



TYPO3

TUGA 89 @ FONDA



Danke an **FONDA** !



Agenda

- Vorstellung Fonda
- Vorstellung Teilnehmer:innen
- News
- Middleware an einem praktischen Beispiel (Georg Ringer)
- Events & EventListener und warum das so cool ist (Georg Ringer)
- TCA Migration mit Rector (Marianne Moser)
- Vorstellung von captcha.eu (Helmut Januschka)
- Erfahrungsaustausch TYPO3 v12



News aus der TYPO3 Welt

- **Government Site Builder GSB v11**

Deutscher Bund investiert € 110 Mio. in Migration auf TYPO3 nächsten Jahre

https://produkt.gsb.bund.de/Produkt_neu/GSB11/gsb11_node.html

- **EXT:content_blocks**

<https://github.com/nhovratov/content-blocks>

- **TUGA**

- 90 in Linz

- 91 in Wien => Xmas-TUGA?



Middleware in TYPO3

- PSR-15
- Standard, Standard, Standards
- Seit TYPO3 v9
- stetig weiter genutzt im Core



Anforderung

- Kunde benötigt ein Logging von Aufrufen
subdomain.domain.at/<irgendein-parameter>
 - durch Aufruf soll ein täglich rotierendes CSV erzeugt werden
mit IP, Datum, Parameter, User-Agent
- subdomain.domain.at hat keinen Content
- Kunde hat aber schon eine TYPO3 Site
- Nicht-Anforderung
 - High-Performance/Load



mögliche Optionen

1. Eigenständige Mini-Applikation mit PHP, Node, ...
 - Was nimmt man?
 - Wer betreut es?
 - Deployment, ...
2. TYPO3 mit eigenem Seitenbaum?
 - Wozu? Gibt keinen Inhalt
3. TYPO3 mit Middleware



Umsetzung

Configuration/RequestMiddlewares.php

```
return [  
    'frontend' => [  
        'typo3/theme/logger' => [  
            'target' => \StudioMitte\Theme\Middleware\ LinkLogger::class,  
            'before' => [  
                'typo3/cms-frontend/timetracker' ,  
            ],  
            'after' => [  
            ],  
        ],  
    ],  
];
```



Umsetzung

```
public function process(  
    ServerRequestInterface $request,  
    RequestHandlerInterface $handler): ResponseInterface  
{  
    if ($this->isValidLoggerPage($request)) {  
        $this->track($request);  
  
        return new RedirectResponse('https://demo-12.ddev.site/danke');  
    }  
    return $handler->handle($request);  
}
```



Demo



Event & EventListener



Vergangenheit: Hooks

```
$GLOBALS [ 'TYPO3_CONF_VARS' ]  
    [ 'SC_OPTIONS' ] [ 't3lib/class.t3lib_tcemain.php' ]  
    [ 'processDatamapClass' ] []  
        = \In2code\Powermail\Hook\CreateMarker::class;
```

- Deprecation nicht möglich
- \$GLOBALS
- keine Verbindlichkeit



Vergangenheit: SignalSlots

```
$signalSlotDispatcher = GeneralUtility::makeInstance(SignalSlot\Dispatcher::class);  
$signalSlotDispatcher->connect(  
    \TYPO3\CMS\Extensionmanager\Utility\InstallUtility::class,  
    'afterExtensionInstall',  
    \T3Monitor\T3monitoringClient\Slots\ExtensionManagerSlot::class,  
    'afterExtensionInstall'  
);
```

- besser aber noch nicht gut genug
- NIH (Not invented here)



Events & EventListeners

- PSR-14
- Standards, Standards, Standards

```
StudioMitte\Theme\EventListener\ControllerShowEventListener
```

```
tags:
```

- name: event.listener
- identifier: 'ext-theme/controller-show'
- event: FriendsOfTYPO3\TtAddress\Events\ControllerShowEvent



<Some> 3rd Party Code

```
public function showAction (Address $address = null)
{
    $event = new ControllerShowEvent ($address, $this->settings);
    $this->eventDispatcher->dispatch ($event);

    $this->view->assignMultiple ([
        'address' => $address,
    ]);
    return $this->htmlResponse ();
}
```



Event - DTO

```
<?php
declare(strict_types=1);

namespace FriendsOfTYPO3\TtAddress\Events;
use FriendsOfTYPO3\TtAddress\Domain\Model\Address;

final class ControllerShowEvent
{
    public function __construct(
        public readonly Address $address,
        public readonly array $settings
    )
    {
    }
}
```



EventListener

```
class ControllerShowEventListener
{
    public function __invoke(ControllerShowEvent $event): void
    {
        $fluidMail = GeneralUtility::makeInstance(FluidEmail::class);
        $fluidMail->subject('Test: ' . $event->address->getFirstName());
        $fluidMail->to('tuga@example.com');
        $fluidMail->from('tuga@example.com');

        $event->address->setBuilding('abc');
        $mailer = GeneralUtility::makeInstance(Mailer::class);
        $mailer->send($fluidMail);
    }
}
```



Vorteile/Features

- wenig Boilerplate
 - klare Definition der Eigenschaften
 - standardisierte Konfiguration
-
- Auch gut nutzbar zum Erweitern eigener Extensions



Demo



TYPO3 12 - Updates & Herausforderungen

Breaking Code Changes

- `defined(TYPO3_MODE)` or `die()`;
- Backend Module
- `f:form.*` / `f:link.action` ohne Extbase Context
- Signals/Hooks
- `Services.yaml`
- `/public/_assets/`
- Ckeditor5
- JavaScript modules
- Commands in `Commands.php`
- `switchableControllerActions`

Features

- CSP (Content Security Policy)
-



Mögliche Vorarbeiten in v10/v11

Projektspezifisch

- Ordnerstruktur mit lokalen packages
- Events nutzen wo schon möglich
- Commands in Services.yaml
- switchableControllerActions migrieren

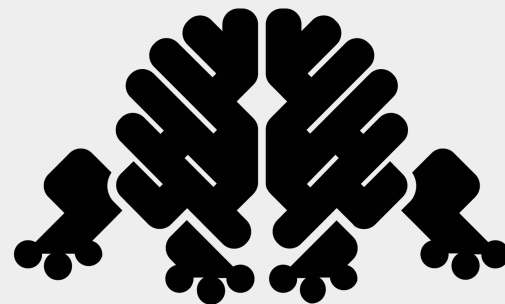
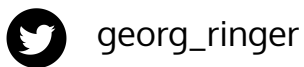
Allgemein

- PHP-CS-Fixer
- Rector



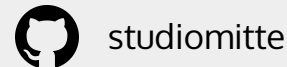
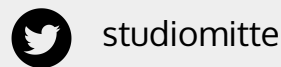
Georg Ringer

- Core Developer
- Extension Developer



STUDIO MITTE

studiomitte.com



Q & A

