</> codeid

PHP ODESSA CONF

PSR

Чуть больше, чем стиль оформления кода

Анатолий Притульский
PHP-developer at Travellizy

Ссылка на слайды

Про стандарты

USB

Стандарты в PHP

PSR

PHP Standards Recommendations

Оформление кода

- Кодировка файлов UTF-8 без BOM-байта.
- Классы объявляются в `StudlyCaps`
- Методы объявляются в lower `camelCase`
- Константы объявляются в ТАКОМ_ВИДЕ
- Закрывающий тег ?> ДОЛЖЕН отсутствовать
- ...

Code MUST use 4 spaces for indenting, not tabs.

PSR

Чуть больше, чем стиль оформления кода

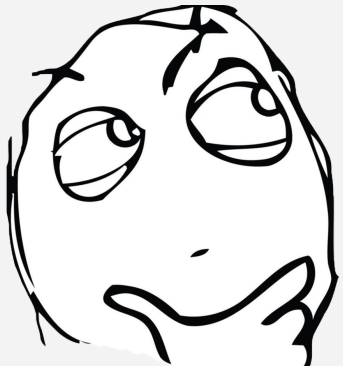
Framework Interoperability Group

https://php-fig.org

Группа была основана в 2009 году

PHP-FIG

Первый стандарт принят в 2010 году



Участники PHP-FIG

Composer, Zend Framework, Yii framework, CakePHP, Slim, PHPixie, Joomla, Magento, Drupal, phpDocumentor и др.

https://www.php-fig.org/personnel/

Бывшие участники PHP-FIG

Symfony, Laravel, Guzzle, Doctrine, Propel и др.

https://www.php-fig.org/personnel/

Почему покидают PHP-FIG?

- Дискуссии и споры
- Наследие и поддержка
- Личные предпочтения

PHP-FIG обсуждают новый стандарт

Стандарты

Это - полезно!

- Понимание
- Порядок
- Переиспользование

Обзор PSR

https://www.php-fig.org/psr/

PSR-1 / PSR-2 / PSR-12

Basic Coding Standard

Coding Style Guide

Extended Coding Style Guide

PSR-1 / PSR-2 / PSR-12

PhpStorm Reformat Code

```php
<?php
class Example {
    public function __construct() {
        $array = array(1, 2, 3);
        if (count($array) > 0)
        {
            foreach ($array as $item)
                echo $item;
        }
    }
}
```

PSR-1 / PSR-2 / PSR-12

Code Sniffer

https://github.com/FriendsOfPHP/PHP-CS-Fixer

PSR-3

Logger Interface

PSR-3

Logger Interface

```php
<?php

namespace Psr\Log;

interface LoggerInterface
{
    public function emergency($message, array $context = array());
    public function alert($message, array $context = array());
    public function critical($message, array $context = array());
    public function error($message, array $context = array());
    public function warning($message, array $context = array());
    public function notice($message, array $context = array());
    public function info($message, array $context = array());
    public function debug($message, array $context = array());
    public function log($level, $message, array $context = array());
}
```

PSR-3

Monolog

https://github.com/Seldaek/monolog

PSR-3

Нет поддержки PSR-3?

PSR-3

Используем адаптер

https://github.com/samdark/yii2-psr-log-target

~~PSR-0~~ / PSR-4

Autoloading Standard

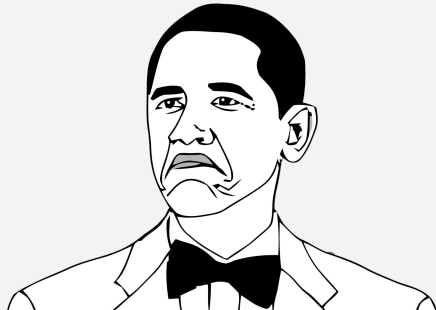
~~PSR-0~~ / PSR-4

Oo ???

```
class MyLib {
   private $CI;
   public function __construct()
   {
      $this->CI = &get_instance();

      $this->CI->load->library('encrypt');
      $this->CI->load->library('session');
      $this->CI->load->library('url');

      $this->CI->load->helper('captcha');
      $this->CI->load->helper('cookie');
      $this->CI->load->helper('email');
   }
   ...
}
```



~~PSR-0~~

deprecated с 2014 года

```php
function autoload($className)
{
    $className = ltrim($className, '\\');
    $fileName  = '';
    $namespace = '';
    if ($lastNsPos = strrpos($className, '\\')) {
        $namespace = substr($className, 0, $lastNsPos);
        $className = substr($className, $lastNsPos + 1);
        $fileName  = str_replace('\\', DIRECTORY_SEPARATOR, $namespace) . DIRECTORY_SEPARATOR;
    }
    $fileName .= str_replace('_', DIRECTORY_SEPARATOR, $className) . '.php';
    require $fileName;
}
spl_autoload_register('autoload');
```

PSR-4

Composer

https://getcomposer.org

PSR-4

Composer Autoloading

```
"autoload": {
      "psr-4": {
         "Acme\\": "src/"
      }
   }
}
```

PSR-6 / PSR-16

Caching Interface

Simple Caching Interface

PSR-16

Simple Caching Interface

```php
<?php

interface CacheInterface
{
    public function get($key, $default = null);
    public function set($key, $value, $ttl = null);
    public function delete($key);
    public function clear();
    public function getMultiple($keys, $default = null);
    public function setMultiple($values, $ttl = null);
    public function deleteMultiple($keys);
    public function has($key);
}
```

PSR-6

Caching Interface

- CacheItemInterface
- CacheItemPoolInterface

https://www.php-fig.org/psr/psr-6/

PSR-6 / PSR-16

Symfony Cache

https://github.com/symfony/cache

PSR-6 / PSR-16

Zend Cache

https://github.com/zendframework/zend-cache

PSR-7

HTTP Message Interface

PSR-7

Zend Diactoros

PSR-7 HTTP message implementations

https://github.com/zendframework/zend-diactoros

PSR-7

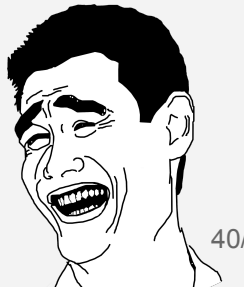
Guzzle

http://docs.guzzlephp.org/en/stable/psr7.html

PSR-7

Symfony HttpFoundation

https://github.com/symfony/psr-http-message-bridge



PSR-11

Container Interface

PSR-11

Container Interface

```php
<?php

interface ContainerInterface
{
    public function get($id);
    public function has($id);
}
```

PSR-11

Zend Expressive

https://docs.zendframework.com/zend-expressive/

PSR-11

- HomePageHandler.php
- HomePageHandlerFactory.php

```php
<?php

class HomePageHandlerFactory
{
    public function __invoke(ContainerInterface $container): RequestHandlerInterface
    {
        $router   = $container->get(RouterInterface::class);
        $template = $container->has(TemplateRendererInterface::class)
            ? $container->get(TemplateRendererInterface::class)
            : null;
        return new HomePageHandler(get_class($container), $router, $template);
    }
}
```

PSR-11

Zend DI

https://github.com/zendframework/zend-di

PSR-11

Symfony DI

https://github.com/symfony/dependency-injection

PSR-13

Hypermedia Links

PSR-13

Общий способ представления hypermedia-ссылок

PSR-13

Пример реализации

https://github.com/blongden/hal

https://github.com/Crell/HtmlModel

PSR-13

Hypermedia

то без чего ваше API не совсем REST

https://habr.com/ru/company/aligntechnology/blog/281206/

PSR-14

Event Dispatcher

PSR-14

```php
namespace Psr\EventDispatcher;
interface EventDispatcherInterface
{
    public function dispatch(object $event);
}
```

```php
namespace Psr\EventDispatcher;
interface ListenerProviderInterface
{
    public function getListenersForEvent(object $event) : iterable;
}
```

```php
namespace Psr\EventDispatcher;
interface StoppableEventInterface
{
    public function isPropagationStopped() : bool;
}
```

PSR-14

YiiSoft Event Dispatcher

https://github.com/yiisoft/event-dispatcher

PSR-14

Symfony leaves PHP-FIG

https://medium.com/@tdutrion/symfony-and-psr-14-event-dispatcher-f5a9db6740e7

PSR-14

Symfony Event Dispatcher

https://github.com/symfony/event-dispatcher

PSR-15

HTTP Handlers

PSR-15

Symfony Controller

```php
<?php

class DefaultController extends AbstractController
{
    /**
     * @Route("/", name="homepage")
     */
    public function index(): Response
    {
        $posts = $this->getDoctrine()->getRepository(Post::class)->findLatest();

        return $this->render('default/index.html.twig', [
            'posts' => $posts,
        ]);
    }
}
```

PSR-15

Symfony Controller

```php
<?php

class DefaultController extends AbstractController
{
    public function list(): Response
    {}

    public function update(): Response
    {}

    public function create(): Response
    {}

    public function delete(): Response
    {}
}
```

PSR-15

Goodbye Controllers
hello request handlers

https://jenssegers.com/85/goodbye-controllers-hello-request-handlers

PSR-15

Request Handler

```php
<?php

class CreateUserHandler
{
    public function __construct(UserRepository $repository, Twig $twig)
    {
        // ...
    }

    public function __invoke(Request $request): Response
    {
        // ...
    }
}
```

PSR-15

Request Handler

```php
<?php
interface RequestHandlerInterface
{
    public function handle(ServerRequestInterface $request): ResponseInterface;
}
```

```php
<?php
interface MiddlewareInterface
{
    public function process(
        ServerRequestInterface $request,
        RequestHandlerInterface $handler
    ): ResponseInterface;
}
```

PSR-15

Zend Expressive Handler

```php
<?php

use Psr\Http\Message\ResponseInterface;
use Psr\Http\Server\RequestHandlerInterface;
use Zend\Diactoros\Response\JsonResponse;
use Psr\Http\Message\ServerRequestInterface;

class PingHandler implements RequestHandlerInterface
{
    public function handle(ServerRequestInterface $request): ResponseInterface
    {
        return new JsonResponse(['ack' => time()]);
    }
}
```

PSR-17

HTTP Factories

PSR-17

Создать Request

```php
<?php

use Psr\Http\Message\UriInterface;
use Psr\Http\Message\RequestInterface;

interface RequestFactoryInterface
{
    public function createRequest(string $method, $uri): RequestInterface;
}
```

PSR-17

Создать Response

```php
<?php

use Psr\Http\Message\ResponseInterface;

interface ResponseFactoryInterface
{
    public function createResponse(
        int $code = 200,
        string $reasonPhrase = ''
    ): ResponseInterface;
}
```

PSR-18

HTTP Client

PSR-18

HTTP Client

```php
<?php

interface ClientInterface
{
    public function sendRequest(RequestInterface $request): ResponseInterface;
}
```

PSR-18

HTTP Client

https://github.com/zelenin/http-client

PSR-18

Guzzle Adapter

https://github.com/ricardofiorani/guzzle-psr18-adapter

PSR

рекомендации

а не жесткие правила

Абстракции не должны зависеть от деталей.

Детали должны зависеть от абстракций.


```
Created project in expressive
> ExpressiveInstaller\OptionalPackages::install
Setup data and cache dir
Setting up optional packages

  Minimal skeleton? (no default middleware, templates, or assets; configuration only)
  [y] Yes (minimal)
  [n] No (full; recommended)
  Make your selection (No):

  Which router do you want to use?
  [1] Aura.Router
  [2] FastRoute
  [3] Zend Router
  Make your selection or type a composer package name and version (FastRoute):
  - Adding package zendframework/zend-expressive-fastroute (^1.0)
  - Copying /config/autoload/routes.global.php

  Which container do you want to use for dependency injection?
  [1] Aura.Di
  [2] Pimple
  [3] Zend ServiceManager
  Make your selection or type a composer package name and version (Zend ServiceManager):
  - Adding package zendframework/zend-servicemanager (^2.7.3 || ^3.0)
  - Adding package ocramius/proxy-manager (^1.0)
  - Copying /config/container.php

  Which template engine do you want to use?
  [1] Plates
  [2] Twig
  [3] Zend View installs Zend ServiceManager
  [n] None of the above
  Make your selection or type a composer package name and version (n): 1
  - Adding package zendframework/zend-expressive-platesrenderer (^1.0)
  - Copying /config/autoload/templates.global.php
  - Copying /templates/error/404.phtml
  - Copying /templates/error/error.phtml
  - Copying /templates/layout/default.phtml
  - Copying /templates/app/home-page.phtml

  Which error handler do you want to use during development?
  [1] Whoops
  [n] None of the above
  Make your selection or type a composer package name and version (Whoops): 1
  - Adding package filp/whoops (^1.1)
  - Copying /config/autoload/errorhandler.local.php
```

Вопросы

https://nepster.pro