Say Hello to Symfony 5

nfq Summit – January 10th 2020

about:me

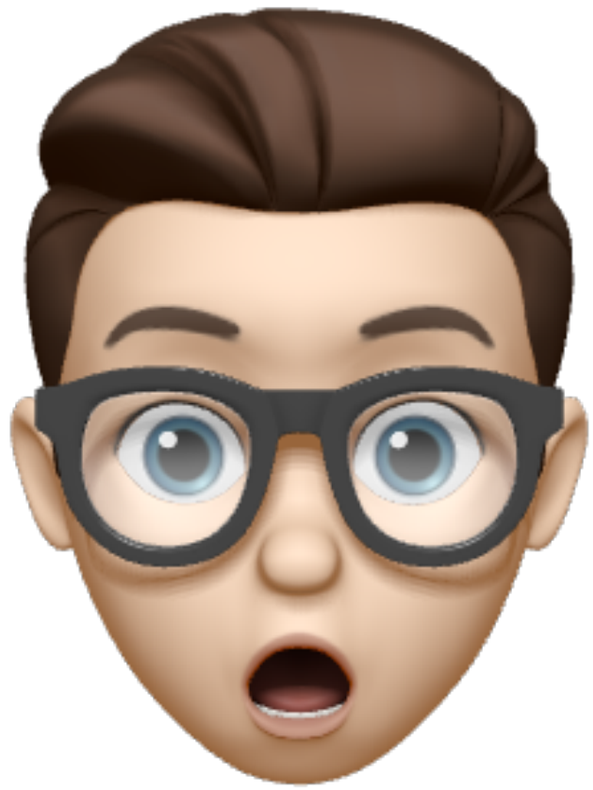
Software Developer – Freelancer

💙 php + Symfony 🖤

❤️ Open Source ❤️

🤎 Legacy Code 🤎

💕 Father of Four 💕



about:symfony

web application framework

set of components

Symfony

Symfony 5

November 2019

released live on stage
@ SymfonyCon Amsterdam

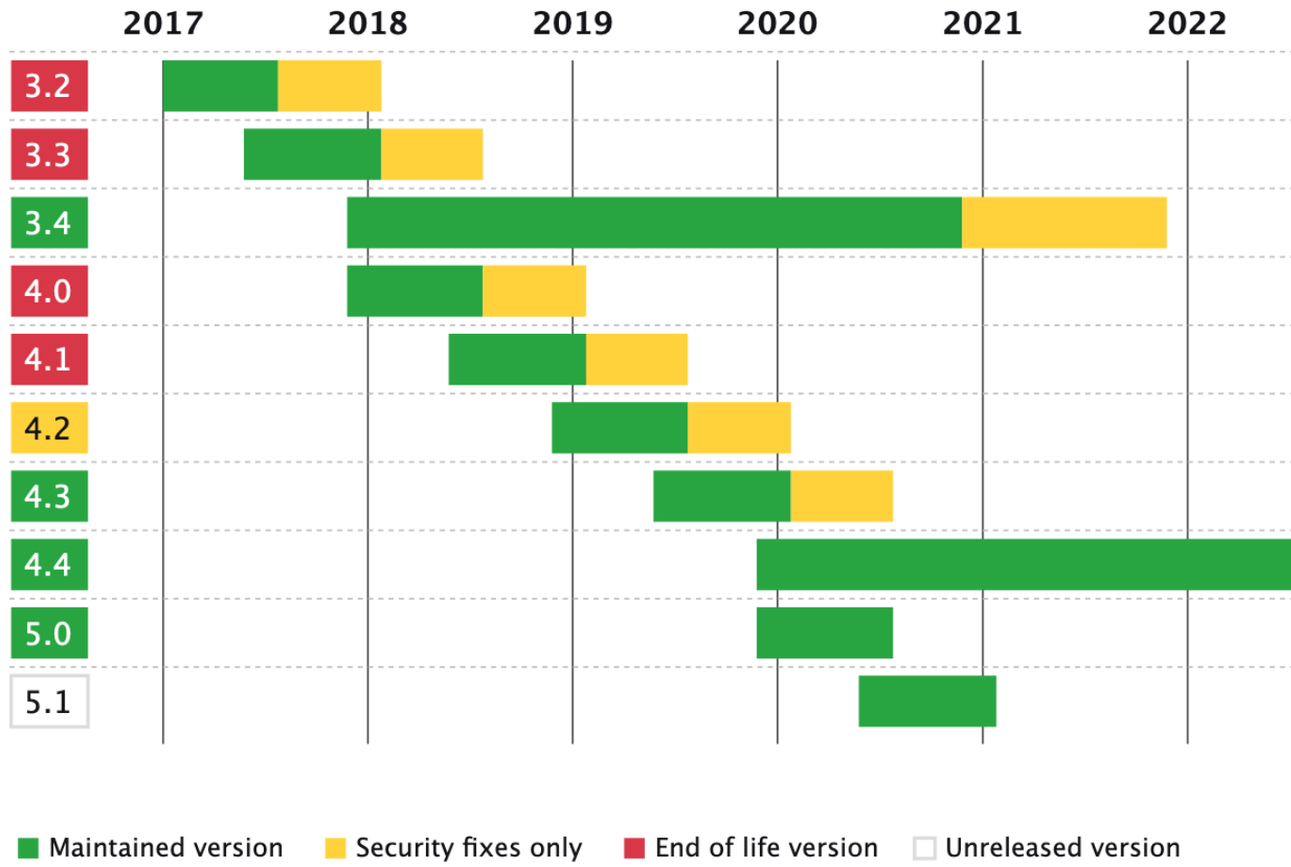
Symfony

New Features since Symfony 4.4?

*

Symfony's Release Process

- minor releases for features
- major releases for cleanup
- LTS release every two years



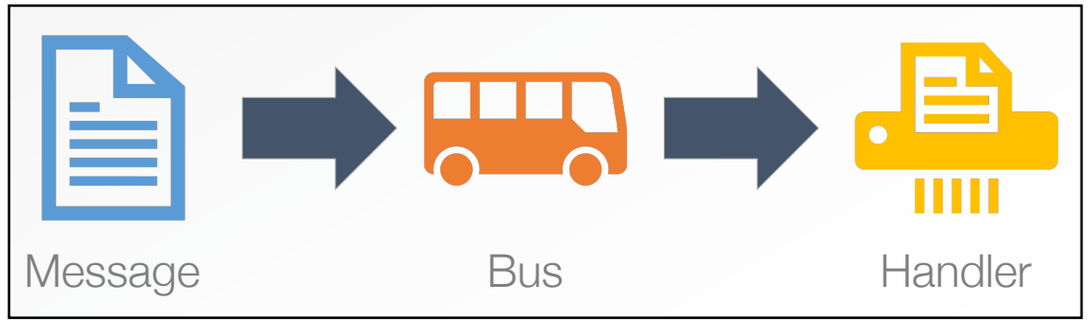
https://symfony.com/releases

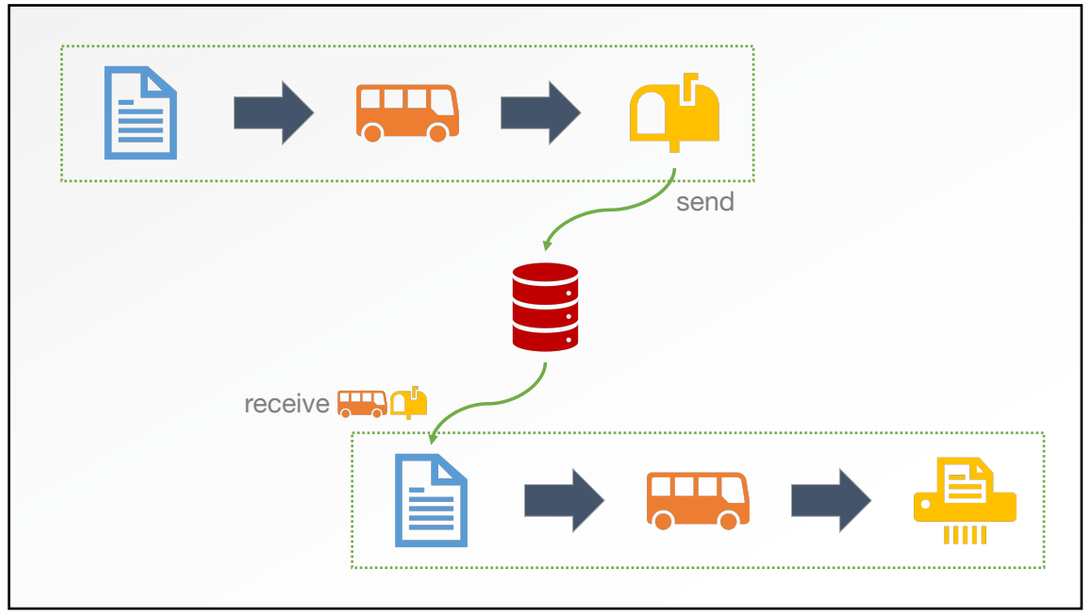
Symfony 5.0 is here,
so what happened since
the release of Symfony 4.0?

Symfony Messenger

Abstraction layer for (asynchronous) message processing

available since Symfony 4.1

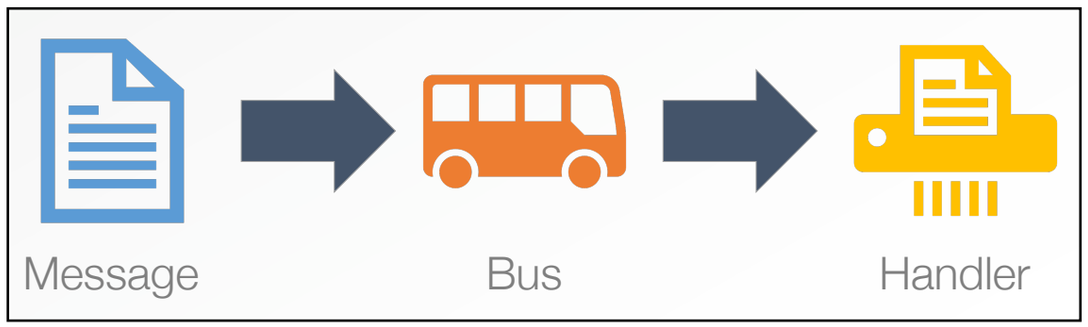




These images were shamelessly stolen from Denis Brumann:
https://speakerdeck.com/dbrumann/symfony-messenger-4-dot-3

Synchronous Processing

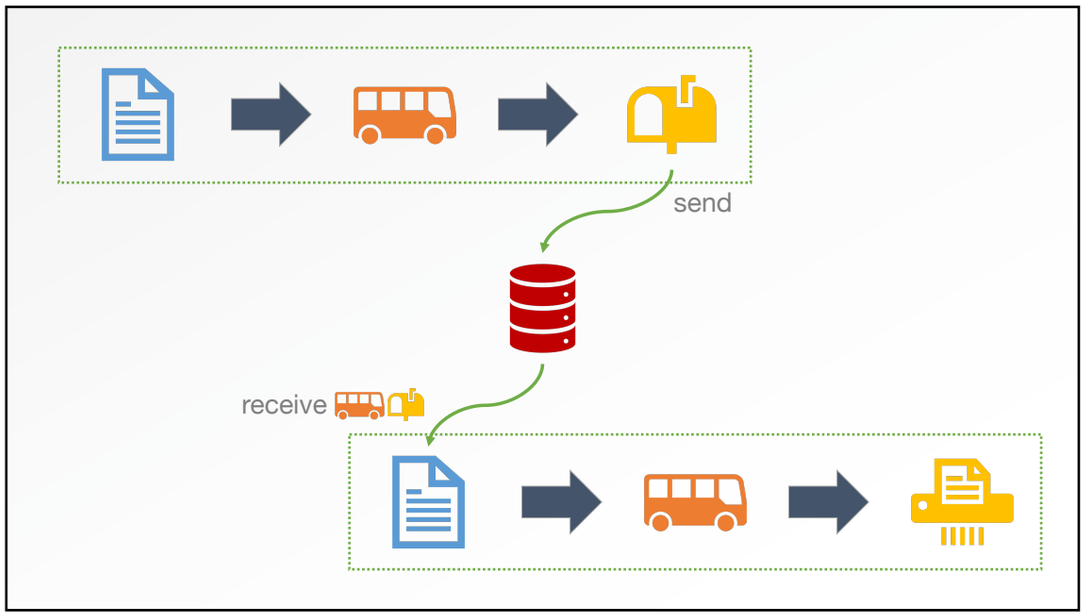
- Message producer and handler on the same process.
- low complexity
- feedback on failures
- good starting point



These images were shamelessly stolen from Denis Brumann:
https://speakerdeck.com/dbrumann/symfony-messenger-4-dot-3

Go Async: Add a Message Queue

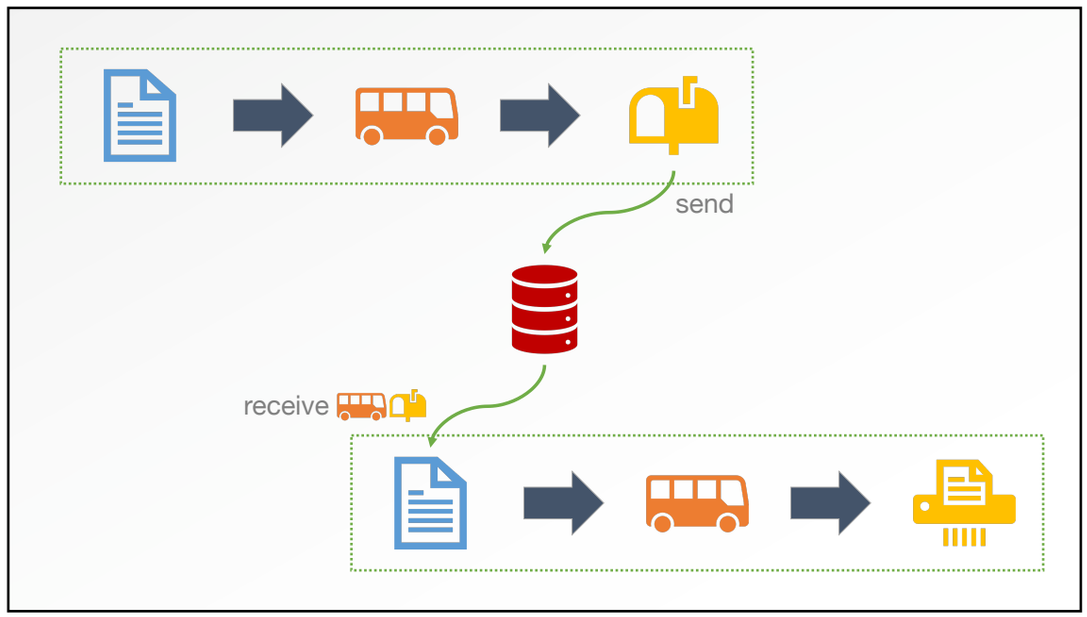
- Message is serialized to a transport storage.
- Dedicated worker processes delegate the message to its handler.



These images were shamelessly stolen from Denis Brumann:
https://speakerdeck.com/dbrumann/symfony-messenger-4-dot-3

Transport Adapters

- AMQP (e.g. RabbitMQ)
- Redis
- SQL databases (via Doctrine)
- in-memory



These images were shamelessly stolen from Denis Brumann:
https://speakerdeck.com/dbrumann/symfony-messenger-4-dot-3

Symfony WebLink

Generate `Link:` headers for asset preloading and HTTP/2 server push

available since Symfony 4.1

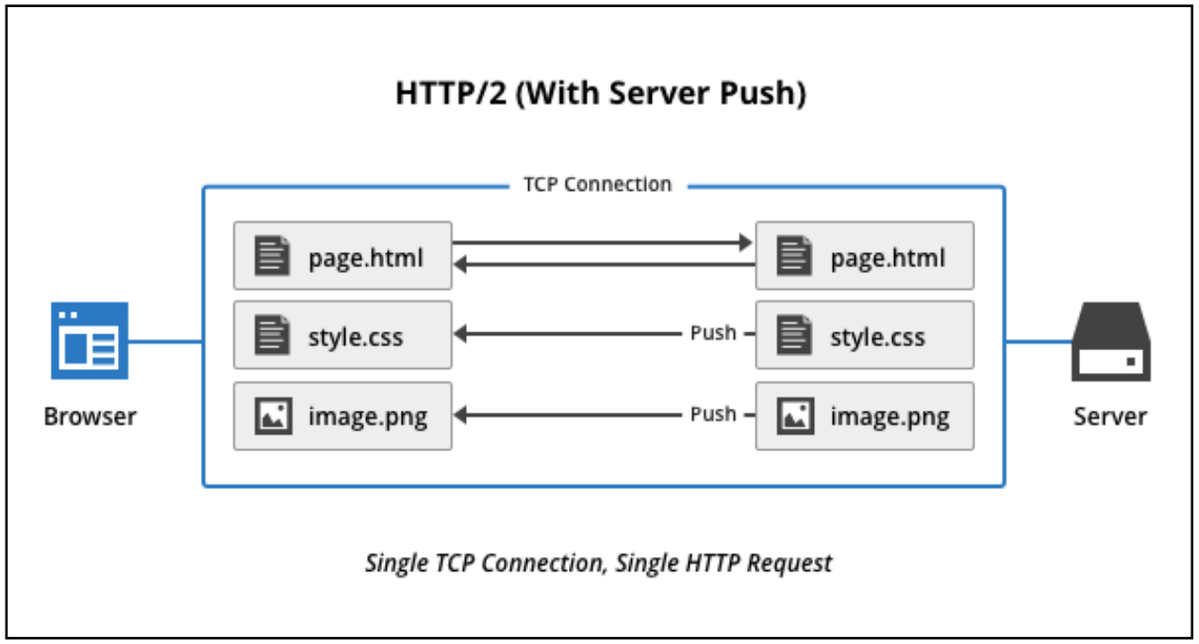


Image source:
https://blog.cloudflare.com/announcing-support-for-http-2-server-push-2/

WebLink: Twig Helper

```
<head>
    <link rel="stylesheet"
          href="{{ preload('/app.css', { as: 'style' }) }}">
</head>
```

```
Link: </app.css>; rel="preload"; as="style"
```

Symfony HttpClient

low-level asynchronous HTTP client

Available since Symfony 4.3

```php
use Symfony\Component\HttpClient\HttpClient;

$client = HttpClient::create();

$response = $client->request(
    'GET',
    'https://api.example.com/my/endpoint'
);

$statusCode = $response
    ->getStatusCode();

$contentType = $response
    ->getHeaders()['content-type'][0];

$content = $response
    ->getContent();
```

Streamed Responses

```php
$url = 'https://releases.ubuntu.com/18.04.1/ubuntu-18.04.1-desktop-amd64.iso';
$response = $client->request('GET', $url);

// Responses are lazy: this code is executed as soon as headers are received
if (200 !== $response->getStatusCode()) {
    throw new \Exception('...');
}

// get the response contents in chunk and save them in a file
// response chunks implement Symfony\Contracts\HttpClient\ChunkInterface
$fileHandler = fopen('/ubuntu.iso', 'w');
foreach ($client->stream($response) as $chunk) {
    fwrite($fileHandler, $chunk->getContent());
}
```

https://symfony.com/doc/5.0/components/http_client.html#streaming-responses

Are you still using Guzzle?

That's okay!

Symfony Mime

Compose and manipulate MIME messages.

Available since Symfony 4.3

```php
use Symfony\Component\Mime\Email;

$email = (new Email())
    ->from('fabien@symfony.com')
    ->to('foo@example.com')
    ->cc('bar@example.com')
    ->bcc('baz@example.com')
    ->replyTo('fabien@symfony.com')
    ->priority(Email::PRIORITY_HIGH)
    ->subject('Important Notification')
    ->text('Lorem ipsum...')
    ->html('<h1>Lorem ipsum</h1> <p>...</p>')
;
```

Symfony Mailer

A replacement for the retired Swiftmailer library.

Available since Symfony 4.3.

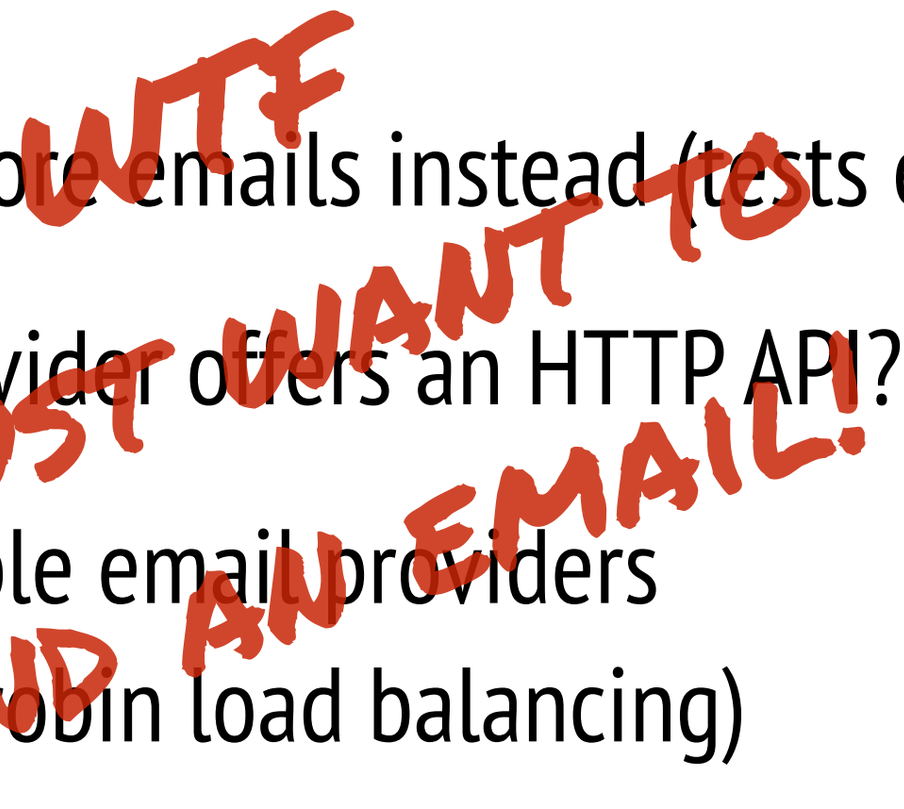
```php
use Symfony\Component\Mailer\Mailer;
use Symfony\Component\Mailer\Transport\Smtp\EsmtpTransport;

$transport = new EsmtpTransport('localhost');
$mailer = new Mailer($transport);
$mailer->send($email);
```

E-Mail = SMTP?

What if…

- … we want to store emails instead (tests etc.)?
- … our email provider offers an HTTP API?
- … we use multiple email providers (failover, round-robin load balancing)



Preconfigured Transports

Amazon SES

Mailchimp Mandrill

Google GMail

Mailgun

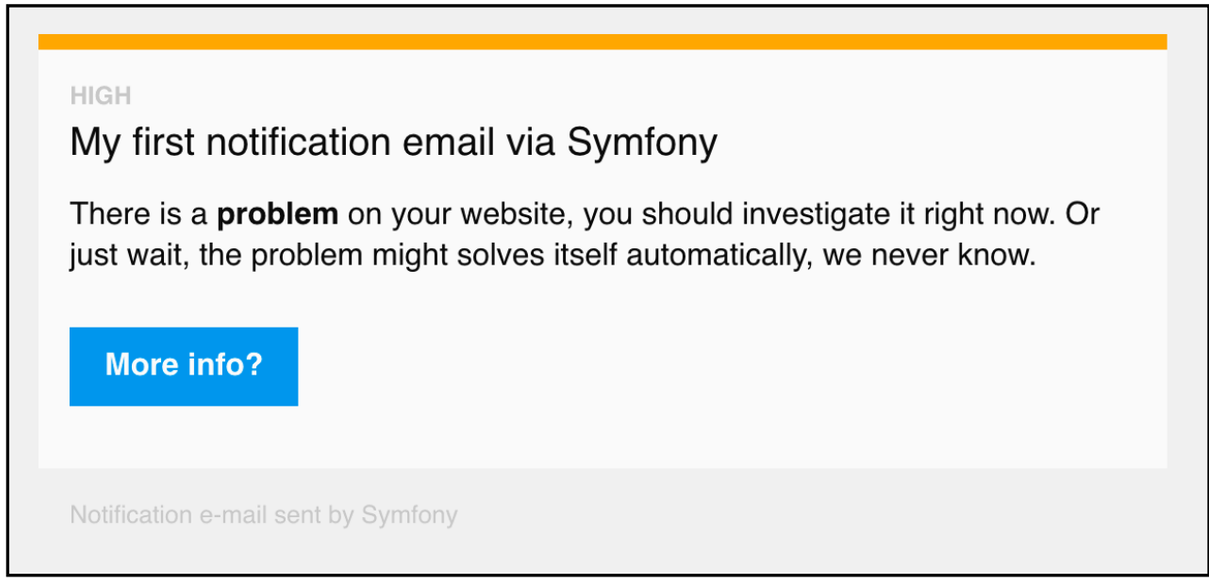
Postmark

Sendgrid

Components used by Mailer

- Mime and TwigBridge for E-Mail composition
- HttpClient for accessing provider APIs
- Messenger for asynchronous delivery

Notification Emails

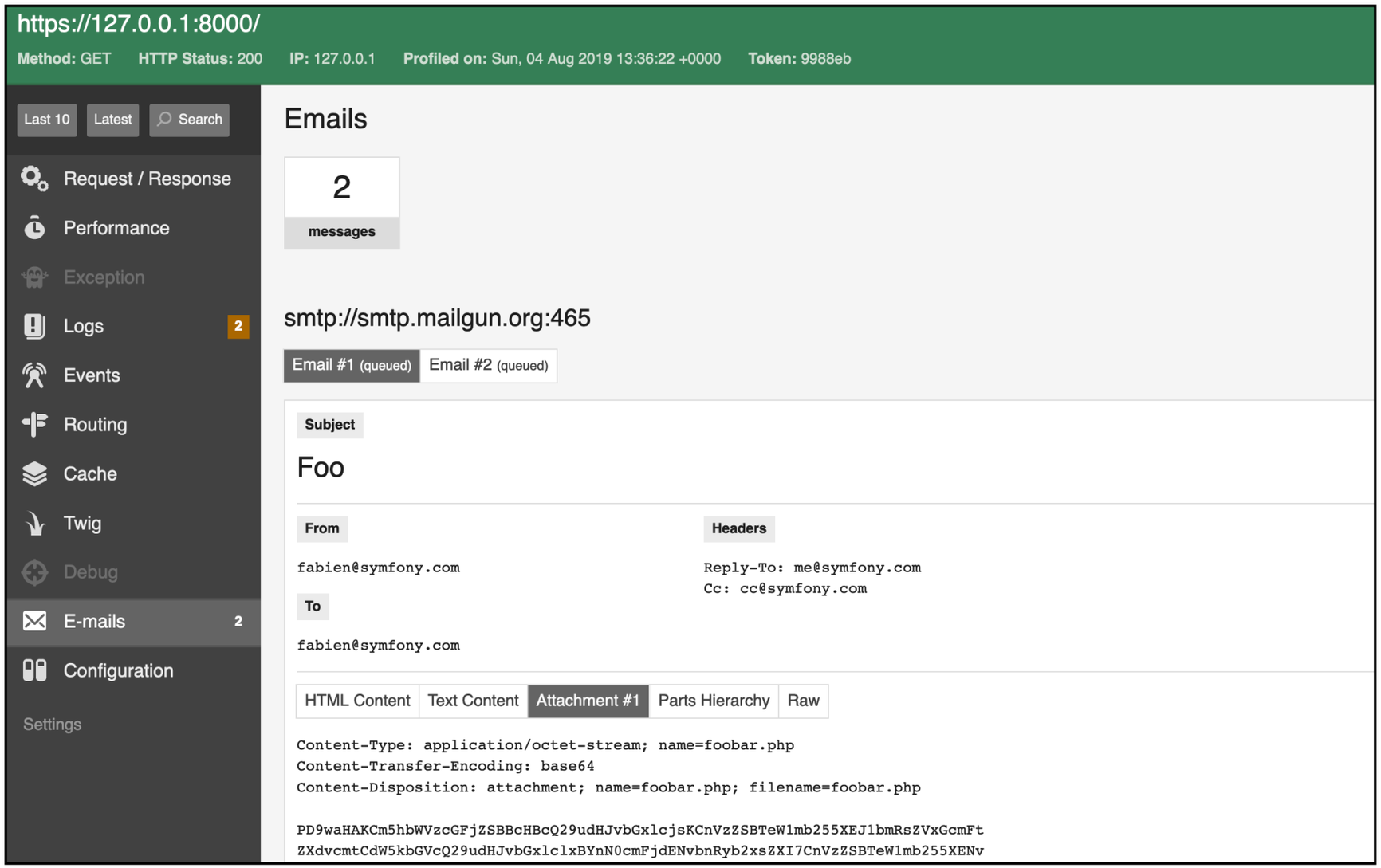



```php
use Symfony\Bridge\Twig\Mime\NotificationEmail;

$email = (new NotificationEmail())
    ->from('fabien@example.com')
    ->to('fabien@example.org')
    ->subject('My first notification email via Symfony')
    ->markdown(<<<EOF
There is a **problem** on your website, you should investigate it right now.
Or just wait, the problem might solves itself automatically, we never know.
EOF
    )
    ->action('More info?', 'https://example.com/')
    ->importance('high')
;
```

https://github.com/symfony/symfony/pull/33605

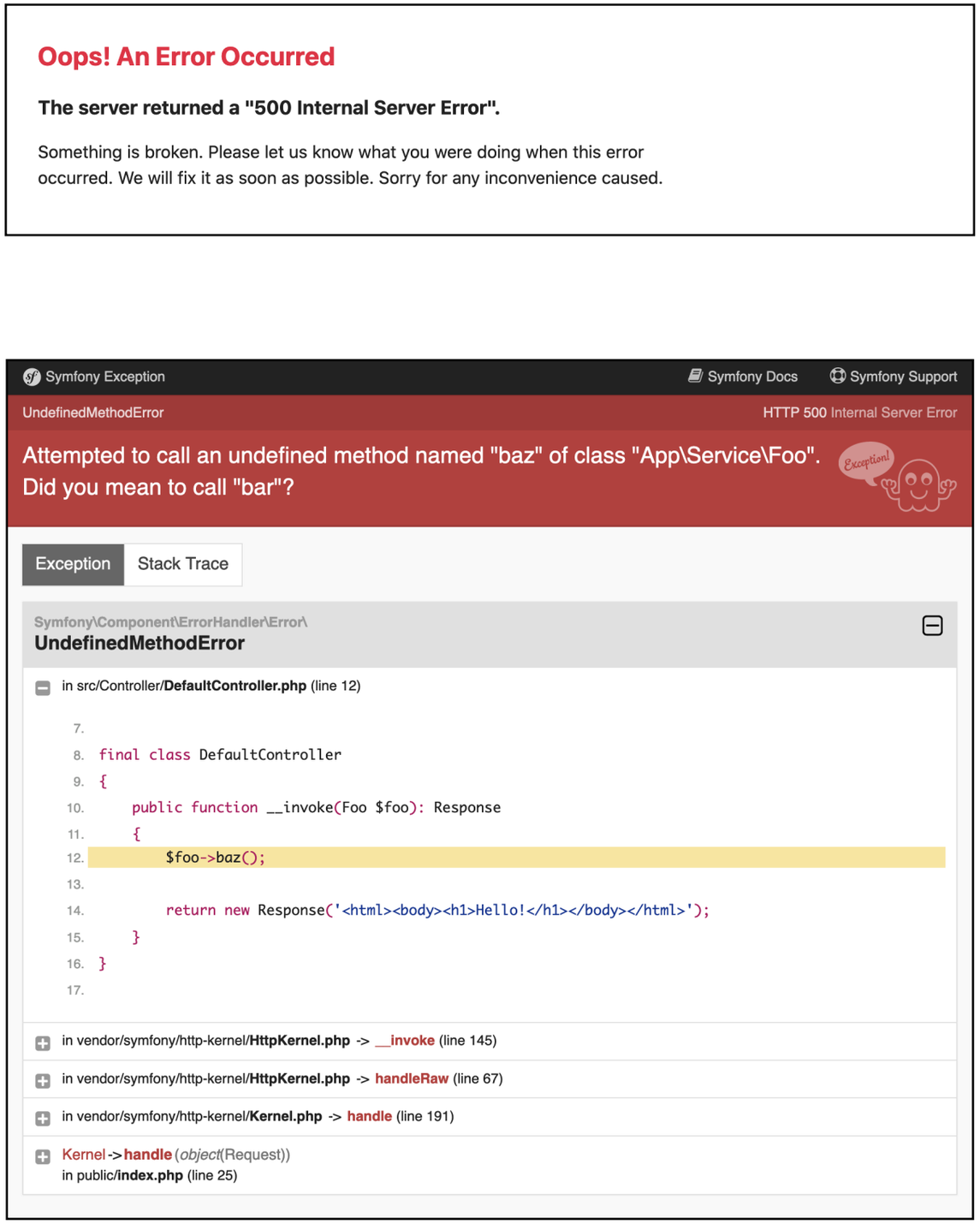
Profiler Integration



https://github.com/symfony/symfony/pull/32912

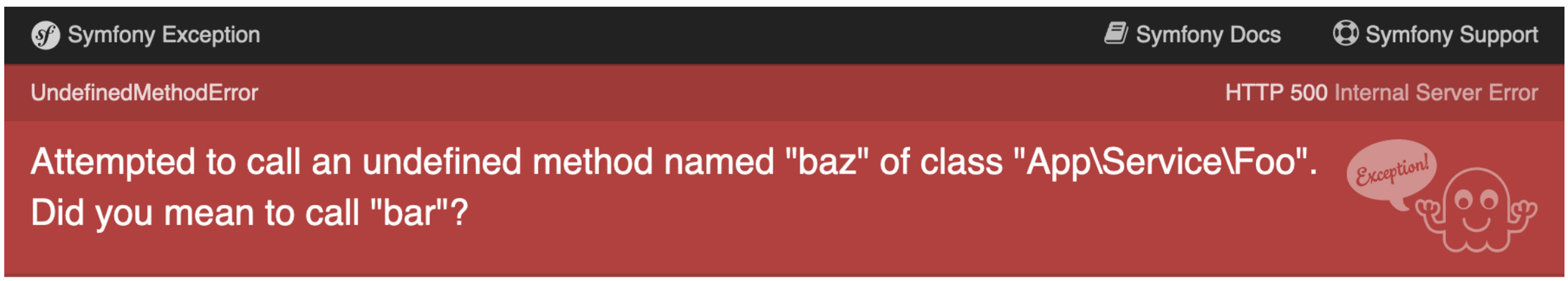
Symfony ErrorHandler

Replacement for the Debug component.



Error Enhancers

Make php's error messages more helpful in dev mode.



OPCache Preloading

New feature in PHP 7.4:
Pre-load classes and functions before processing a request.

```
; php.ini
opcache.preload=/path/to/project/var/cache/prod/App_KernelProdContainer.preload.php
```

https://symfony.com/blog/new-in-symfony-4-4-preloading-symfony-applications-in-php-7-4

Password Migrations

Migrate passwords on login when stronger hashes become available.

```yaml
# config/packages/security.yaml
security:
    encoders:
        legacy:
            algorithm: sha256
            encode_as_base64: false
            iterations: 1

        App\Entity\User:
            # the new encoder, along with its options
            algorithm: sodium
            migrate_from:
                - bcrypt # uses the "bcrypt" encoder with the default options
                - legacy # uses the "legacy" encoder configured above
```

lint:container

Find problems in your service configurations by running a console command.

```php
namespace App\SomeNamespace;

class SomeService
{
    public function __construct(int $someProperty = 7)
    {
        // ...
    }
}
```

```yaml
services:
    App\SomeNamespace\SomeService: ~
```

```
Invalid definition for service "App\SomeNamespace\SomeService":
argument 1 of "App\SomeNamespace\SomeService::__construct" accepts
"int", "NULL" passed.
```

Type Declarations

Many Symfony classes and interfaces have been updated with scalar type declarations.

```php
interface HttpKernelInterface
{
    const MASTER_REQUEST = 1;
    const SUB_REQUEST = 2;

    public function handle(
        Request $request,
        $type = self::MASTER_REQUEST,
        $catch = true
    );
}
```

4.4

```php
interface HttpKernelInterface
{
    const MASTER_REQUEST = 1;
    const SUB_REQUEST = 2;

    public function handle(
        Request $request,
        int $type = self::MASTER_REQUEST,
        bool $catch = true
    );
}
```

5.0

Type Declarations

- Goal:
  Userland code should not break.
- php 7.2 allows
  parameter type widening
- Return type declarations:
  planned for Symfony 6.

```php
interface TranslatorInterface
{
    public function trans(
        $id,
        array $parameters = [],
        $domain = null,
        $locale = null
    );
}
```

4.4

```php
interface TranslatorInterface
{
    public function trans(
        string $id,
        array $parameters = [],
        string $domain = null,
        string $locale = null
    );
}
```

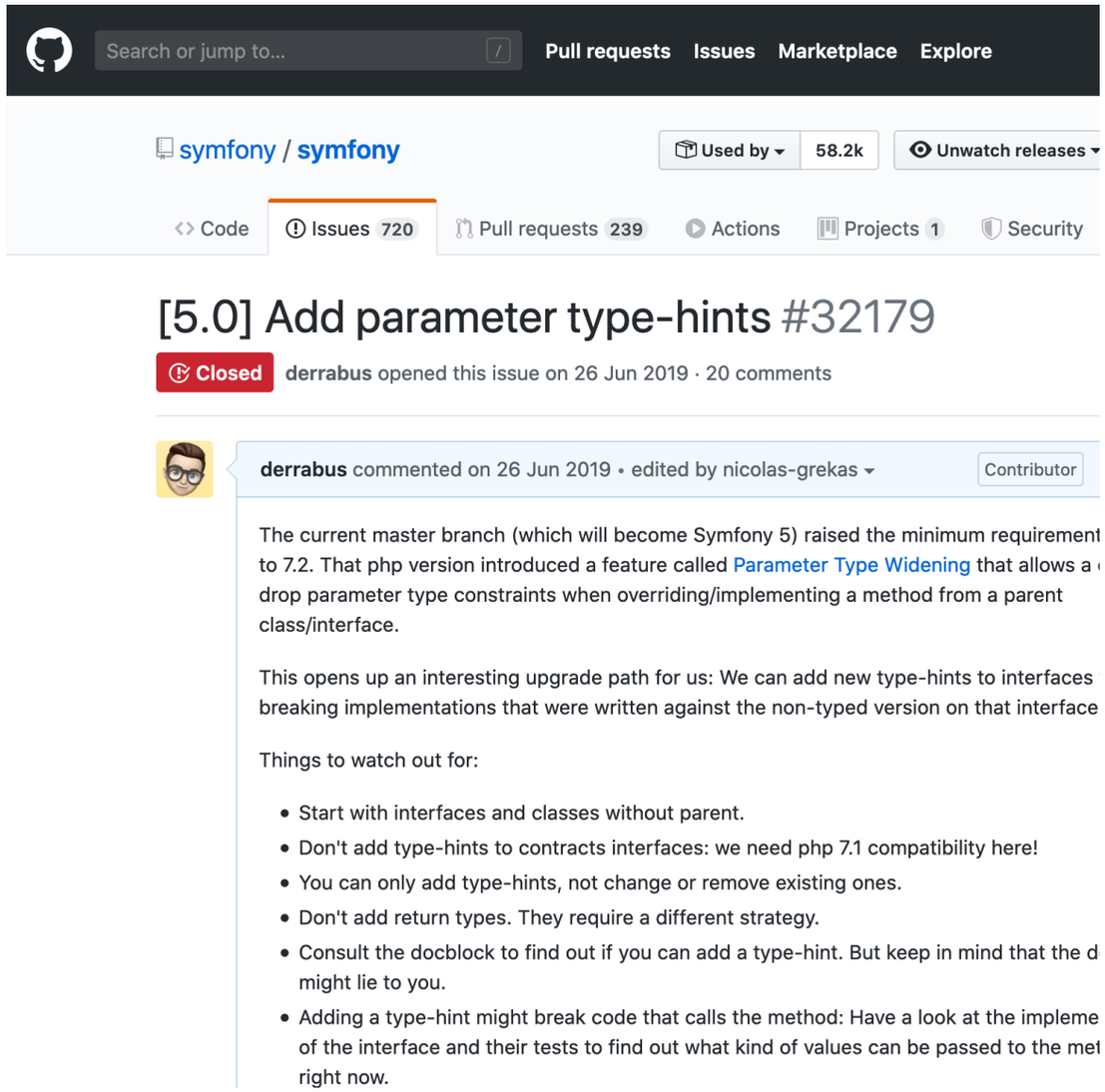
5.0

Type Declarations: What we've found

- bugs
- broken/incomplete mocks in unit tests
- lying PHPDoc blocks
- strange execution paths
- Working on type declarations for Symfony 5 improved the codebase of Symfony 3.4.

Community Effort

- Work was split by component and tracked in a GitHub issue.
- More than 10 developers worked on it.



https://github.com/symfony/symfony/issues/32179

Symfony String

Consistent object-oriented interface for string manipulation.

experimental in Symfony 5.0

```php
use function Symfony\Component\String\u;

$text = u('This is a déjà-vu situation.')
    ->trimEnd('.')
    ->replace('déjà-vu', 'jamais-vu')
    ->append('!');

// $text = 'This is a jamais-vu situation!'
```

https://symfony.com/blog/new-in-symfony-5-0-string-component

Symfony Notifier

Send messages via a unified API.

experimental in Symfony 5.0

Symfony 4 -› 5

- New stable components:
  Messenger, WebLink, HttpClient, Mime, Mailer
- Experimental components:
  String, Notifier
- Improvements to existing components.
- A better php 7 experience through stricter typing.

Thank You!

https://twitter.com/derrabus
https://github.com/derrabus
https://facebook.com/derrabus