / Founder @ Teamcrop.com

Interests

- PHP
- Web Scalability & Optimization
- Data Mining
- E-Commerce
- Docker & Microservices

Table of contents

- What is PHP-FIG?
- What is PSR?
- Micro Framework
- Middleware

1. What is PHP-FIG?

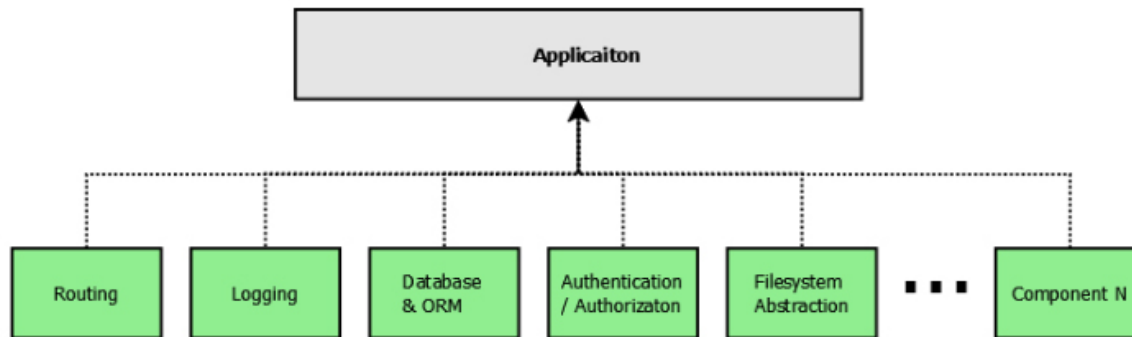
- PHP Framework Interop Group
 - A group of PHP projects whose goal is to talk about commonalities between projects and find ways can work better together.
- Members:
 - CakePHP, Composer, Doctrine, Drupal, Joomla, Laravel, Phalcon, phpBB, phpDocumentor, PrestaShop, SugarCRM, Symfony2, Yii, Zend Framework 2...

2. What is PSR?

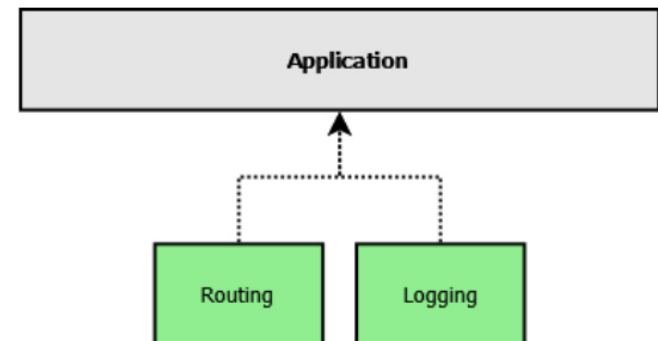
- PSR = PHP Standard Recommendation
- Accepted PSRs:
 - PSR-1: Basic Coding Standard
 - PSR-2: Coding Style Guide
 - PSR-3: Logger Interface
 - PSR-4: Autoloading Standard
 - PSR-6: Caching Interface
 - PSR-7: HTTP Message Interface

3. Micro Framework

Sample Architecture of a Full-Stack Framework



Sample Architecture of a MicroFramework



3.1. Micro Framework Feature

- Small Code-base
- HTTP Request
- HTTP Response
- Routing
- Error Handling
- Middleware

3.2. Micro Framework Usage

- API Restful Web Service
- Small Project
- Microservice architecture

3.3. Micro Framework Names

- Slim
- Silex
- Lumen
- Flight
- Zend Expressive
- ...

3.4. An Example in Slim

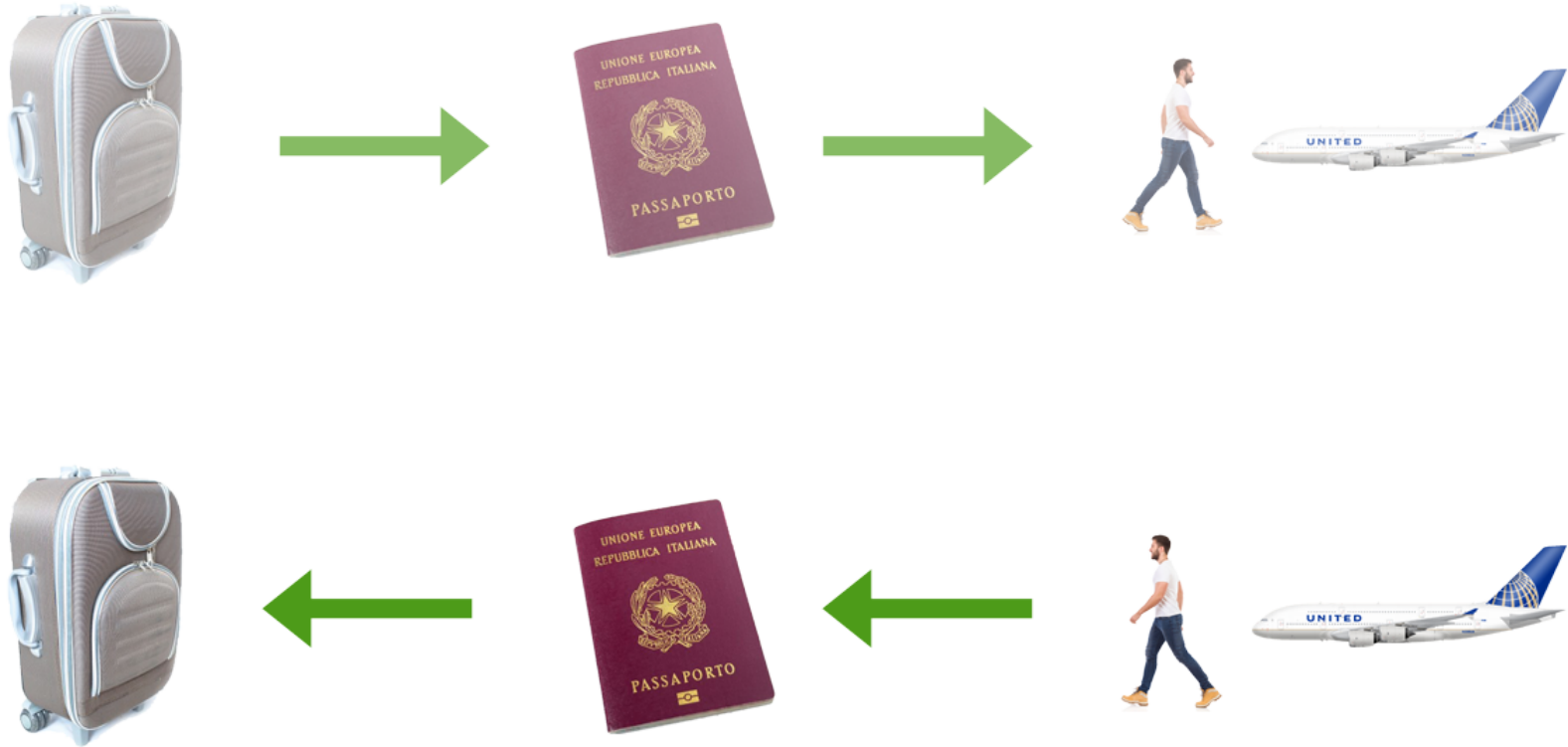
```
<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;

require 'vendor/autoload.php';

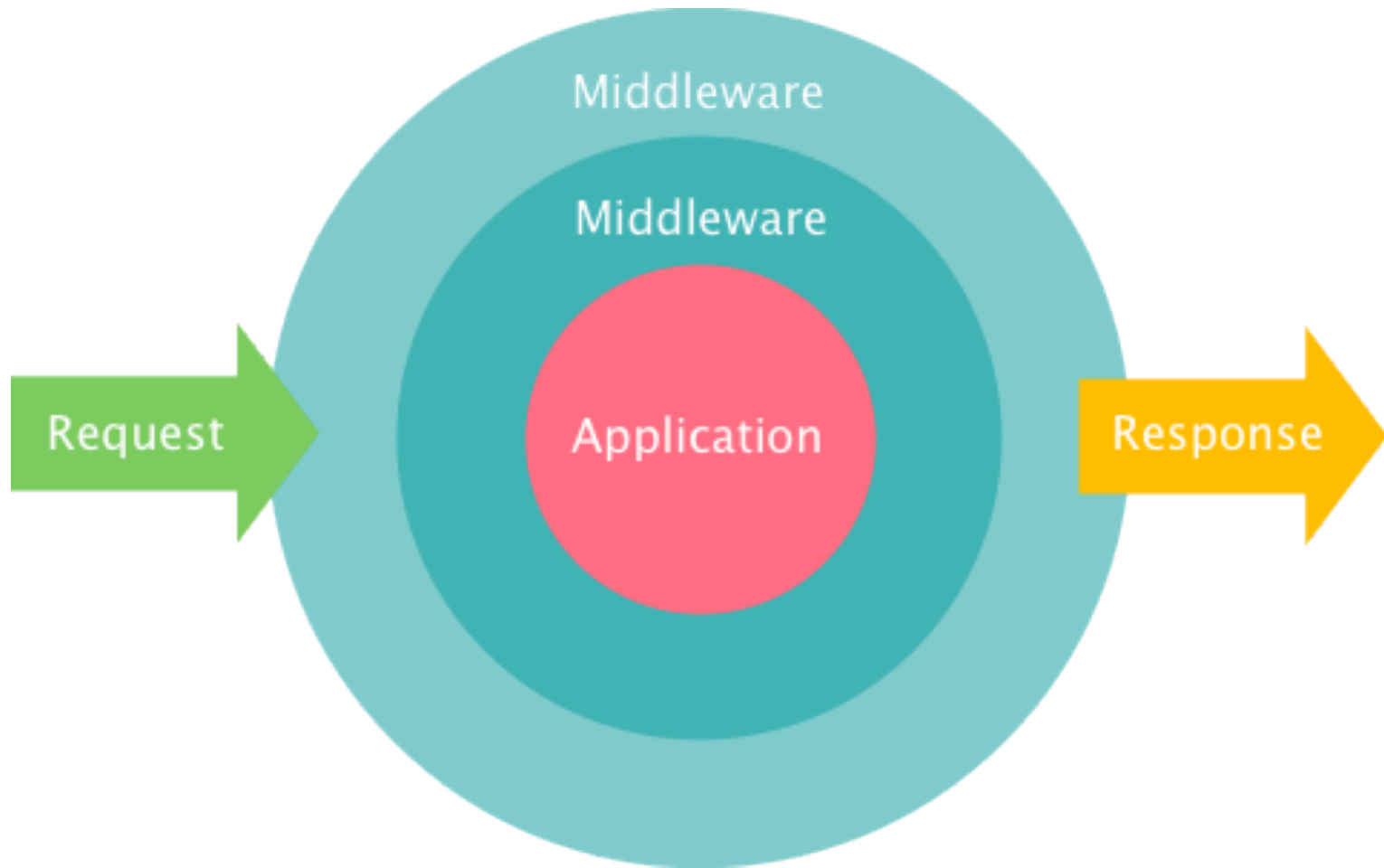
$app = new \Slim\App;
$app->get('/hello/{name}', function (Request $request, Response $response) {
    $name = $request->getAttribute('name');
    $response->getBody()->write("Hello, $name");

    return $response;
});
$app->run();
```

4. Middleware



4. Middleware (cont)



4.1. Middleware in practice

```
$app->add(new RoleExtract());  
$app->add(new JwtAuthentication($publickey));  
$app->add(new RestApiLogger());  
$app->add(new ApiRateLimit());  
$app->add(new TrustedNetworkAccess());  
$app->add(new CorsMiddleware());
```

4.1. Slim Middleware example (cont)

```
<?php

namespace Teamcrop\Middleware;

use Litpi\Registry;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;

class TrustedNetworkAccess
{
    public function __invoke(ServerRequestInterface $request, ResponseInterface $response, $next)
    {
        $trustedkey = $request->getHeader('ClientKey')[0];

        // Check whether need to check AccessTrusted
        if ($trustedkey != '') {
            $registry = Registry::getInstance();
            $conf = $registry->get('conf');

            //Check if requestor keys defined in allowkeys in config
            if (is_array($conf['trustedkey']['allowkeys'])
                && in_array($trustedkey, $conf['trustedkey']['allowkeys'])) {
                $registry->set('accesstrusted', true);
            }
        }

        $response = $next($request, $response);

        return $response;
    }
}
```

4.1. Slim Middleware example (cont)

```
<?php

namespace Teamcrop\Middleware;

use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;

class CorsMiddleware
{
    public function __invoke(ServerRequestInterface $request, ResponseInterface $response, $next)
    {
        // Get reference to application
        /** @var ResponseInterface $response */
        $response = $response->withHeader('Access-Control-Allow-Origin', '*')
            ->withHeader('Access-Control-Allow-Methods', 'PUT, GET, POST, DELETE, OPTIONS')
            ->withHeader('Access-Control-Allow-Headers', 'Authorization, Content-Type');

        if(strtoupper($request->getMethod()) == 'OPTIONS') {
            //Cache OPTIONS
            $response = $response->withHeader('Access-Control-Max-Age', 1728000);
        } else {
            // Run inner middleware and application
            $response = $next($request, $response);
        }

        return $response;
    }
}
```



Thanks!

CONTACT:

tuanmaster2012@gmail.com

0938 916 902

teamcrop.com