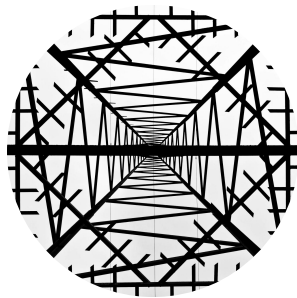


Les événements – utilisation dans Symfony



Meetup symfony montpellier
30 janvier 2017 - 1001 Pharmacies

Bonjour et désolé!



Julien Vinber

Développeur Application et Web
pour l'éditeur de logiciel Yooda

The background is white and decorated with various colorful circles and dashed lines. In the top left, there is a large orange circle with a dashed red outline, overlapping a yellow circle. Below them is a small pink circle. In the top right, there is a green circle with a white dot in the center, a small orange circle, and a yellow circle with a dashed green outline. In the bottom left, there is a green circle with a dashed green outline, a large yellow circle, and a small cyan circle. In the bottom right, there is a large cyan circle with a white dot in the center, and a cyan circle with a dashed blue outline. In the center, there is a large dashed blue circle containing the number 1.

1

Programmation séquentielle

The background features a large, faint dashed circle. Scattered around it are various solid-colored circles and arcs in shades of green, yellow, orange, red, and blue. Some circles have smaller circles inside them, creating a nested effect. The overall aesthetic is modern and abstract.

“

la programmation séquentielle est un paradigme de programmation dans laquelle le déroulement des instructions du programme est toujours le même (les instructions elles-mêmes peuvent être différentes en fonctions des embranchements,...).

Wikipedia

Simple ?


$$1+1=3$$



L'homme

Plus on est libre, plus les
choix à faire sont complexes

Sauve-moi - **Guillaume Musso**

Quelle solution ?

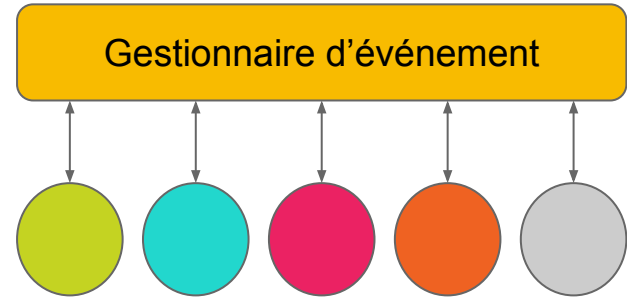
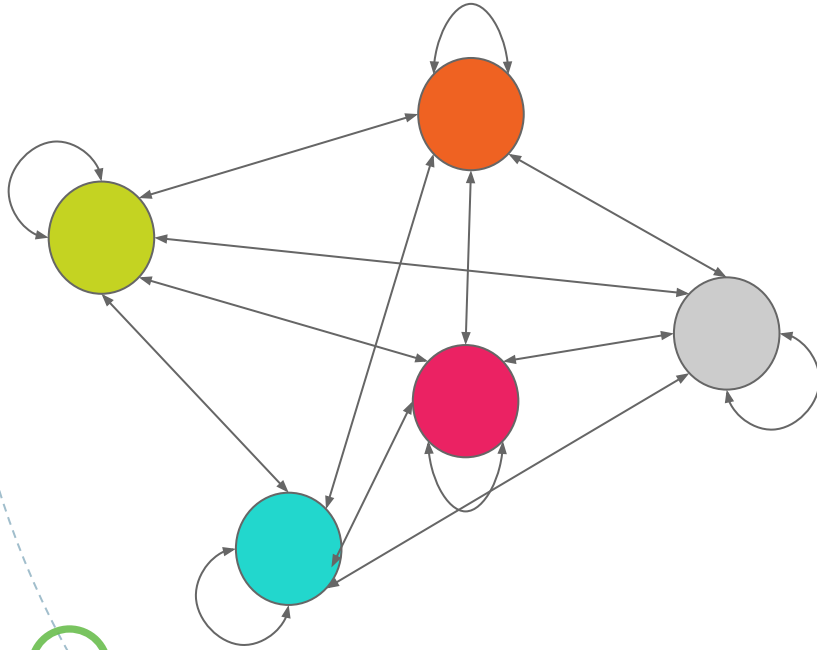


The background is white and decorated with various colorful circles and dashed lines. In the top left, there is a large orange circle with a dashed red outline, overlapping a yellow circle. Below them is a small pink circle. In the top right, there is a green circle with a white dot in the center, a small orange circle, and a lime green circle with a dashed yellow outline. In the bottom left, there is a green circle with a dashed green outline, a large lime green circle, and a small cyan circle. In the bottom right, there is a large cyan circle with a white dot in the center, and a small cyan circle with a dashed blue outline.

2

Programmation
événementielle

On ne cherche plus à complexifier du code pour gérer les interactions. Au contraire on le découpe en petite action simple et autonome.





Delphi

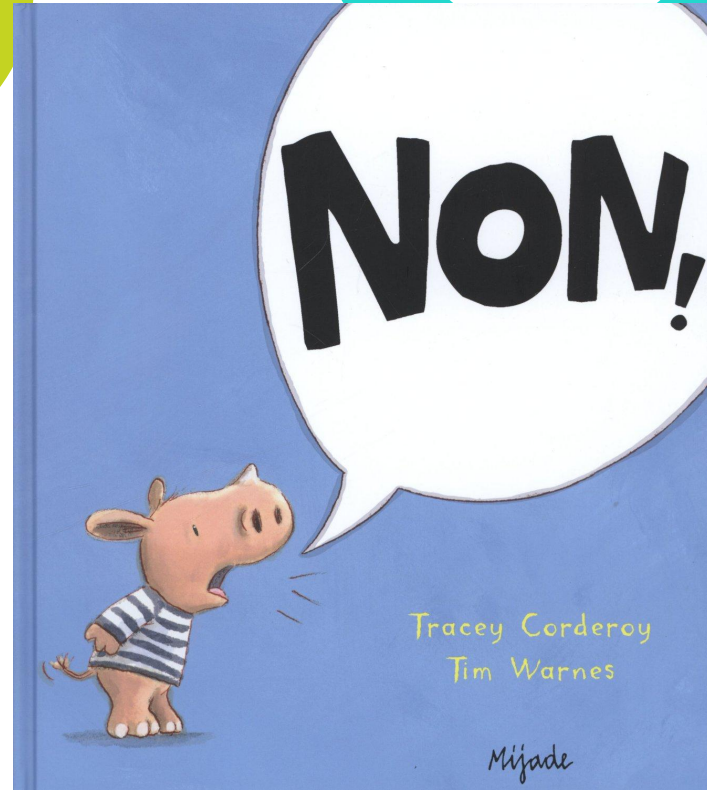


Objective-C

The background is white and decorated with various colorful circles and shapes. In the top left, there is a large orange circle with a dashed red outline, overlapping a yellow circle. Below them is a small pink circle. In the top right, there is a green circle with a white dot in the center, a small orange circle, and a yellow circle with a dashed green outline. In the bottom left, there is a green circle with a dashed green outline, a large yellow circle, and a small cyan circle. In the bottom right, there is a large cyan circle with a white dot in the center, and a cyan circle with a dashed blue outline. A large, light blue dashed circle is centered in the upper half of the image, containing the number 3.

3

PHP et les événements ?



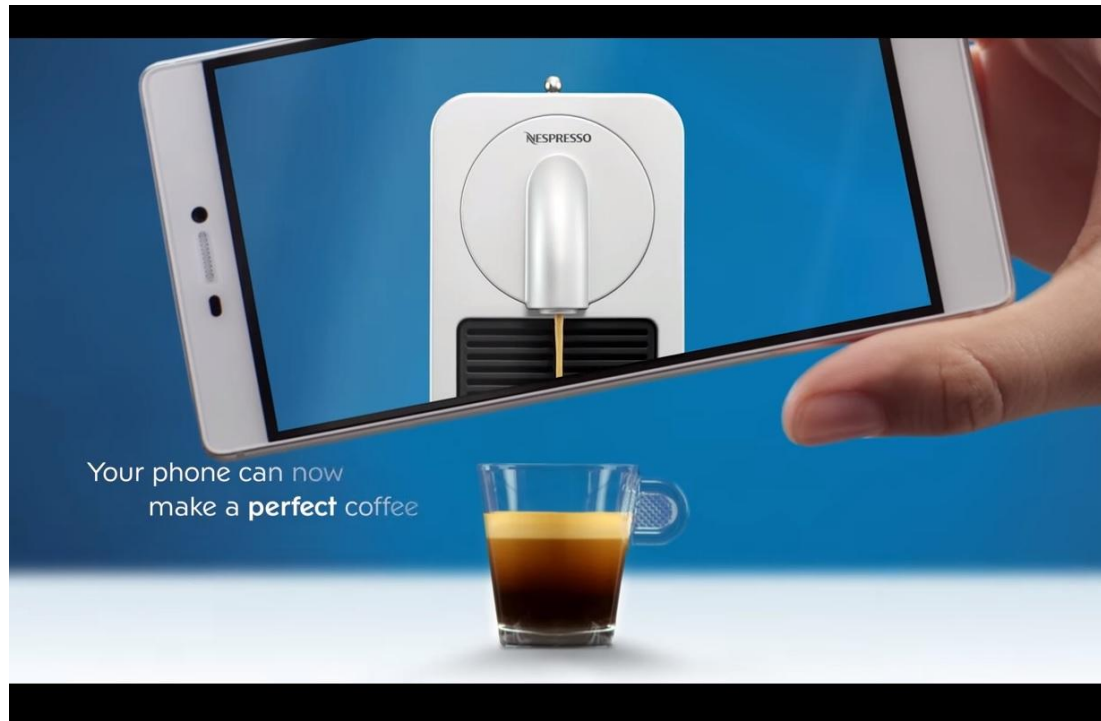
The slide features a decorative graphic consisting of various colored circles and rings. In the top left, there are pink, orange, and blue shapes. In the top right, there are green, blue, and yellow shapes. On the left side, there is a large green ring and a smaller blue circle. In the bottom left, there is a large green ring and a smaller orange circle. In the bottom right, there is a small green circle. The background is white.

Oui, mais voilà, les applications se complexifient

Par exemple lors de la création d'un utilisateur on peut vouloir :

- Envoyer un mail de confirmation
- Créer un utilisateur dans la gestion Co.
- Envoyer une alerte aux commerciales pour contacter le client.
- Lui envoyer dans 7 jours une offre promo.
- Valider son mail.
- Vérifier s'il y a un parrain et lui envoyer une récompense.
- ...

Résultat, un code qui fait tout et n'importe quoi.



The background is white and decorated with various colorful circles and shapes. In the top left, there is a large orange circle with a dashed red outline, overlapping a yellow circle. Below them is a small pink circle. In the top center, a large blue number '4' is centered within a large, light blue dashed circle. In the top right, there is a green circle with a white dot in the center, a small orange circle, and a yellow circle with a dashed green outline. In the bottom left, there is a green circle with a dashed green outline, a small cyan circle, and a large yellow circle. In the bottom right, there is a large blue circle with a white dot in the center, a small cyan circle with a dashed blue outline, and a small cyan circle.

4

Les événements avec
Symfony.

A decorative graphic consisting of various colored circles and rings. In the top left, there are overlapping pink, orange, and blue circles, with a dashed pink circle around the pink one. Below these is a solid blue ring and a small solid blue circle. In the top right, there are green, blue, and yellow circles, with a dashed yellow circle around a yellow one. In the bottom left, there is a large green ring with a dashed green outline, a solid green circle, and a small orange ring. In the bottom right, there is a small solid green circle.

Principe

Exemple lors de la création d'un utilisateur :

- On crée l'utilisateur en base
- Déclenchement d'un événement
- Fin.

On crée un écouteur pour l'envoi des mails

- On enregistre l'écouteur
- Je suis appelé
- J'envoie mon mail
- Fin
- ...



Avantage

- Enlève les dépendances
- Découpage en action simple.
- Test simplifier.
- On revient sur un découpage métier.

Inconvénient

- Plus de code dans l'absolu.
- Perte de linéarité du code.
- Plus difficile à débbugger

Particularité

N'étant pas géré en natif. Le code reste mono thread et synchrone.



5

Exemple de code :



5.1



Créer un événement



Créer l'objet événement

```
class UserEvent extends Event
{
    const EVENT_USERCREATE = 'user.create';
    protected $user;

    public function __construct (User $user)
    {
        $this->user = $user;
    }

    public function getUser ()
    {
        return $this->user;
    }
}
```

Déclenchement

```
$user = new User();
```

```
$user->setEmail('julien@vinber.fr');
```

```
$event = new UserEvent($user);
```

```
$this->get('event_dispatcher')->dispatch(UserEvent::EVENT_US  
ERCREATE, $event);
```



5.2



Répondre à un événement.





Le code qui est appelé.

```
class EnvoyerMailConfirmation
{
    public function onUserCreate (UserEvent $event)
    {
        $user = $event->getUser ();
        dump ($user);
    }
}
```



Enregistrement

app.envoyermail.user.create:

```
class: App\AppBundle\Event\EnvoyerMailConfirmation
```

```
tags:
```

```
- { name: kernel.event_listener, event: user.create }
```


The background features several decorative elements: a large dashed light blue circle in the upper center; a thick orange ring in the top left; a yellow circle below it; a small pink dot below the yellow circle; a green circle with a dashed outline in the bottom left; a large green circle in the top right; a small yellow dot above it; a lime green circle with a dashed outline below the yellow dot; a large teal ring in the bottom right; a small teal circle with a dashed outline below the teal ring; and a large lime green circle in the bottom left.

6

Événement existant

http://obim.lc/app_dev.php/

Method: GET HTTP Status: 200 IP: 127.0.0.1 Profiled on: Mon, 30 Jan 2017 13:11:42 +0100 Token: 00345e

Last 10 Latest Search

Request / Response

Performance

Forms

Exception

Logs 23

Events

Routing

Translation 19

Security

Twig

Doctrine

E-Mails

Debug

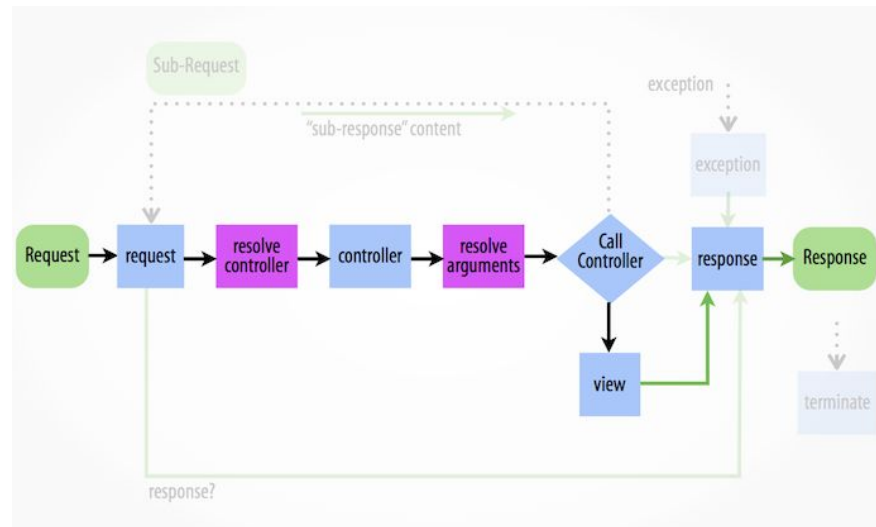
Configuration

Event Dispatcher

Called Listeners 32 Not Called Listeners 28

Priority	Listener
kernel.request	
2048	"Symfony\Component\HttpKernel\EventListener\DebugHandlersListener::configure() "
256	"Symfony\Component\HttpKernel\EventListener\ValidateRequestListener::onKernelRequest() "
128	"Symfony\Bundle\FrameworkBundle\EventListener\SessionListener::onKernelRequest() "
48	"Symfony\Component\HttpKernel\EventListener\FragmentListener::onKernelRequest() "
32	"Symfony\Component\HttpKernel\EventListener\RouterListener::onKernelRequest() "
16	"Symfony\Component\HttpKernel\EventListener\LocaleListener::onKernelRequest() "
10	"Symfony\Component\HttpKernel\EventListener\TranslatorListener::onKernelRequest() "
8	"Symfony\Component\Security\Http\Firewall::onKernelRequest() "
0	"Knp\Bundle\MenuBundle\EventListener\VoterInitializerListener::onKernelRequest() "
0	"Knp\Bundle\PaginatorBundle\Subscriber\SlidingPaginationSubscriber::onKernelRequest() "
0	"Symfony\Bundle\AsseticBundle\EventListener\RequestListener::onKernelRequest() "
kernel.controller	
0	"Symfony\Bundle\FrameworkBundle\DataCollector\RouterDataCollector::onKernelController() "
0	"Symfony\Bundle\FrameworkBundle\DataCollector\RequestDataCollector::onKernelController() "

Kernel.request
Kernel.controller
Kernel.view
Kernel.response
Kernel.terminate
Kernel.exception



Merci!



Des questions?

julien@vinber.fr