Migrating from Symfony 5 to 6

SensioLabs
Créateur de Symfony

Program

SemVer?

What is SemVer ?

- The SemVer, for Semantic Versionning, is a specification that aims to unify applications, libraries, and packages version numbers, their format and incrementation.
- Symfony and its components follow the SemVer, adding a specific schedule and agenda to each version release

https://semver.org/

What is SemVer ?

SemVer versions are composed of three parts

X.Y.Z as Major.Minor.Patch

- **Major: X**

  removing deprecated features every 2 years
  updating minimum PHP requirements
- **Minor: 5.Y**

  background compatible features and changes every 6 months

  adding new deprecations to prepare the next Major
- **Patch: 5.1.Z**

  bug and security fixes, every month

Note: LTS

The minor versions X.4 are called LTS versions, for Long Term Support.

When every other minor versions (X.0, X.1, X.2, X.3) are update and patched during 1 year, LTS versions benefit from 3 years of updates and fixes plus 1 additional year of security fixes only.

Backward Compatibility Promise

Backward Compatibility Promise

- In respect of the SemVer specification, Symfony promises Backward Compatibility (BC) for every minor version since Symfony 2.3
- Practically, Symfony ensures every minor version change inside a major version will not break your code
- This comes at certain conditions that you can check in the documentation

https://symfony.com/doc/current/contributing/code/bc.html

Migration Path

Migration Path

All other factors set aside, your migration path may change greatly depending on:

- Your current version
- Your target version
- The time you have for the migration
- The estimated number of deprecations in your code

Migration Path

To estimate the size of your migration, check the number of deprecations in your code:

- Check the changelogs for deprecation announcements
- Check the various tools given in the chapter “Deprecations”

Also, keep in mind that you don’t have to upgrade one minor at the time (ex 5.1 to 5.2), or to a first major (ex 5.Y to 6.0).

Your migration can jump any minor you don’t specifically need (ex 5.2 to 5.4 or 5.4 to 6.1).

Migration Path

The recommended way to migrate from one minor version to the next major version (ex from 5.2 to 6.0) would be:

- Upgrade Symfony to the next LTS version (X.4)
- Make your code deprecation free
- Upgrade Symfony to the next major version ({X+1}.0)

This last step should be seamless, as new major versions are feature-identical to LTS versions, with the deprecation layer removed

Migration Path

The previous migration path as the advantage to be short. But it leaves you with some technical debt as you haven’t updated your code to use Symfony’s latest features.

If you have more time, you can also use the following path:

- Upgrade Symfony to the next minor version (X.{current+1})
- Incorporate the new features into your code
- Make your code deprecation free
- Repeat until you reach the desired version

Changelog

Changelog

The official changelog files provide you with an easy way to check the differential between your version and your target version.

You can find them on the official Symfony repository on GitHub.

You can also find there UPGRADE-{version}.md files to help you check your migration path.

Here follow examples of changelogs.

About the changelogs

The following changelogs are partials and opinionated.
There are much more changes between most versions.
Only “big” minor changes have been kept in these lists.

This choice has been made to give a sense of the evolution of Symfony across a single major version change, and which changes you might expect.

Symfony 5.1

- New loginUser() method for tests
- Uid component
- Server-side request forgery protection in HttpClient
- Deprecated the Inflector component
- Form improvements
- New Security system
- Misc. improvements: constants for command exit codes, stand-alone YAML linter command, better RoundRobin mailer transport, separate log channel for deprecations, severity in ConstraintViolationList, new deprecation contract, "dark mode" in exception pages.

https://github.com/symfony/symfony/blob/5.4/CHANGELOG-5.1.md

Symfony 5.2

- Doctrine types for UUID and ULID
- TranslatableMessage objects
- PHP 8 attributes support
- Session profiling
- Uid serialization and validation
- Form mapping callbacks
- Rate Limiter component, Login Throttling, and Login Links
- Constraints as PHP attributes
- Semaphore component
- Notifier improvements

https://github.com/symfony/symfony/blob/5.4/CHANGELOG-5.2.md

Symfony 5.3

- PasswordHasher Component
- Form Handler Helper (renderForm)
- Improvements for Security Users (UserInterface simplification, username to userIdentifier)
- UID Improvements
- Session Service Deprecation
- Configure Multiple Environments in a Single File (when@{env})
- Prototype Options
- CSRF Tokens randomizing (against BREACH attacks)
- Translation Providers
- Notifier Integrations

https://github.com/symfony/symfony/blob/5.4/CHANGELOG-5.3.md

Symfony 5.4

- Console Autocompletion
- Controller Changes and deprecations
- Route Aliasing
- PHP Enumerations Support
- Translation Improvements
- DependencyInjection Improvements (union types autowiring…)
- Messenger Improvements
- Serializer improvements
- Misc. features: new strict mode for assets, new Nil ULID class, improved gitignore support in Finder, new command debug:dotenv, simplified creation of constraint errors…

https://github.com/symfony/symfony/blob/5.4/CHANGELOG-5.4.md

Symfony 6.0

Symfony 6.0 doesn't provide any new feature or change compared to Symfony 5.0.

BUT

All deprecations from Symfony 5.X versions have been removed in 6.0, and the minimum PHP version requirement has been upped to PHP 8.0

https://github.com/symfony/symfony/blob/5.4/UPGRADE-6.0.md

Fix the Stack

PHP Version

- The minimum version to run Symfony 6.0 application has been raised to PHP 8.0.
- Beware though that Symfony 6.1 also exceptionally raised its minimum PHP version to 8.1.
- Also, if not already done, switch your Composer version to 2.X and Symfony Flex version to 2.X
- Don’t forget to update Flex’s Composer recipes after upgrading Symfony

Deprecations

Deprecations ?

Deprecations are a way to mark some code as obsolete. They allow you to upgrade your application easily by giving you a warning as to which functions and classes will be removed in the future so you can replace them.

You can check for them in many places.

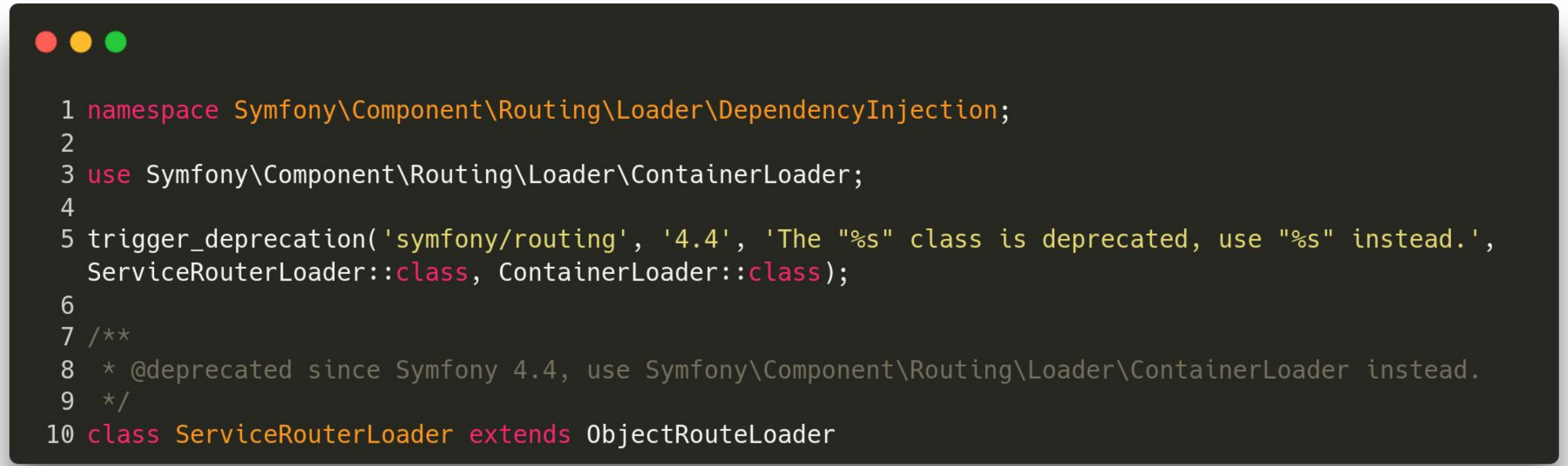
Deprecations ?

Practically, deprecations are silenced error triggered when you use code that will be removed later.

Usually, some code is marked as deprecated during a minor version update (version X.{any}).

In the next major version, the code marked as deprecated in the previous version is removed (in the version {X+1}.0).

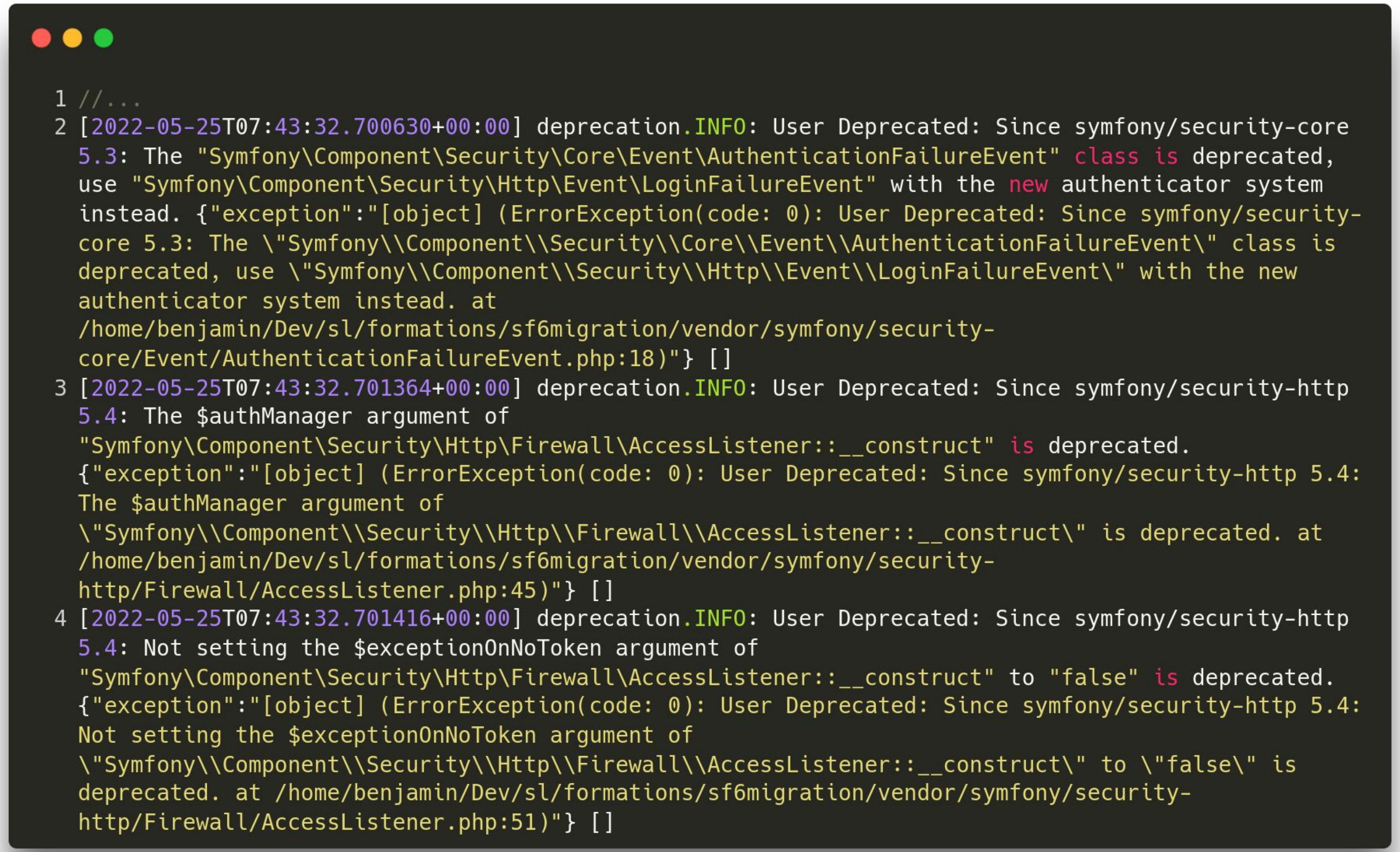
Deprecations ?



```php
namespace Symfony\Component\Routing\Loader\DependencyInjection;

use Symfony\Component\Routing\Loader\ContainerLoader;

trigger_deprecation('symfony/routing', '4.4', 'The "%s" class is deprecated, use "%s" instead.',
ServiceRouterLoader::class, ContainerLoader::class);

/**
 * @deprecated since Symfony 4.4, use Symfony\Component\Routing\Loader\ContainerLoader instead.
 */
class ServiceRouterLoader extends ObjectRouteLoader
```

Where to find them ? Logs

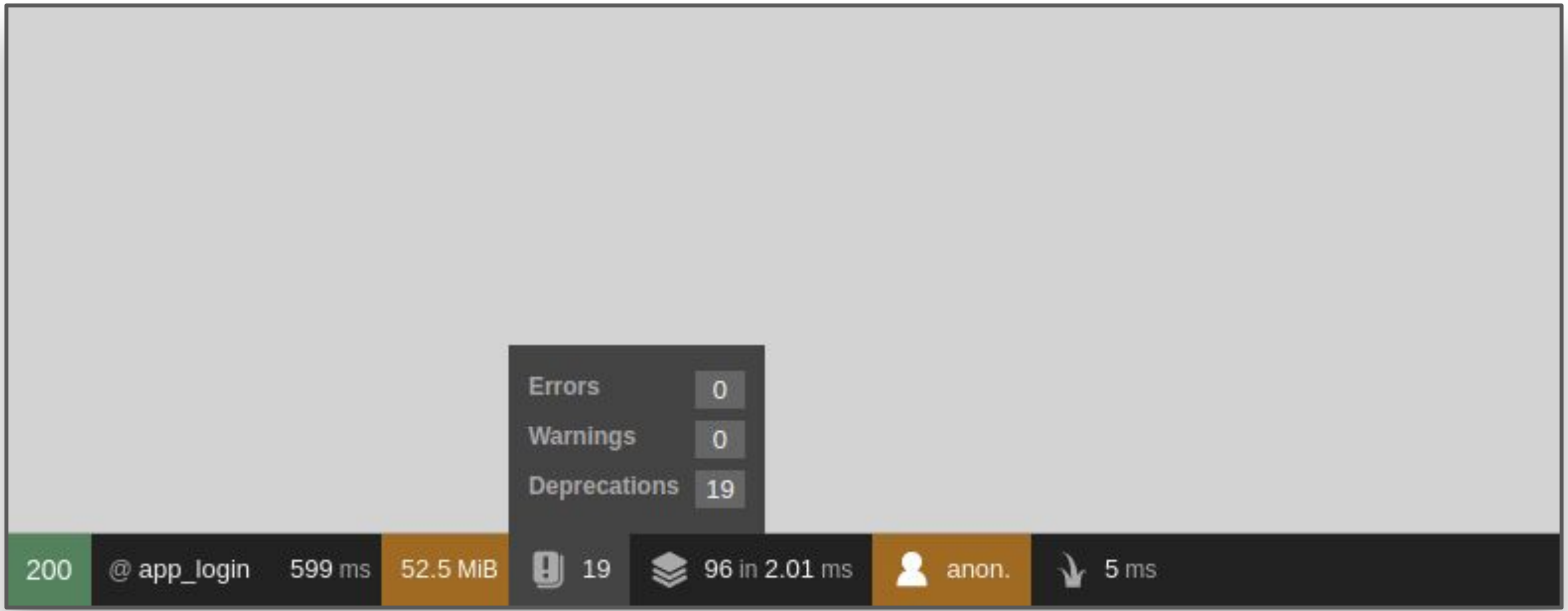
In your env’s log file (ex: ./var/log/dev.log):



```
//...
[2022-05-25T07:43:32.700630+00:00] deprecation.INFO: User Deprecated: Since symfony/security-core
5.3: The "Symfony\Component\Security\Core\Event\AuthenticationFailureEvent" class is deprecated,
use "Symfony\Component\Security\Http\Event\LoginFailureEvent" with the new authenticator system
instead. {"exception":"[object] (ErrorException(code: 0): User Deprecated: Since symfony/security-
core 5.3: The \"Symfony\\Component\\Security\\Core\\Event\\AuthenticationFailureEvent\" class is
deprecated, use \"Symfony\\Component\\Security\\Http\\Event\\LoginFailureEvent\" with the new
authenticator system instead. at
/home/benjamin/Dev/sl/formations/sf6migration/vendor/symfony/security-
core/Event/AuthenticationFailureEvent.php:18)"} []
[2022-05-25T07:43:32.701364+00:00] deprecation.INFO: User Deprecated: Since symfony/security-http
5.4: The $authManager argument of
"Symfony\Component\Security\Http\Firewall\AccessListener::__construct" is deprecated.
{"exception":"[object] (ErrorException(code: 0): User Deprecated: Since symfony/security-http 5.4:
The $authManager argument of
\"Symfony\\Component\\Security\\Http\\Firewall\\AccessListener::__construct\" is deprecated. at
/home/benjamin/Dev/sl/formations/sf6migration/vendor/symfony/security-
http/Firewall/AccessListener.php:45)"} []
[2022-05-25T07:43:32.701416+00:00] deprecation.INFO: User Deprecated: Since symfony/security-http
5.4: Not setting the $exceptionOnNoToken argument of
"Symfony\Component\Security\Http\Firewall\AccessListener::__construct" to "false" is deprecated.
{"exception":"[object] (ErrorException(code: 0): User Deprecated: Since symfony/security-http 5.4:
Not setting the $exceptionOnNoToken argument of
\"Symfony\\Component\\Security\\Http\\Firewall\\AccessListener::__construct\" to \"false\" is
deprecated. at /home/benjamin/Dev/sl/formations/sf6migration/vendor/symfony/security-
http/Firewall/AccessListener.php:51)"} []
```

Where to find them ? Profiler

In the Profiler:



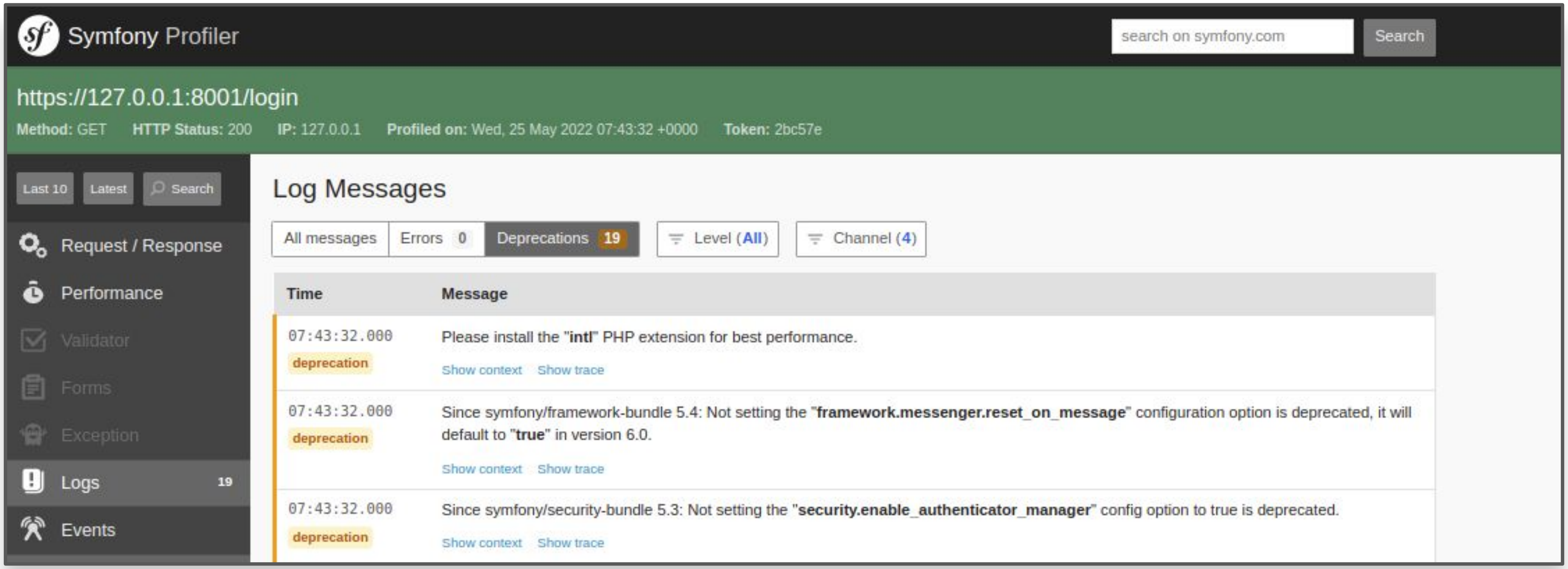
```
Errors        0
Warnings      0
Deprecations  19
```

200 | @ app_login | 599 ms | 52.5 MiB | 19 | 96 in 2.01 ms | anon. | 5 ms

SensioLabs

Where to find them ? Profiler

In the Profiler:



https://127.0.0.1:8001/login

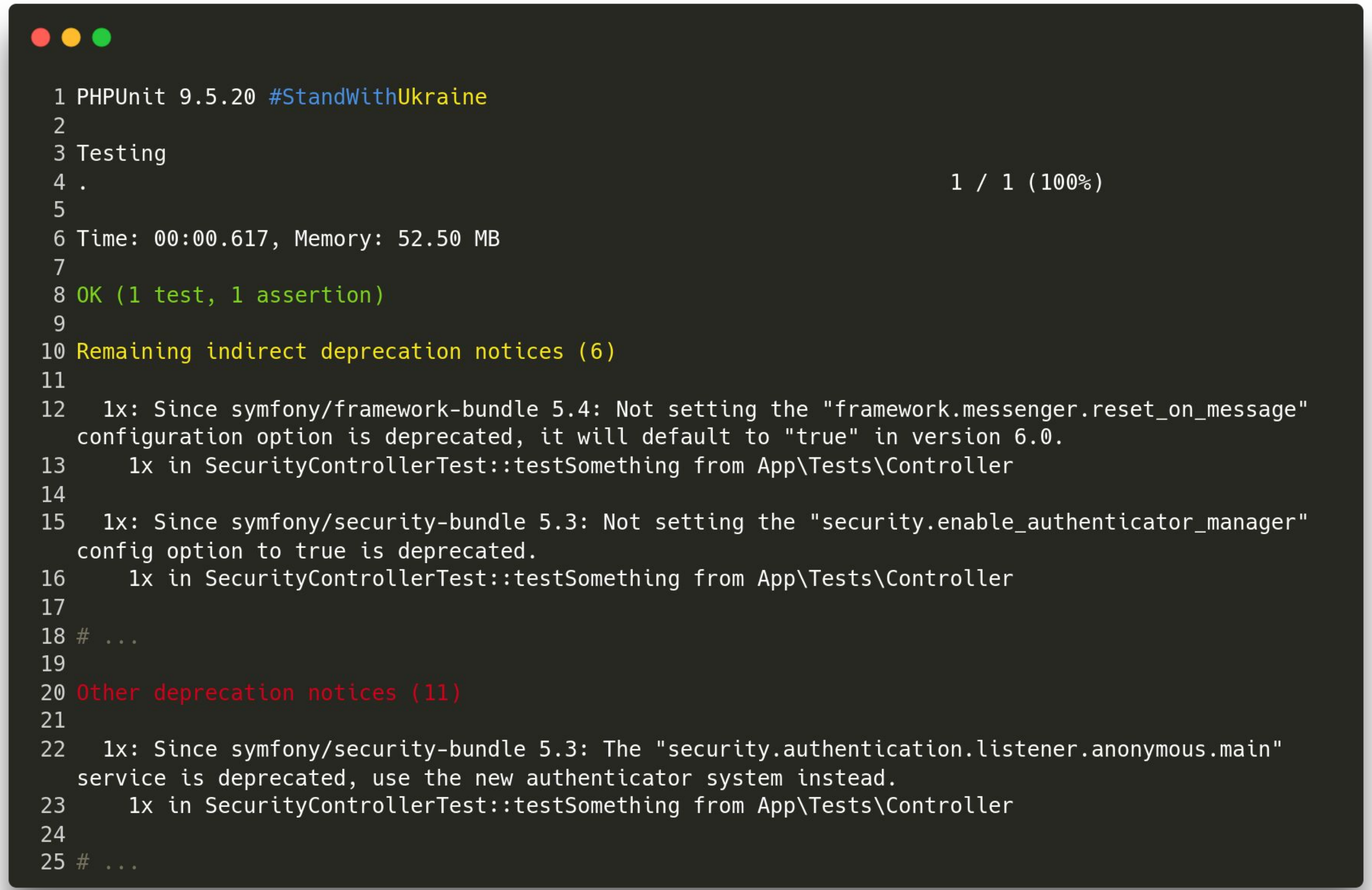
Method: GET **HTTP Status:** 200 **IP:** 127.0.0.1 **Profiled on:** Wed, 25 May 2022 07:43:32 +0000 **Token:** 2bc57e

Log Messages

Time	Message
07:43:32.000 deprecation	Please install the "**intl**" PHP extension for best performance. Show context Show trace
07:43:32.000 deprecation	Since symfony/framework-bundle 5.4: Not setting the "**framework.messenger.reset_on_message**" configuration option is deprecated, it will default to "**true**" in version 6.0. Show context Show trace
07:43:32.000 deprecation	Since symfony/security-bundle 5.3: Not setting the "**security.enable_authenticator_manager**" config option to true is deprecated. Show context Show trace

Where to find them ? PHPUnit

When you run PHPUnit:



```
PHPUnit 9.5.20 #StandWithUkraine

Testing
.                                                         1 / 1 (100%)

Time: 00:00.617, Memory: 52.50 MB

OK (1 test, 1 assertion)

Remaining indirect deprecation notices (6)

  1x: Since symfony/framework-bundle 5.4: Not setting the "framework.messenger.reset_on_message"
configuration option is deprecated, it will default to "true" in version 6.0.
    1x in SecurityControllerTest::testSomething from App\Tests\Controller

  1x: Since symfony/security-bundle 5.3: Not setting the "security.enable_authenticator_manager"
config option to true is deprecated.
    1x in SecurityControllerTest::testSomething from App\Tests\Controller

# ...

Other deprecation notices (11)

  1x: Since symfony/security-bundle 5.3: The "security.authentication.listener.anonymous.main"
service is deprecated, use the new authenticator system instead.
    1x in SecurityControllerTest::testSomething from App\Tests\Controller

# ...
```

SensioLabs

Deprecations: getting rid of them

When these different tools show a deprecation, 2 different cases may arise:

- The deprecation comes from code you wrote.
- The deprecation comes from a vendor.

Deprecations: getting rid of them

- If the deprecation comes from code you wrote (ex: you called a deprecated function or class in one of your services), you need to refactor it to use the replacement usually given in the deprecation.
- If the deprecation comes from a vendor, usually upgrading said vendor will be enough to remove the depreciation. If not, you may have to find a replacement package.

Focus: The Security Component

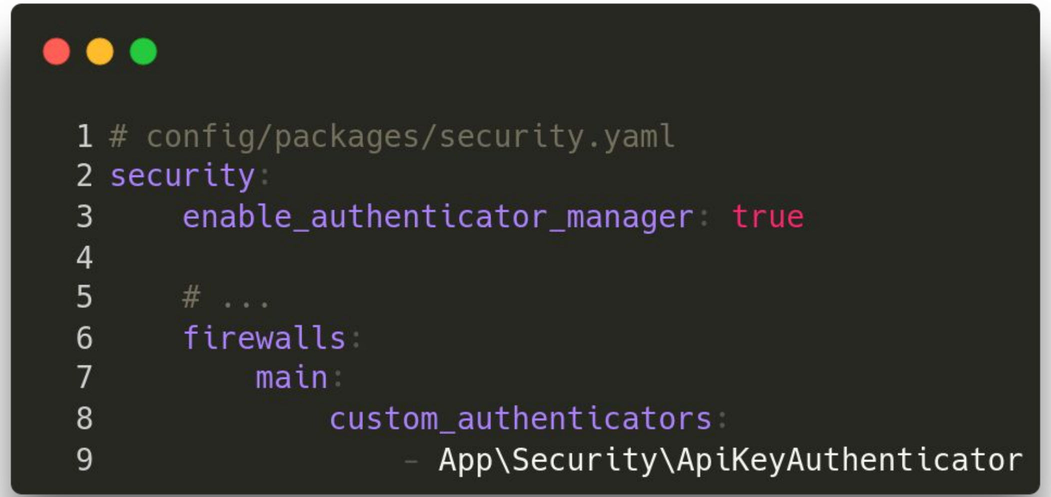
Security Component

- Symfony 5.1 introduced a whole new experimental Security system.
- In Symfony 5.3, the old Guard system and component was deprecated, and removed in Symfony 6.0

Security Component

- This new system has to be enabled in your configuration files (./config/packages.security.yaml)
- Your GuardAuthenticators will have to be converted to new Authenticators and added as custom authenticators
- You will have to remove the `anonymous: true` parts in your firewalls

Security Component



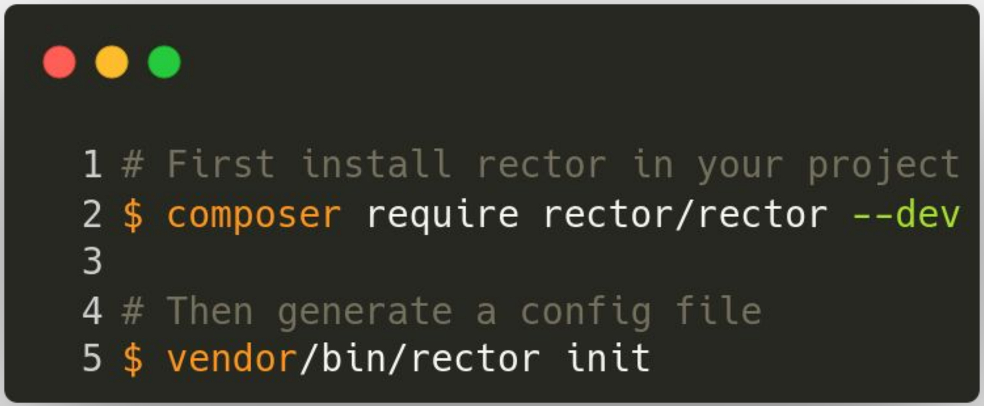
```yaml
# config/packages/security.yaml
security:
    enable_authenticator_manager: true

    # ...
    firewalls:
        main:
            custom_authenticators:
                - App\Security\ApiKeyAuthenticator
```

Help: RectorPHP

RectorPHP

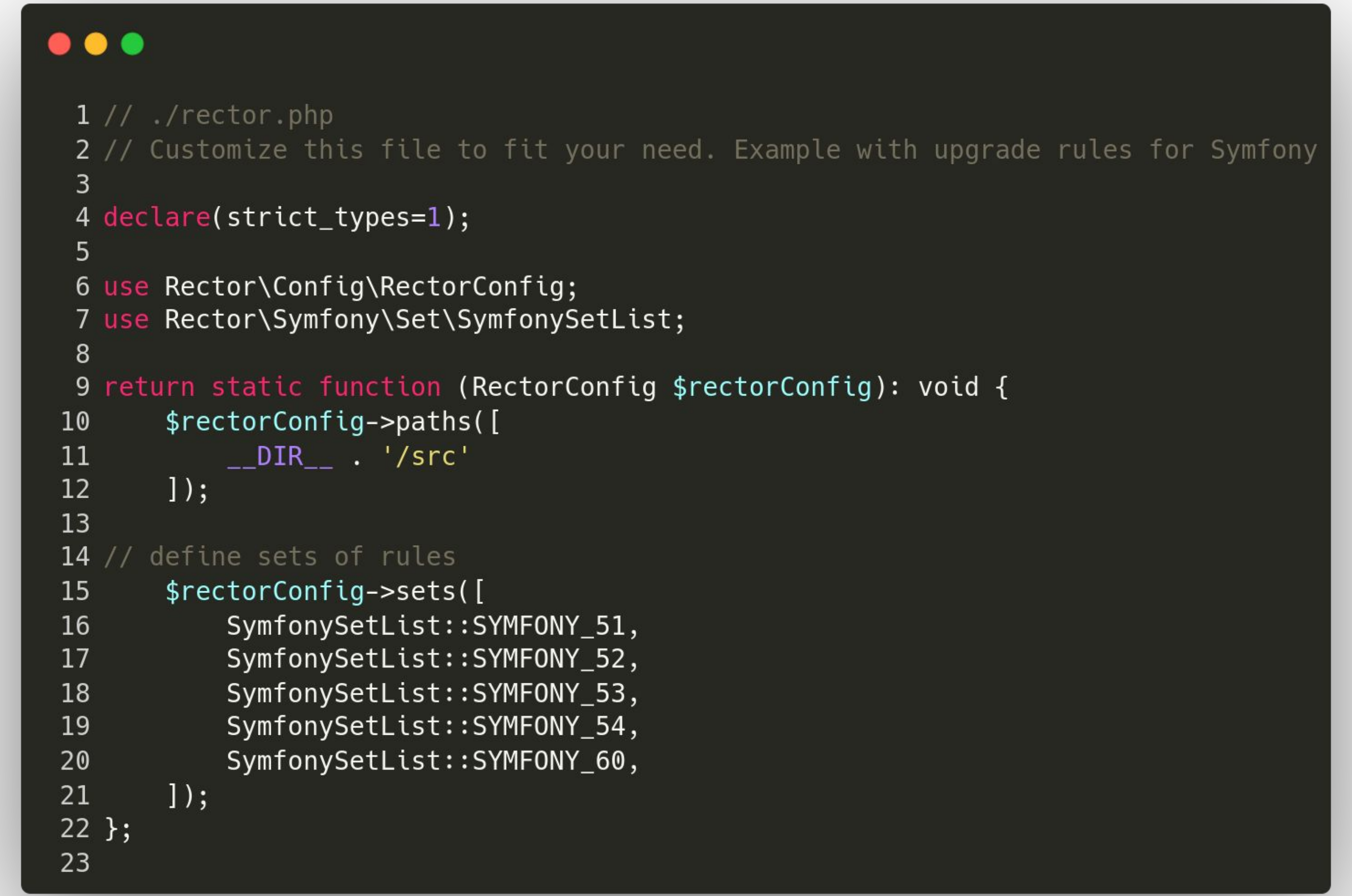
- Rector is an automated upgrade and refactoring tool for PHP.
- It allows by means of a simple configuration file to define rules, or Rectors, to be applied to your code.
- The Rector engine will then parse your code and apply a refactoring each time your code matches a Rector.
- Predefined Rectors include rules to upgrade your PHP versions from 5.3 up until 8.1, rules for code styling and PSRs, and rules for many PHP projects including Symfony and Doctrine.

Rector



```
# First install rector in your project
$ composer require rector/rector --dev

# Then generate a config file
$ vendor/bin/rector init
```

Rectors

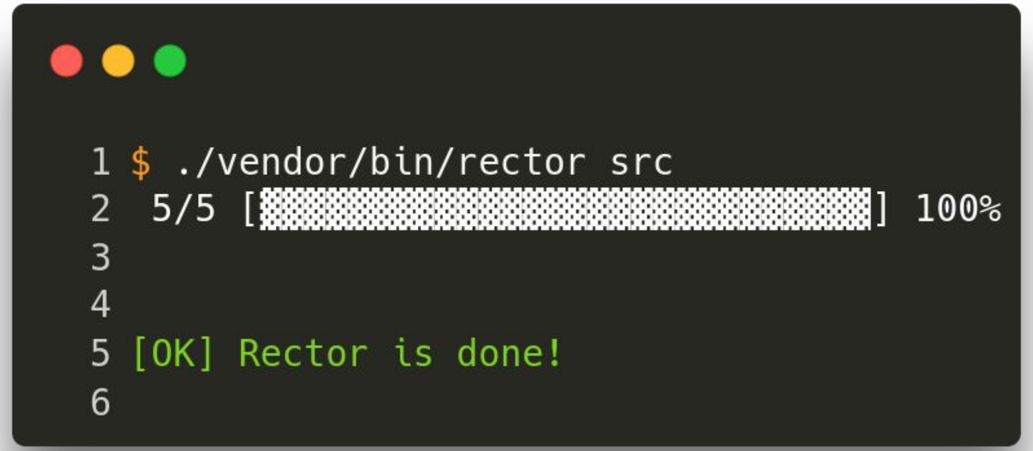


```php
// ./rector.php
// Customize this file to fit your need. Example with upgrade rules for Symfony

declare(strict_types=1);

use Rector\Config\RectorConfig;
use Rector\Symfony\Set\SymfonySetList;

return static function (RectorConfig $rectorConfig): void {
    $rectorConfig->paths([
        __DIR__ . '/src'
    ]);

// define sets of rules
    $rectorConfig->sets([
        SymfonySetList::SYMFONY_51,
        SymfonySetList::SYMFONY_52,
        SymfonySetList::SYMFONY_53,
        SymfonySetList::SYMFONY_54,
        SymfonySetList::SYMFONY_60,
    ]);
};

```

SensioLabs

Rector

Then run Rector:



```
$ ./vendor/bin/rector src
 5/5 [▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓] 100%


[OK] Rector is done!

```

Rector

Don’t forget to manually check the changes made by Rector anyway.

You can also run PHP-CS-Fixer to help clean Rector’s job.

Resources

https://semver.org/

https://symfony.com/doc/current/setup/upgrade_major.html

https://github.com/symfony/symfony/blob/6.0/UPGRADE-6.0.md

https://symfony.com/doc/current/contributing/code/bc.html

https://github.com/rectorphp/rector-symfony

@sensiolabs
sensiolabs.com

SensioLabs
Créateur de Symfony