

Воронеж, 24.03.2023

# Конфигурация приложения в Yii3



**Сергей Предводителей**

# Обо мне



PHP-разработчик



Участник Yii Core Team



Авторский канал [@sergei\\_predvoditelev](https://t.me/@sergei_predvoditelev)

Хроники Yii3 [@yii3chronicles](https://t.me/@yii3chronicles)

# Что конфигурируем?

- Контейнер зависимостей
- События
- Виджеты
- ...

# Что такое конфигурация?

```
[
  RequestFactoryInterface::class => RequestFactory::class,
  ServerRequestFactoryInterface::class => ServerRequestFactory::class,
  ResponseFactoryInterface::class => ResponseFactory::class,
  StreamFactoryInterface::class => StreamFactory::class,
  UriFactoryInterface::class => UriFactory::class,
  UploadedFileFactoryInterface::class => UploadedFileFactory::class,

  LoggerInterface::class => [
    'class' => Logger::class,
    '__construct()' => [
      'targets' => ReferencesArray::from([
        FileTarget::class,
      ]),
    ],
  ],

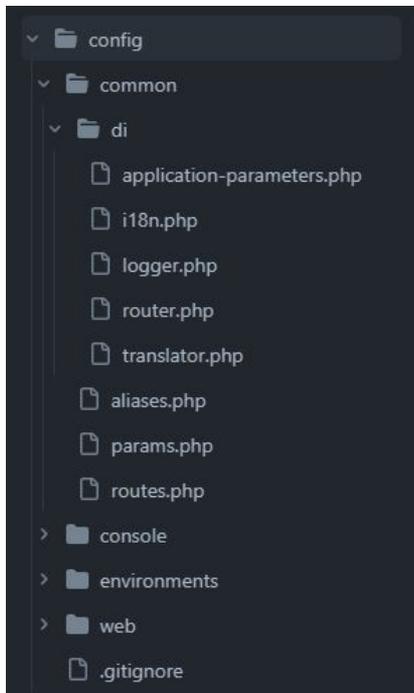
  Aliases::class => [
    '__construct()' => [$params['yiisoft/aliases']['aliases']],
  ],
]
```

```
[
  ApplicationStartup::class => [
    [Debugger::class, 'startup'],
    [ConsoleAppInfoCollector::class, 'collect'],
  ],
  ApplicationShutdown::class => [
    [ConsoleAppInfoCollector::class, 'collect'],
    [Debugger::class, 'shutdown'],
  ],
  ConsoleCommandEvent::class => [
    [ConsoleAppInfoCollector::class, 'collect'],
    [CommandCollector::class, 'collect'],
  ],
  ConsoleErrorEvent::class => [
    [ConsoleAppInfoCollector::class, 'collect'],
    [CommandCollector::class, 'collect'],
  ],
  ConsoleTerminateEvent::class => [
    [ConsoleAppInfoCollector::class, 'collect'],
    [CommandCollector::class, 'collect'],
  ],
];
```

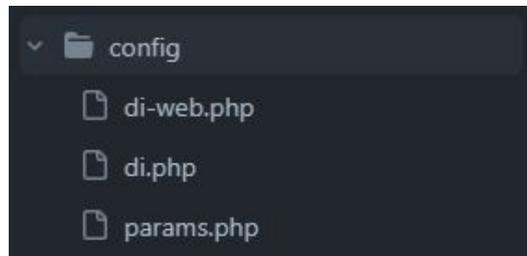
Массив

# Где лежит конфигурация?

Приложение



Подключаемые пакеты



Конфигурация пакета  
по умолчанию

# Yii Config

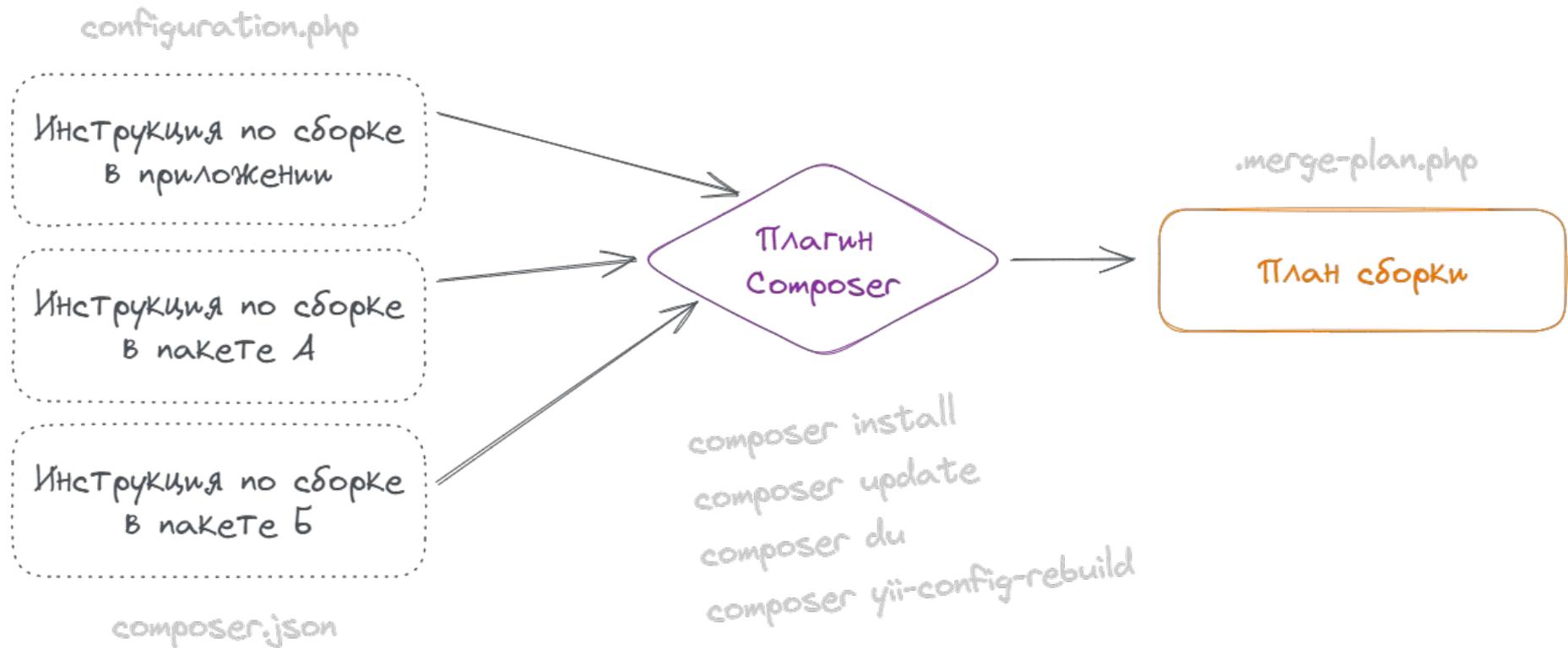


Плагин Composer

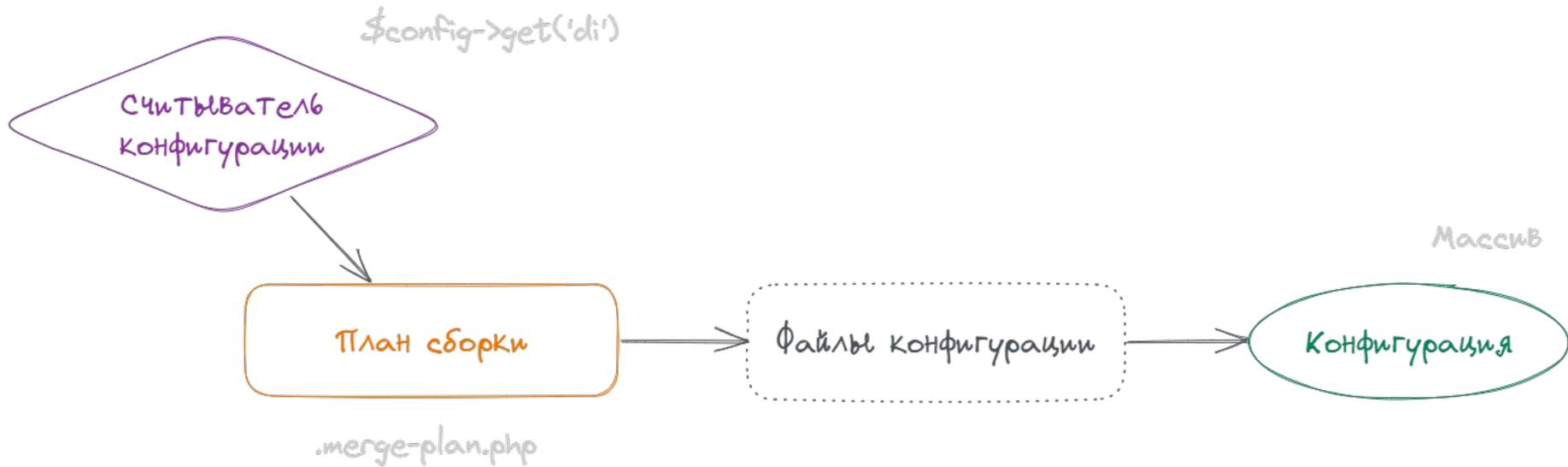
Считыватель конфигурации

<https://github.com/yiisoft/config>

# Логика работы



# Логика работы



# Инструкция по сборке

## PHP-файл

```
return [  
    'config-plugin' => [  
        'params' => [  
            'common/params.php',  
        ],  
        'params-web' => [  
            '$params',  
            'web/params.php',  
        ],  
        'di' => 'common/di/*.php',  
        'di-web' => [  
            '$di',  
            'web/di/*.php',  
        ],  
        // ...  
    ],  
    'config-plugin-environments' => [  
        'dev' => [  
            'params' => 'environments/dev/params.php',  
        ],  
        'prod' => [  
            'params' => 'environments/prod/params.php',  
        ],  
        // ...  
    ],  
    'config-plugin-options' => [  
        'source-directory' => 'config',  
    ],  
];
```

## composer.json

```
{  
    "extra": {  
        "config-plugin-options": {  
            "source-directory": "config"  
        },  
        "config-plugin": {  
            "params": "params.php",  
            "di": "di.php",  
            "di-web": "di-web.php"  
        }  
    }  
}
```

- **Произвольная** структура файлов
- Структура файлов в пакетах **не зависит** от структуры файлов в приложении

# Когда отработывает Composer плагин?

- `composer install`
- `composer update`
- `composer dump-autoload`
- `composer yii-config-rebuild`

```
// Do not edit. Content will be replaced.
```

```
return [  
    '/' => [  
        'di' => [...],  
        'params' => [...],  
        'bootstrap' => [...],  
        'di-web' => [  
            // ///  
            'yiiisoft/view' => [  
                'config/di-web.php',  
            ],  
            'yiiisoft/yii-event' => [  
                'config/di-web.php',  
            ],  
            '/' => [  
                '$di',  
                'web/di/*.php',  
            ],  
        ],  
        'di-console' => [...],  
        'events-console' => [...],  
        'di-delegates' => [...],  
        'bootstrap-web' => [...],  
        'routes' => [...],  
        // ...  
    ],  
    'dev' => [  
        'params' => [  
            '/' => [  
                'environments/dev/params.php',  
            ],  
        ],  
    ],  
    'prod' => [...],  
    'test' => [...],  
];
```

# План сборки

- Должен быть в .gitignore
- Нельзя редактировать вручную

.merge-plan.php

# Считыватель конфигурации

```
$config = new Config(new ConfigPaths(__DIR__));  
$config->get('di');
```

---

По плану сборки объединяет все файлы  
для запрашиваемой конфигурации  
и возвращает итоговый массив

# Параметры

Особая конфигурация, которая собирается и передаётся внутри всех файлов при сборке других конфигураций.

```
return [
    'app' => [
        'charset' => 'UTF-8',
        'locale' => 'en',
        'name' => 'My Project',
    ],

    'yiiisoft/aliases' => [
        'aliases' => require __DIR__ . '/aliases.php',
    ],

    'yiiisoft/translator' => [
        'locale' => 'en',
        'fallbackLocale' => 'en',
        'defaultCategory' => 'app',
    ],

    'yiiisoft/yii-view' => [
        'injections' => [
            Reference::to(CommonViewInjection::class),
            Reference::to(CsrfViewInjection::class),
            Reference::to(LayoutViewInjection::class),
            Reference::to(TranslatorViewInjection::class),
        ],
    ],
];
```

params.php

```
/* @var array $params */

return [
    Aliases::class => [
        '__construct()' => [
            $params['yiiisoft/aliases']['aliases'],
        ],
    ],
];
```

di.php

# Слои

- Конфигурация в пакетах (vendor)
- Конфигурация в пакетах (приложение)
- Конфигурация в приложении
- Конфигурация окружения

# Запрет дублирования

**Запрещено** дублирование ключей  
в рамках одного слоя — будет исключение

- **Отсутствие** проблемы приоретизации  
Не нужно решать, кто кого должен переопределять.
- **Однозначность** в процессе разработки  
Даёт уверенность, что в рамках слоя не будет переопределения.

# Окружения

```
new Config(  
    new ConfigPaths(__DIR__),  
    environment: 'dev',  
);
```

---

Используется  
только в приложении

```
"extra": {  
    "config-plugin-options": {  
        "source-directory": "config"  
    },  
    "config-plugin": {  
        "params": "params.php",  
        "web": "web.php"  
    },  
    "config-plugin-environments": {  
        "dev": {  
            "params": "dev/params.php",  
            "app": [  
                "$web",  
                "dev/app.php"  
            ]  
        },  
        "prod": {  
            "app": "prod/app.php"  
        }  
    }  
}
```

# Конфигурация в пакетах (приложение)

vendor-override-layer

```
"require": {
    "vendor-name/app-1": "^1.0",
    "vendor-name/app-2": "^2.0",
    "vendor-name/helper": "^1.3",
    "yiisoft/config": "^1.0" 1.3.0
},
"extra": {
    "config-plugin-options": {
        "vendor-override-layer": [
            "vendor-name/app-*",
            "vendor-name/helper"
        ]
    }
},
```

# Модификаторы

```
new Config(  
  new ConfigPaths(__DIR__, 'config'),  
  modifiers: [  
    RecursiveMerge::groups('events'),  
    RecursiveMerge::groups('params', 'events'),  
  ],  
);
```

Меняют поведение  
считывателя конфигурации

# Рекурсивное объединение

```
RecursiveMerge::groups('params', 'events')
```

app/params

```
return [  
  'key' => [  
    'a' => 1,  
    'b' => 2,  
  ],  
];
```

+

vendor/params

```
return [  
  'key' => [  
    'c' => 3,  
    'd' => 4,  
  ],  
];
```

=

result

```
[  
  'key' => [  
    'c' => 3,  
    'd' => 4,  
    'a' => 1,  
    'b' => 2,  
  ],  
];
```

---

Используется для параметров и событий

# Реверсивное объединение

```
ReverseMerge::groups('events')
```

app/events

```
return [  
  'a' => 1,  
  'b' => 2,  
];
```

+

vendor/events

```
return [  
  'c' => 3,  
  'd' => 4,  
];
```

=

result

```
[  
  'a' => 1,  
  'b' => 2,  
  'c' => 3,  
  'd' => 4,  
];
```

---

Используется для событий

# Удаление конфигурации из пакета

```
RemoveFromVendor::keys(['c'], ['d'])  
    ->package('yiiisoft/view', 'params'),
```

---

```
RemoveFromVendor::groups([  
    'yiiisoft/auth' => '*',  
    'yiiisoft/http' => 'params',  
    'yiiisoft/view' => ['params', 'di'],  
]),
```

# Дополнительные настройки

- `source-directory`
- `merge-plan-file / $mergePlanFile`
- `build-merge-plan`
- `$paramsGroup`
- `$configDirectory`
- `$vendorDirectory`

# Yii Config можно использовать вне Yii

```
"require": {  
    "php": "^8.0",  
    "composer-plugin-api": "^2.0",  
    "yiisoft/arrays": "^3.0",  
    "yiisoft/strings": "^2.0",  
    "yiisoft/var-dumper": "^1.1"  
},
```

- Минимум зависимостей
- Легко интегрировать

# Ссылки

## Yii Config

<https://github.com/yiisoft/config>

## Концепция конфигурации в Yii3 (EN)

<https://github.com/yiisoft/docs/blob/master/guide/en/concept/configuration.md>

## Соглашение об именовании конфигураций (EN)

<https://github.com/yiisoft/docs/blob/master/022-config-groups.md>

## Конфигурация Yii на пальцах (RU)

<https://predvoditelev.ru/notes/yii-configuration/>

# Вопросы?

Конфигурация приложения в Yii3

---

## Сергей Предводителей

Для связи: [sergei@predvoditelev.ru](mailto:sergei@predvoditelev.ru)

GitHub: <https://github.com/vjik>

Авторский канал: [https://t.me/sergei\\_predvoditelev](https://t.me/sergei_predvoditelev)

Хроники Yii3: <https://t.me/yii3chronicles>