

# TYPO3 NEUE APIs

Oliver Bartsch



# AGENDA

- Wer ist das?
- API - Was ist das?
- Fakten!
- Migration / Integration?!
- Sehr nett - Was kommt als nächstes?
- Ich habe eine Idee / Frage!

*WER IST DAS?*

# OLIVER BARTSCH

- Arbeitet @ b13
- Core-Entwicklung & Kundenprojekte
- TYPO3 Core Team seit 2021
- TYPO3 Ecosystem & Extensions



@oli



@\_obartsch





# API - WAS IST DAS?

*ENABLES SOFTWARE TO COMMUNICATE  
WITH EACH OTHER USING A SET OF  
DEFINITIONS AND PROTOCOLS.*

*API* ♥ TYPO3

*ENABLES YOU TO EXTEND AND CUSTOMIZE  
TYPO3 TO YOUR CUSTOM(ER) NEEDS.*



**FAKTEN**

215  
**FEATURES**

A close-up photograph of a large pile of colorful gumballs. The gumballs are scattered across the frame in various colors: red, yellow, green, and orange. They are glossy and appear to be made of a hard candy or plastic material. The lighting is bright, reflecting off the surfaces of the gumballs.

**HIDDEN  
GEMS**

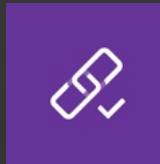
DON'T GET HOOKED  
LISTEN TO EVENTS

> 69  
**PSR-14 EVENTS**

ModifyPageLinkConfigurationEvent	AfterTemplatesHaveBeenDeterminedEvent	AfterRecordSummaryForLocalizationEvent
FilterMenuItemsEvent	ModifyEditFormUserAccessEvent	CustomFileControlsEvent
RecordAccessGrantedEvent	BeforeRecordLanguageOverlayEvent	ModifyFileReferenceControlsEvent
ModifyInfoModuleContentEvent	AfterRecordLanguageOverlayEvent	EnrichPasswordValidationContextDataEvent
ModifyGenericBackendMessagesEvent	ModifyFileReferenceEnabledControlsEvent	BeforeUserLogoutEvent
ModifyCacheLifetimeForPageEvent	EvaluateModifierFunctionEvent	AfterUserLoggedOutEvent
ModifyLinkExplanationEvent	ModifyQueryForLiveSearchEvent	AfterUserLoggedInEvent
ModifyInlineElementEnabledControlsEvent	BeforeMailerSentMessageEvent	AfterDefaultUploadFolderWasResolvedEvent
ModifyInlineElementControlsEvent	AfterMailerSentMessageEvent	ModifyRedirectManagementControllerViewDataEvent
ModifyAutoCreateRedirectRecordBeforePersistingEvent	RedirectWasHitEvent	BeforeRedirectMatchDomainEvent
ModifyNewContentElementWizardItemsEvent	ModifyPageLayoutContentEvent	AfterAutoCreateRedirectHasBeenPersistedEvent
ModifyVersionDifferencesEvent	AfterLinkIsGeneratedEvent	ModifyImageManipulationPreviewUrlEvent
AfterCacheableContentIsGeneratedEvent	ModifyButtonBarEvent	ModifyLanguagePacksEvent
AfterCachedPageIsPersistedEvent	ShouldUseCachedPageDataIfAvailableEvent	ModifyResultItemInLiveSearchEvent
AfterPageTreeItemsPreparedEvent	AfterBackendPageRenderEvent	AfterVideoPreviewFetchedEvent
IsContentUsedOnPageLayoutEvent	ModifyLinkHandlersEvent	AfterRecordPublishedEvent
ModifyDatabaseQueryForContentEvent	BeforePagePreviewUriGeneratedEvent	ModifyBlindedConfigurationOptionsEvent
PageContentPreviewRenderingEvent	AfterPagePreviewUriGeneratedEvent	SlugRedirectChangelItemCreatedEvent
BeforePageLanguageOverlayEvent	BeforePageWithRootLineIsResolvedEvent	ModifyEditFileFormDataEvent
BeforeFlexFormConfigurationOverrideEvent	AfterPageAndLanguageIsResolvedEvent	BeforeFlexFormDataStructureParsedEvent
IsFileSelectableEvent	ModifyDatabaseQueryForRecordListingEvent	AfterFlexFormDataStructureParsedEvent
SiteConfigurationBeforeWriteEvent	LoginAttemptFailedEvent	BeforeFlexFormDataStructureIdentifierInitializedEvent
SiteConfigurationLoadedEvent		AfterFlexFormDataStructureIdentifierInitializedEvent
		...

6

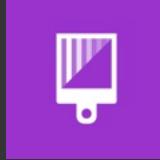
# *BACKEND MODULES*

A purple square icon containing a white link symbol.

Check links

A blue square icon containing a white page and configuration symbol.

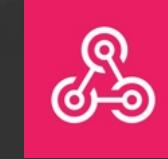
Page TSconfig

A purple square icon containing a white TypoScript symbol.

TypoScript



Content Security Policy



Reactions



Webhooks

10  
**TCA TYPES**

 Color Datetime Email File Folder Json Link Number Password Uuid

10

# TCA CONFIG OPTIONS

- Allowed
- AllowedTypes
- DbType
- Disallowed
- EnableCodeEditor
- ElementBrowserEntryPoints
- Hashed
- IgnorePageTypeRestriction
- Min
- Nullable
- PasswordPolicy
- Required

# PHP **ATTRIBUTES**

- *Backend controller*
- *Extbase annotations*
- *Webhook messages*
- *Upgrade wizards*

# NEUE UND VERBESSERTE REGISTRIERUNG

- Upgrade wizards
- Webhooks
- Reactions
- Data processors
- Extbase type converters
- Content objects
- Element browsers
- Reports and Statuses
- Module functions
- Backend toolbar items
- Context menu item providers
- Linkvalidator linktypes

A photograph of a person from the back, wearing a yellow vest over a grey cap with a small circular logo. They are standing in a field of wildflowers, with a vast mountain range in the background under a cloudy sky.

**MEHR ZU  
ENTDECKEN**

A close-up photograph of a monarch butterfly resting on a textured, light-colored surface. The butterfly's wings are spread, showing its characteristic orange, black, and white patterns. In the blurred background, the blue and white feathers of a bird's head are visible, suggesting the butterfly is near its nest.

# *MIGRATION & INTEGRATION*

# WERBUNG

# TYPO3 SURFCAMP

FUERTEVENTURA  
07 - 14 APRIL 2024

[SURFCAMP.TYPO3.COM](http://SURFCAMP.TYPO3.COM)



**HOOKS → EVENTS**

- Einheitliche **Registrierung**
- Einheitliche **Struktur**
- Vereinfachte **Benutzung**
- Leicht **erweiterbar**
- Out-of-the-box Support für **webhooks**



```
$GLOBALS['TYPO3_CONF_VARS']['SC_OPTIONS']['t3lib/class.t3lib_userauth.php']
['postLoginFailureProcessing']['sendEmailOnFailedLoginAttempt'] =
FailedLoginAttemptNotification::class . '->sendEmailOnLoginFailures';
```



```
TYPO3\CMS\Backend\EventListener\FailedLoginAttemptNotification:
tags:
- name: event.listener
  identifier: 'typo3/cms-backend/failed-login-attempt-notification'
  event: TYPO3\CMS\Core\Authentication\Event\LoginAttemptFailedEvent
  method: 'sendEmailOnLoginFailures'
```



TYP03\CMS\Backend\EventListener\FailedLoginAttemptNotification:

tags:

- name: event.listener  
identifier: 'typo3/cms-backend/failed-login-attempt-notification'
- name: event.listener  
identifier: 'typo3/cms-backend/failed-mfa-verification-notification'



```
public function __invoke(  
    LoginAttemptFailedEvent|MfaVerificationFailedEvent $event  
): void {  
}
```



```
#[AsEventListener(identifier: 'typo3/cms-backend/failed-login-attempt-notification', event: LoginAttemptFailedEvent::class)]
#[AsEventListener(identifier: 'typo3/cms-backend/failed-mfa-verification-notification', event: MfaVerificationFailedEvent::class)]
public function __invoke(
    LoginAttemptFailedEvent | MfaVerificationFailedEvent $event
): void {
}
```

```
● ● ●  
class WizardItemsHook implements NewContentElementWizardHookInterface  
{  
    public function manipulateWizardItems(  
        &$wizardItems,  
        &$parentObject  
    ): void {  
    }  
}
```



```
● ● ●  
class WizardItemsListener  
{  
    public function manipulateWizardItems(  
        ModifyNewContentElementWizardItemsEvent $event  
    ): void {  
        $event->hasWizardItem('itemIdentifier');  
        $event->getWizardItem('itemIdentifier');  
        $event->getWizardItems();  
        $event->setWizardItem('newItem', [], ['after' => 'itemIdentifier']);  
        $event->setWizardItems([]);  
    }  
}
```

# REACTIONS & WEBHOOKS

- Basiert auf PHP-Klassen & Datenbank-Einträgen
- TYPO3 stellt built-in Reactions und Webhooks zur Verfügung
- Verwaltung über eigene Backend Module
- Komplett flexibel und erweiterbar

# REACTIONS

General Access

**Configuration**  
General configuration of the reaction.

**Reaction Type [reaction\_type]**

 Create database record [create-record] ▾

**Name [name]**  
Meaningful name of the reaction  
Create a new category X

**Description [description]**  
Additional information about the reaction.  
This reaction will create a new category record.

**Identifier [identifier]**  
This is your unique reaction identifier within the TYPO3 URL  
87d7c5ba-5c6f-4783-88ce-dd2e7d48839c Copy

**Secret [secret]**  
The secret is required to call the reaction from the outside. The secret can be re-created anytime, but will only be visible once (until the record got saved).  
68b1a916450752efd42e42a1fc73d90857607347 Copy

Additional configuration

**Table [table\_name]**

Select one tables to display the corresponding fields.

 Category [sys\_category] 

**Storage PID [storage\_pid]**

Select the page on which a new record is created on.

 Find records

[No title] [pages\_637]   

 Page

 [No title] [637] /styleguide/

**Impersonate User [impersonate\_user]**

Select the user with the appropriate access rights that is allowed to add a record of this type. If in doubt, use the CLI user.

 Find records

\_cli\_ [be\_users\_18]   

 Backend user

 \_cli\_ [18] /

**Fields [fields]**

The available fields depend on the selected table. The usage of placeholders like \${foo.bar} is possible.

**Title**

Category \${title}

**Description**

My new category



```
curl -X 'POST' \
'https://typo3.site/typo3/reaction/87d7c5ba-5c6f-4783-88ce-dd2e7d48839c' \
-H 'accept: application/json' \
-H 'x-api-key: 68b1a916450752efd42e42a1fc73d90857607347' \
-H 'Content-Type: application/json' \
-d '{"title": "dynamic title"}'
```



```
{  
    "success": true  
}
```

- Registrierung über **ReactionInterface**
- Handling in **react()** Methode
- TCA Registrierung und Konfiguration



```
final class MyReaction implements ReactionInterface
{
    public static function getType(): string
    {
        return 'my-reaction';
    }

    public static function getDescription(): string
    {
        return 'My simple reaction';
    }

    public static function getIconIdentifier(): string
    {
        return 'actions-info';
    }

    public function react(
        ServerRequestInterface $request,
        array $payload,
        ReactionInstruction $reaction
    ): ResponseInterface {
        return new JsonResponse(['success' => true]);
    }
}
```

# WEBHOOKS



General    HTTP Settings    Access

**Configuration**  
General configuration of the webhook.

**Webhook Trigger [webhook\_type]**  
Trigger the webhook, ...  
... when a page is added or changed [typo3/content/page-modification] ▾

**Identifier [identifier]**  
This is your unique webhook identifier  
af7ac75f-cdb0-4eee-b36c-e9123c671a7a 

**Name [name]**  
Meaningful name of the webhook  
Page changed 

**Description [description]**  
Additional information about the webhook.  


**URL [url]**  
Target URL that should be called.  
 https://www.typo3.site/page-trigger  

**Secret [secret]**  
The secret is mandatory to create a hash that is sent with the request. The secret can be used to verify the payload has not been modified, see the documentation for more details.  
5e5d025cfafecfec489a97e33160ff63f4d136d8 

General    **HTTP Settings**    Access

### HTTP Settings

Advanced HTTP request settings.

**HTTP Method** [method]

POST [POST] ▾

**Verify SSL** [verify\_ssl]

Verify that the connection is secure and valid, this setting should only be disabled in case a verification is not possible.

[1]

**Additional Headers** [additional\_headers]

Additional headers that should be added to the HTTP request. Data must be provided as valid JSON string.

```
1 v {  
2   "X-CUSTOM-HEADER": "New API"  
3 }
```



```
X-Custom-Header: New API
Webhook-Signature-Algo: sha256
Webhook-Signature: 642934714fa90cc0215bde2a9e1f8080dee096db12ea923c36ee4442a54147da
Content-Type: application/json
Content-Length: 1311
User-Agent: TYP03
Host: typo3.site
```

```
{
  "action": "new",
  "identifier": 1186,
  "record": {
    "title": "new-page",
    "slug": "/webhooks/new-page",
  },
  "url": "https://typo3.site/webhooks/new-page.html",
  "site": "styleguide",
  "workspace": 0,
  "author": {
    "uid": 1,
    "username": "oli",
    "isAdmin": true
  }
}
```



```
hash_hmac('sha256', sprintf(
  '%s:%s',
  $webhookRecordIdentifier,
  $requestBody,
), $webhookRecordSecret);
```

- **PHP Klasse**, auto-tagged über **WebhookMessage Attribut**
  - identifier
  - description
  - (factory) Methode - optional
- Automatische Konfiguration über CompilerPass
- Automatische TCA Registrierung

## WebhookMessageHandler

→ *WebhookTypesRegistry*

getConfiguredWebhooksByType(\$message)

→ *WebhookInstructions*

sendRequest(\$webhookInstruction, \$message)

# **VERWENDUNG VON EVENTS**



```
#[WebhookMessage(
    identifier: 'my-ext/cache-flushed',
    description: '... when cache is flushed'
)]
final class CacheFlushMessage implements WebhookMessageInterface
{
    public function __construct(
        private readonly array $errors,
        private readonly array $groups
    ) {
    }

    public static function createFromEvent(CacheFlushEvent $event): self
    {

        return new self($event->getErrors(), $event->getGroups());
    }

    public function jsonSerialize(): array
    {
        return [
            'errors' => $this->errors,
            'groups' => $this->groups,
        ];
    }
}
```

- Handling über “Factory Methode” `createFromEvent($event)`
- `MessageListener` hört auf \$event
- `MessageFactory` ruft “Factory Methode” und erstellt Message

TCA

***TYPES & OPTIONS***

**WENIGER KONFIGURATION  
MEHR KLARHEIT**

```
● ● ●  
'file' => [  
    'label' => 'File field',  
    'config' => [  
        'type' => 'inline',  
        'foreign_table' => 'sys_file_reference',  
        'foreign_field' => 'uid_foreign',  
        'foreign_sortby' => 'sorting_foreign',  
        'foreign_table_field' => 'tablenames',  
        'foreign_match_fields' => [  
            'fieldname' => 'file',  
        ],  
        'foreign_label' => 'uid_local',  
        'foreign_selector' => 'uid_local',  
        'overrideChildTca' => [  
            'columns' => [  
                'uid_local' => [  
                    'config' => [  
                        'appearance' => [  
                            'elementBrowserType' => 'file',  
                            'elementBrowserAllowed' => 'jpg,png',  
                        ],  
                    ],  
                ],  
            ],  
            'filter' => [  
                [  
                    'userFunc' => FileExtensionFilter::class . '->filterInlineChildren',  
                    'parameters' => [  
                        'allowedFileExtensions' => $GLOBALS['TYPO3_CONF_VARS']['GFX']['imagefile_ext'],  
                        'disallowedFileExtensions' => 'gif',  
                    ],  
                ],  
            ],  
        ],  
    ],  
],
```



```
● ● ●  
'file' => [  
    'label' => 'File field',  
    'config' => [  
        'type' => 'file',  
        'allowed' => ['common-media-types'],  
        'disallowed' => ['gif']  
    ],  
],
```



```
'password_field' => [
    'label' => 'Password',
    'config' => [
        'type' => 'input',
        'eval' => 'trim,password,saltedPassword',
    ],
],
```



```
'password_field' => [
    'label' => 'Password',
    'config' => [
        'type' => 'password',
        'passwordPolicy' => 'custom',
        'fieldControl' => [
            'passwordGenerator' => [
                'renderType' => 'passwordGenerator',
                'options' => [
                    'title' => 'Generate a password',
                    'allowEdit' => false,
                    'passwordRules' => [
                        'length' => 38,
                        'digitCharacters' => false,
                        'specialCharacters' => true,
                    ],
                ],
            ],
        ],
    ],
],
```



# PASSWORD POLICIES

- Erzwinge “password policies” im Back- und Frontend
- Built-in Regeln and Validatoren
- Komplett individualisier- und erweiterbar
- Eigener TCA Typ mit Optionen und "fieldControl"
- PSR-14 Event zum Erweitern der “Validation Context” Daten

# CONTENT SECURITY POLICY

# Backend Module API

The image shows the TYPO3 v11 Warp Speed backend interface. At the top, there is a search bar with the placeholder "Enter search term". Below the search bar is a toolbar with various icons: a document, a user group, a forward arrow, a red back arrow, a link, a folder, a trash can, and a divide icon. The main content area displays a list of files and folders. The first item in the list is "The Motion Picture". Other items visible include "The Wrath of Khan", "The Search for Spock", "The Voyage Home", "The Final Frontier", and "The Undiscovered Country". On the left side of the screen, there is a sidebar with navigation links: Dashboard (with a purple icon), Web (with a white icon), Page (with an orange icon), View (with a red icon), List (with an orange icon), and Forms (with a blue icon). The overall theme is dark with light-colored text and icons.

- `$TBE_MODULES` und *ModuleLoader* entfernt
- Registrierung zur **build-time** in der *ModuleRegistry*
- Konfiguration in *Configuration/Backend/Modules.php*
- Zugriff auf Module über ***ModuleProvider API***
- Modul-Objekt im PSR-7 Request
- *PSR-14 Event zur Manipulation der Konfiguration*



```
ExtensionManagementUtility::addModule(  
    'web',  
    'module',  
    'top',  
    '',  
    [  
        'routeTarget' => MyModuleController::class . '::handleRequest',  
        'name' => 'web_module',  
        'access' => 'admin',  
        'workspaces' => 'online',  
        'iconIdentifier' => 'module-example',  
        'labels' => 'LLL:EXT:example/Resources/Private/Language/locallang_mod.xlf',  
        'navigationComponentId' => 'TYP03/CMS/Backend/PageTree/PageTreeElement',  
    ]  
);
```



```
return [
    'web_module' => [
        'parent' => 'web',
        'position' => ['before' => '*'],
        'access' => 'admin',
        'workspaces' => 'live',
        'path' => '/module/web/example',
        'iconIdentifier' => 'module-example',
        'navigationComponent' => 'TYPO3/CMS/Backend/PageTree/PageTreeElement',
        'labels' => 'LLL:EXT:example/Resources/Private/Language/locallang_mod.xlf',
        'routes' => [
            '_default' => [
                'target' => MyExampleModuleController::class . '::handleRequest',
            ],
        ],
    ],
];
```



```
'extensionName' => 'MyExtbaseExtension',
'controllerActions' => [
    MyExtbaseExtensionModuleController::class => [
        'index',
        'detail'
    ],
],
```

# SPEZIFISCHE ROUTEN & DYNAMISCHE URL SEGMENTE



```
return [
    'web_module' => [
        'parent' => 'web',
        'path' => '/module/web/example',
        'routes' => [
            '_default' => [
                'target' => MyModuleController::class . '::overview',
            ],
            'edit' => [
                'path' => '/custom-path',
                'target' => MyModuleController::class . '::edit',
            ],
            'manage' => [
                'target' => AnotherController::class . '::manage',
                'methods' => ['POST'],
            ],
            'delete' => [
                'path' => '/delete/{identifier}',
                'target' => AnotherController::class . '::delete',
                'methods' => ['POST'],
            ],
        ],
    ],
];
```



```
public function delete(ServerRequestInterface $request): ResponseInterface
{
    $identifier = $request->getAttribute('routing')['identifier'];
}
```

# ALIASES



```
return [
    'new_name' => [
        'parent' => 'web',
        ...
        'aliases' => [ 'web_module' ],
    ],
];
```

# *MODULE FUNCTIONS API*

- *\$TBE\_MODULES\_EXT* entfernt
- Definition von Zugriffsberechtigungen möglich
- Identischer Funktionsumfang wie für “main” Module
- Definition von Zugriffsberechtigungen möglich



```
ExtensionManagementUtility::insertModuleFunction(  
    'new_name',  
    MyThirdLevelController::class,  
    '',  
    'LLL:EXT:extkey/Resources/Private/Language/locallang.xlf:mod_title'  
);
```



```
'new_name_third' => [
    'parent' => 'new_name',
    'path' => '/module/new/name/thrid',
    'labels' => [
        'title' => 'LLL:EXT:extkey/Resources/Private/Language/locallang.xlf:mod_title',
    ],
    'routes' => [
        '_default' => [
            'target' => MyThirdLevelController::class . '::handleRequest',
        ],
    ],
    'moduleData' => [
        'someData' => 0,
    ],
],
```

# *MODULE DATA API*

- Speichert Benutzerspezifischen “module state”
- Definition der erlaubten GET/POST Parameter
- ModuleData Objekt verfügbar im PSR-7 Request



```
return [
    'new_name' => [
        'parent' => 'web',
        ...
        'moduleData' => [
            'language' => 0,
            'showHiddenElements' => true,
        ],
    ],
];
```



```
$moduleData = $request->getAttribute('moduleData');
$moduleIdentifier = $request->getAttribute('module')->getIdentifier();

if ($moduleData->clean('showHiddenElements', [true, false])) {
    $backendUser->pushModuleData($moduleIdentifier, $moduleData->toArray());
}

$showHiddenElements = $moduleData->get('showHiddenElements');
```

# MODULE TEMPLATE API

- Rendering des “content frame” im Backend
  - Erstellung durch Factory-Methode
- 
- Automatisches Auffinden der Templates
  - Automatische Erstellung der PSR-7 Response
  - Unterstützung von globalen “Template overrides”



```
$moduleTemplate = $this->moduleTemplateFactory->create($request);
$view = GeneralUtility::makeInstance(StandaloneView::class);
$view->setTemplateRootPaths(['EXT:my_ext/Resources/Private/Templates']);
$view->assign('content', 'My module content');
$moduleTemplate->setContent($view->render('MyModuleTemplate'));
return $this->responseFactory->createResponse()
    ->withHeader('Content-Type', 'text/html; charset=utf-8')
    ->withBody($this->streamFactory->createStream($moduleTemplate->renderContent()));
```



```
return $this->moduleTemplateFactory
    ->create($request)
    ->assign('content', 'My module content')
    ->renderResponse('MyModuleTemplate');
```



**LINK API**

- Neue **LinkFactory** basiert auf **LinkResult(Interface)**
- Verbessertes **LinkResult**
  - Stellt Standard “**Ausgabe-Formate**” bereit
- PSR-14 Event zur Manipulation **aller generierten Links**

# SITE SETTINGS API

- **SiteSettings (READ) API**
- Spezielles "settings file"
- TypoScript **getData** Support
- PSR-14 Events zur Manipulation der "settings"
  
- "Glob pattern" in Importen
- "Site config" im Konfigurations-Modul verfügbar



```
$siteSettings = $request->getAttribute('site')->getSettings();  
  
$allSettings = $siteSettings->all();  
$allSitePids = $siteSettings->get('sites');  
$newsDetailPid = (int)$siteSettings->get('sites.newsDetail', 110);
```

WAS KOMMT  
ALS NÄCHSTES?



- Ersetzen letzter Hooks durch PSR-14 Events
- Hinzufügen von mehr PSR-14 Events
- Verbesserung / Erweiterung der TCA API
- Verbesserung / Erweiterung Settings API
- Doctrine DBAL 4 & CommonTableExpressions API

***FRAGEN?***

***IDEEN?***

**VIELEN DANK!**