TCA Migration mit Rector

Marianne Moser

TUGA #89 – 10.10.2023

Rector Recap

Was ist Rector und wofür überhaupt?

- PHP-Library zur (Teil-)Automatisierung von Upgrades und zum Refactoring von PHP-Code.
- Funktioniert regelbasiert, es können beliebige Rules oder Sets angewendet werden.
- Offizielle Doku: https://getrector.com/documentation/
- Erweiterung für TYPO3: https://github.com/sabbelasichon/typo3-rector

Abstract Syntax Tree (AST)

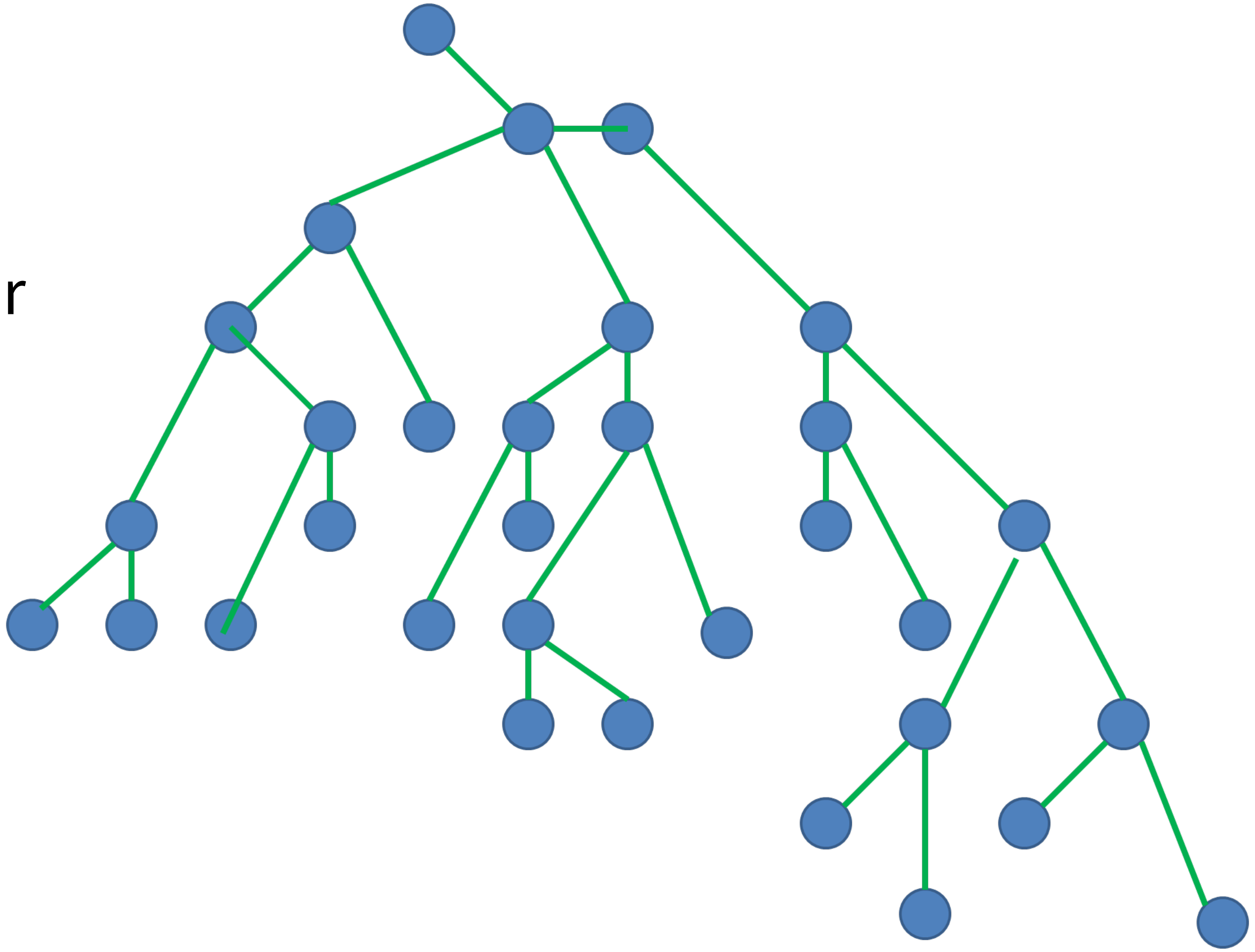
Rector iteriert über alle Files und bildet einen „Abstract Syntax Tree“.

-> Alle Elemente sind in Form von Nodes in einer Baumstruktur gegliedert.

z. B. ist ebenfalls ein Node:

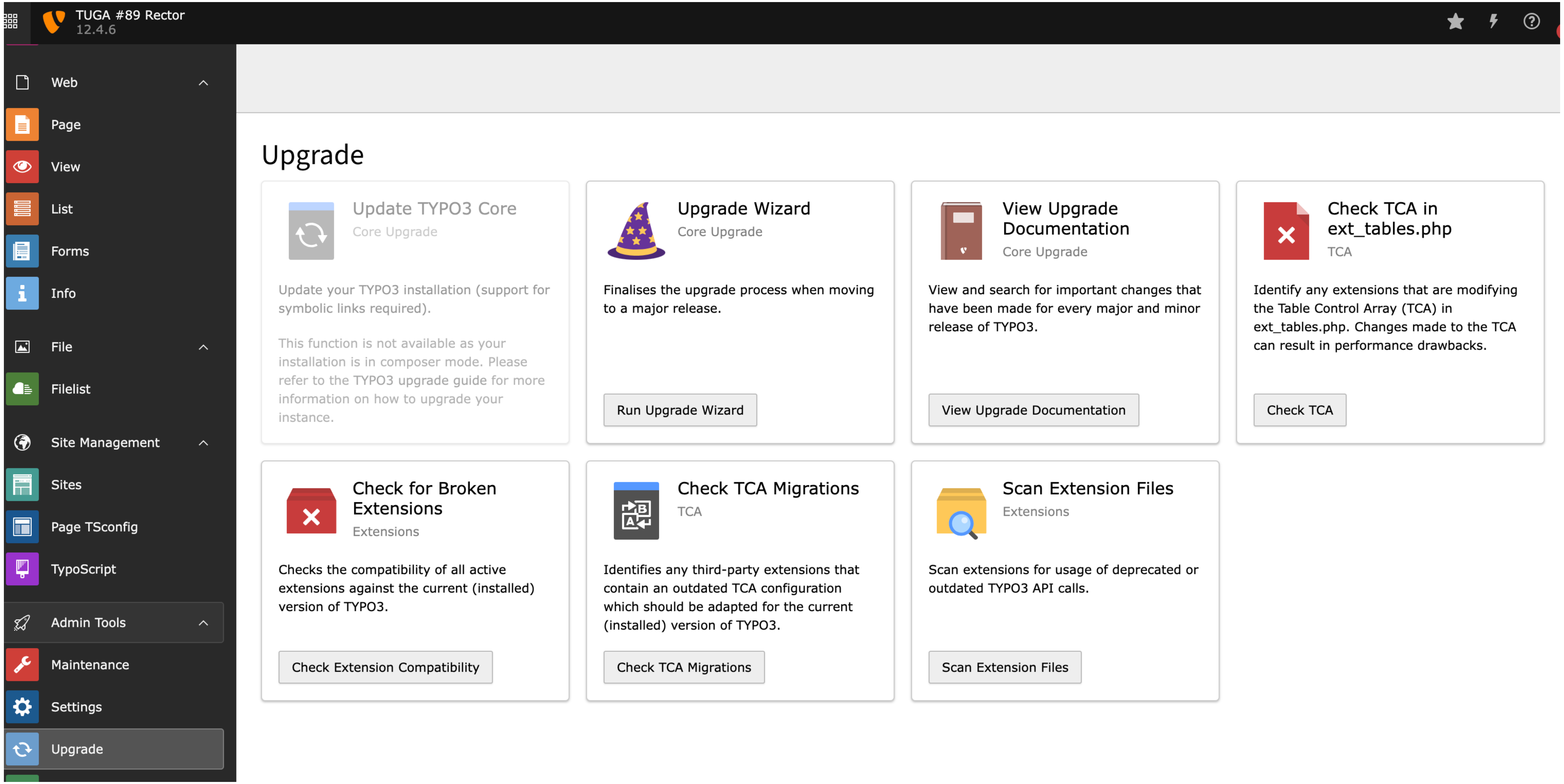
```
new String_('hello world!');
```

AST wird wieder zu PHP-Code geschrieben.



Hands-on: TCA Migration Rule

Beispiel: TCA Migrations



Upgrade

Update TYPO3 Core
Core Upgrade

Update your TYPO3 installation (support for symbolic links required).

This function is not available as your installation is in composer mode. Please refer to the TYPO3 upgrade guide for more information on how to upgrade your instance.

Upgrade Wizard
Core Upgrade

Finalises the upgrade process when moving to a major release.

Run Upgrade Wizard

View Upgrade Documentation
Core Upgrade

View and search for important changes that have been made for every major and minor release of TYPO3.

View Upgrade Documentation

Check TCA in ext_tables.php
TCA

Identify any extensions that are modifying the Table Control Array (TCA) in ext_tables.php. Changes made to the TCA can result in performance drawbacks.

Check TCA

Check for Broken Extensions
Extensions

Checks the compatibility of all active extensions against the current (installed) version of TYPO3.

Check Extension Compatibility

Check TCA Migrations
TCA

Identifies any third-party extensions that contain an outdated TCA configuration which should be adapted for the current (installed) version of TYPO3.

Check TCA Migrations

Scan Extension Files
Extensions

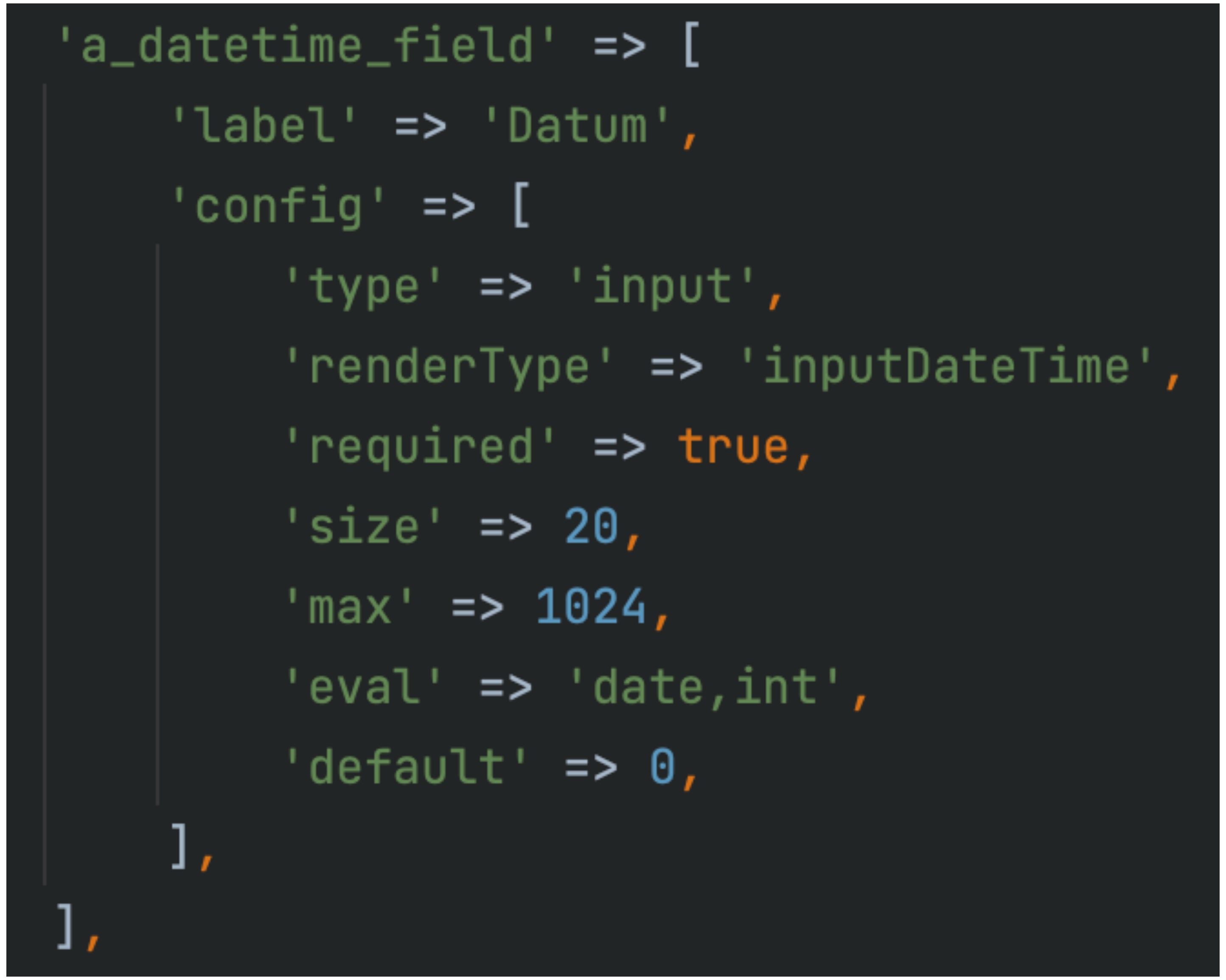
Scan extensions for usage of deprecated or outdated TYPO3 API calls.

Scan Extension Files

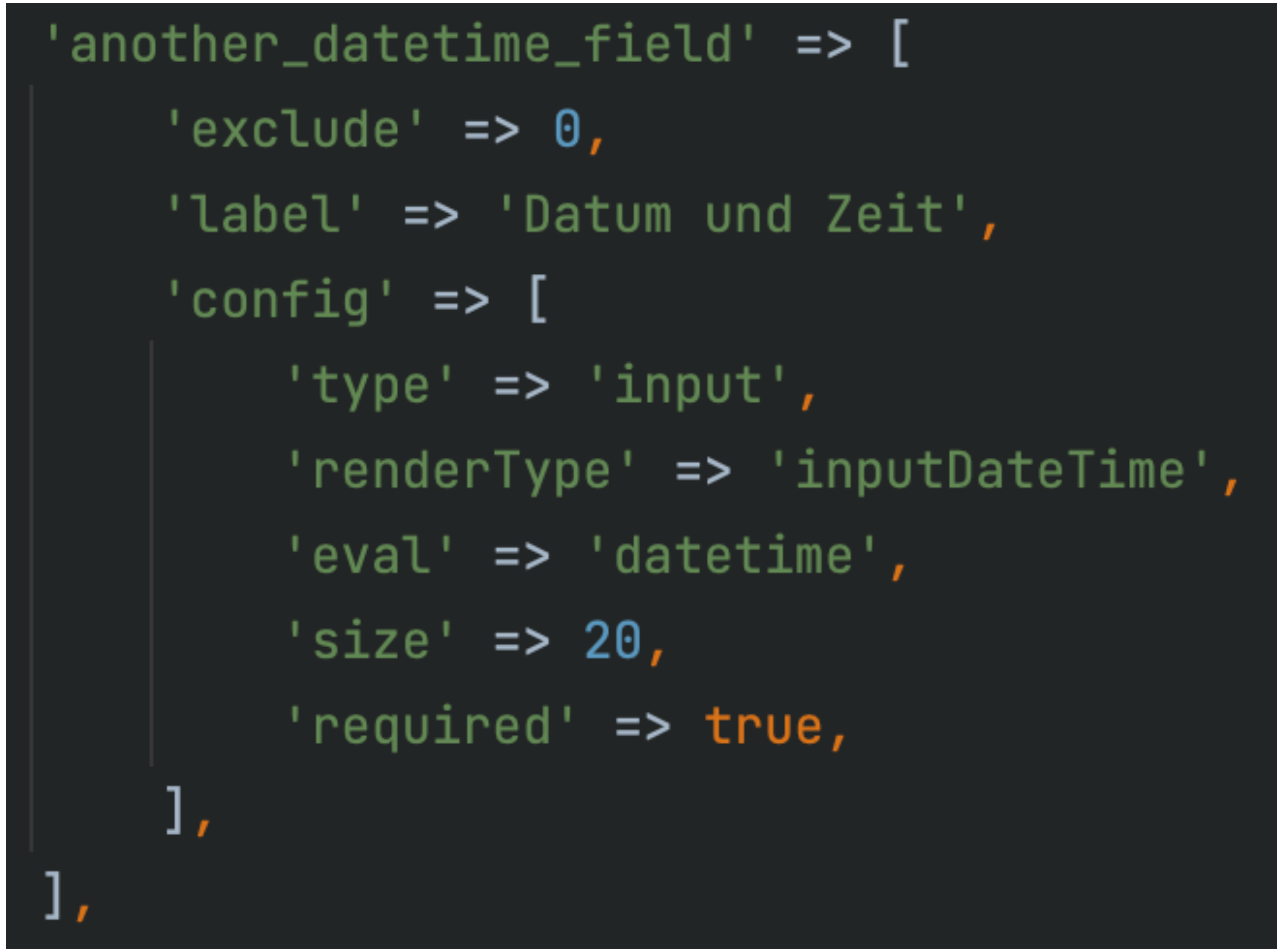
Problem identifizieren: Check TCA Migrations

> The TCA field 'a_datetime_field' of table 'pages' defines renderType="inputDateTime". The field has therefore been migrated to the TCA type 'datetime'. This includes corresponding migration of the "eval" list, as well as obsolete field configurations, such as "max". Please adjust your TCA accordingly.

> The TCA field 'another_datetime_field' of table 'pages' defines renderType="inputDateTime". The field has therefore been migrated to the TCA type 'datetime'. This includes corresponding migration of the "eval" list, as well as obsolete field configurations, such as "max". Please adjust your TCA accordingly.

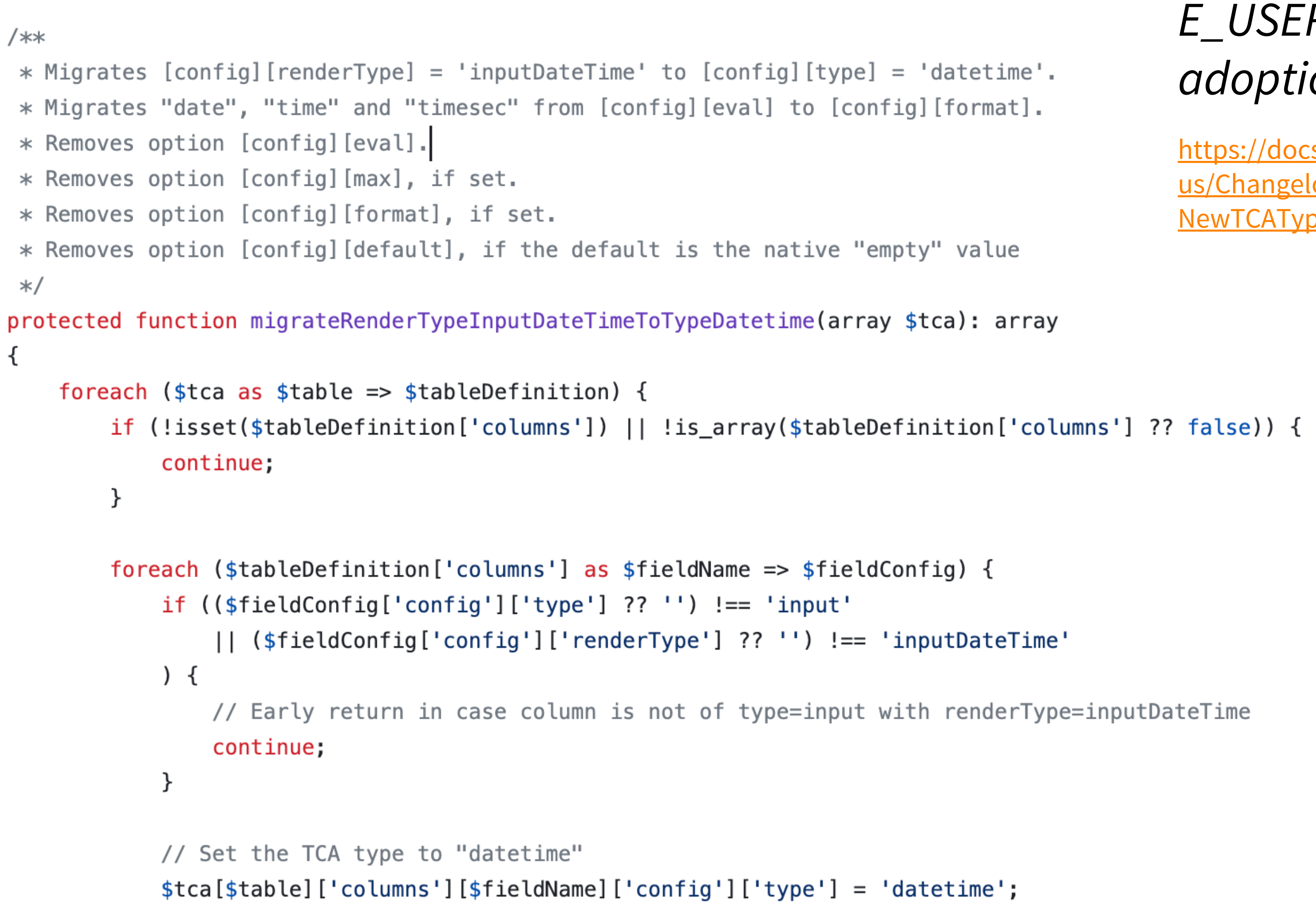


```
'a_datetime_field' => [
    'label' => 'Datum',
    'config' => [
        'type' => 'input',
        'renderType' => 'inputDateTime',
        'required' => true,
        'size' => 20,
        'max' => 1024,
        'eval' => 'date,int',
        'default' => 0,
    ],
],
```



```
'another_datetime_field' => [
    'exclude' => 0,
    'label' => 'Datum und Zeit',
    'config' => [
        'type' => 'input',
        'renderType' => 'inputDateTime',
        'eval' => 'datetime',
        'size' => 20,
        'required' => true,
    ],
],
```

Automatic TCA Migrations



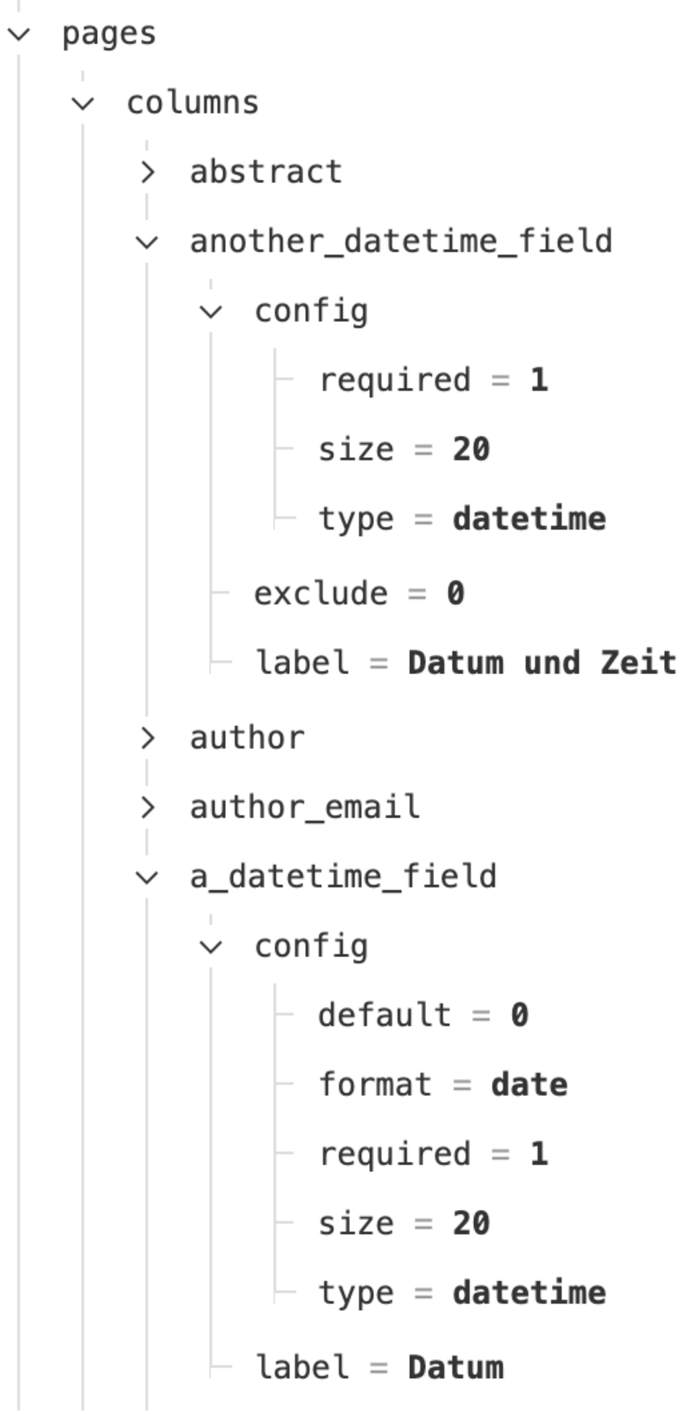
```
/**
 * Migrates [config][renderType] = 'inputDateTime' to [config][type] = 'datetime'.
 * Migrates "date", "time" and "timesec" from [config][eval] to [config][format].
 * Removes option [config][eval].
 * Removes option [config][max], if set.
 * Removes option [config][format], if set.
 * Removes option [config][default], if the default is the native "empty" value
 */
protected function migrateRenderTypeInputDateTimeToTypeDatetime(array $tca): array
{
    foreach ($tca as $table => $tableDefinition) {
        if (!isset($tableDefinition['columns']) || !is_array($tableDefinition['columns'] ?? false)) {
            continue;
        }

        foreach ($tableDefinition['columns'] as $fieldName => $fieldConfig) {
            if (($fieldConfig['config']['type'] ?? '') !== 'input'
                || ($fieldConfig['config']['renderType'] ?? '') !== 'inputDateTime'
            ) {
                // Early return in case column is not of type=input with renderType=inputDateTime
                continue;
            }

            // Set the TCA type to "datetime"
            $tca[$table]['columns'][$fieldName]['config']['type'] = 'datetime';
```

> *„An automatic TCA migration is performed on the fly, migrating all occurrences to the new TCA type and triggering a PHP E_USER_DEPRECATED error where code adoption has to take place.“*

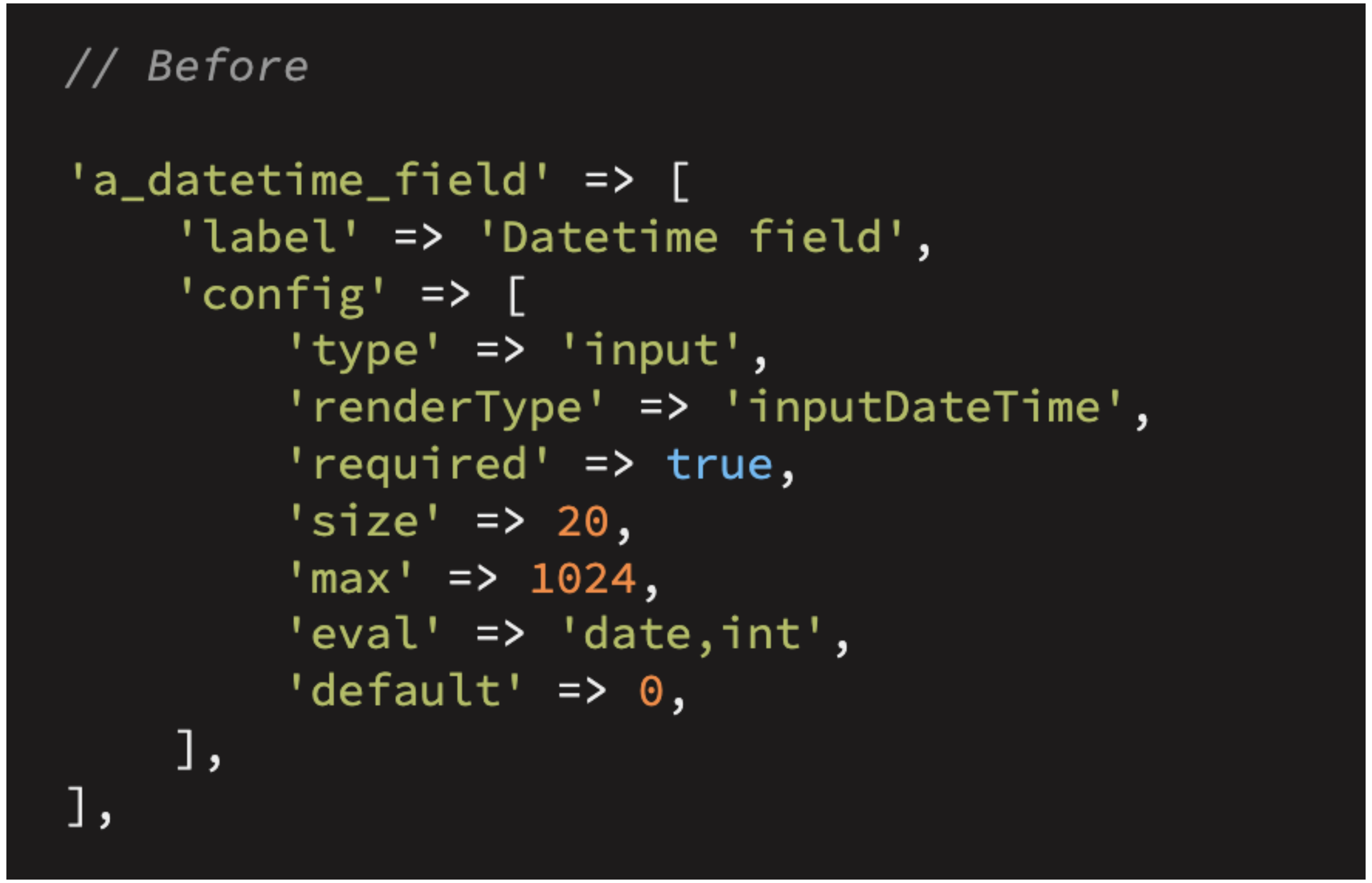
https://docs.typo3.org/c/typo3/cms-core/main/en-us/Changelog/12.0/Feature-97232-NewTCATypeDatetime.html



```
⌄ pages
  ⌄ columns
    › abstract
    ⌄ another_datetime_field
      ⌄ config
          required = 1
          size = 20
          type = datetime
        exclude = 0
        label = Datum und Zeit
    › author
    › author_email
    ⌄ a_datetime_field
      ⌄ config
          default = 0
          format = date
          required = 1
          size = 20
          type = datetime
        label = Datum
```

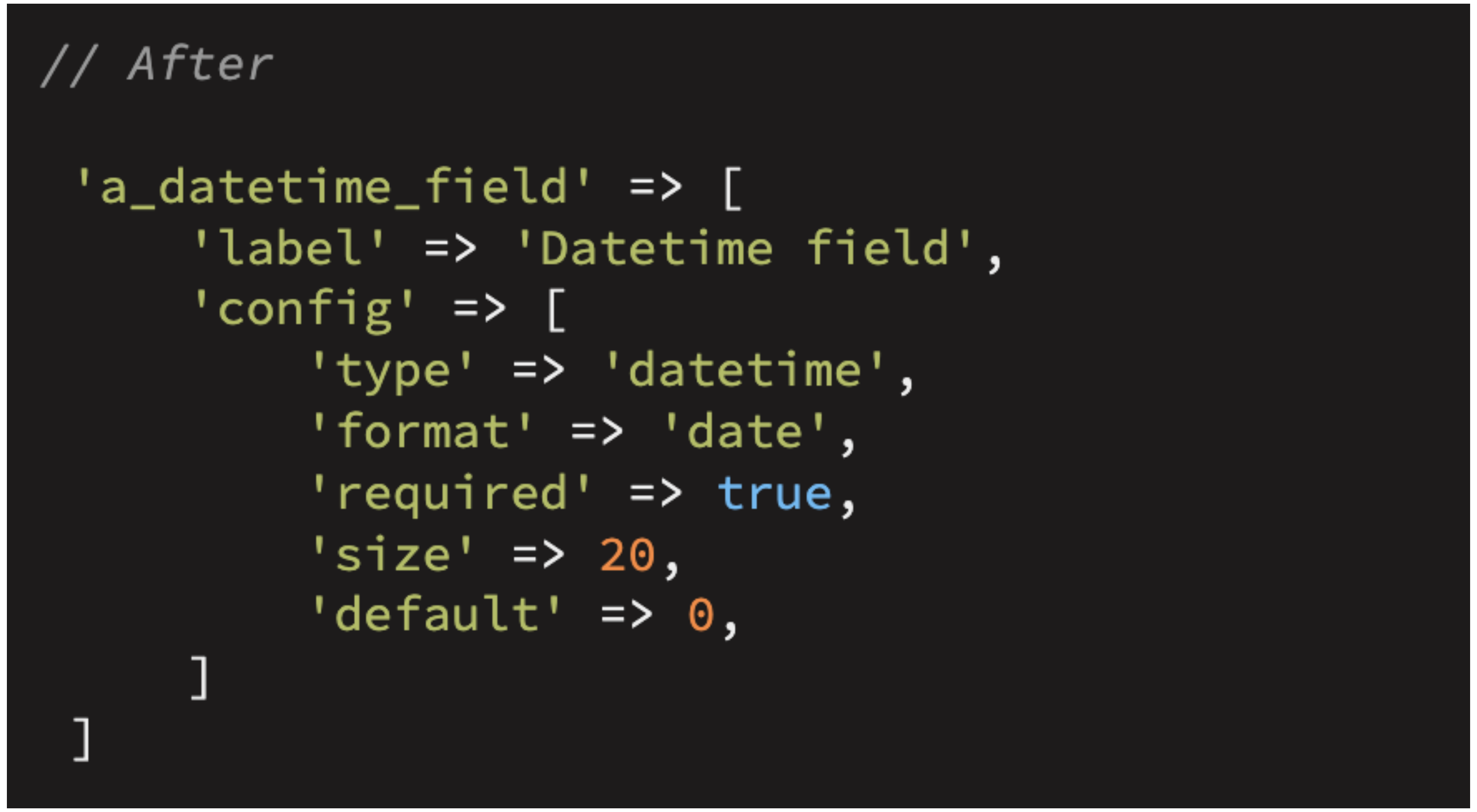
Quelle: https://github.com/TYPO3/typo3/blob/main/typo3/sysext/core/Classes/Migrations/TcaMigration.php

TYPO3 12.4: TCA Reference - Datetime



```
// Before

'a_datetime_field' => [
    'label' => 'Datetime field',
    'config' => [
        'type' => 'input',
        'renderType' => 'inputDateTime',
        'required' => true,
        'size' => 20,
        'max' => 1024,
        'eval' => 'date,int',
        'default' => 0,
    ],
],
```



```
// After

'a_datetime_field' => [
    'label' => 'Datetime field',
    'config' => [
        'type' => 'datetime',
        'format' => 'date',
        'required' => true,
        'size' => 20,
        'default' => 0,
    ]
]
```

Quelle: https://docs.typo3.org/m/typo3/reference-tca/12.4/en-us/ColumnsConfig/Type/Datetime/Index.html

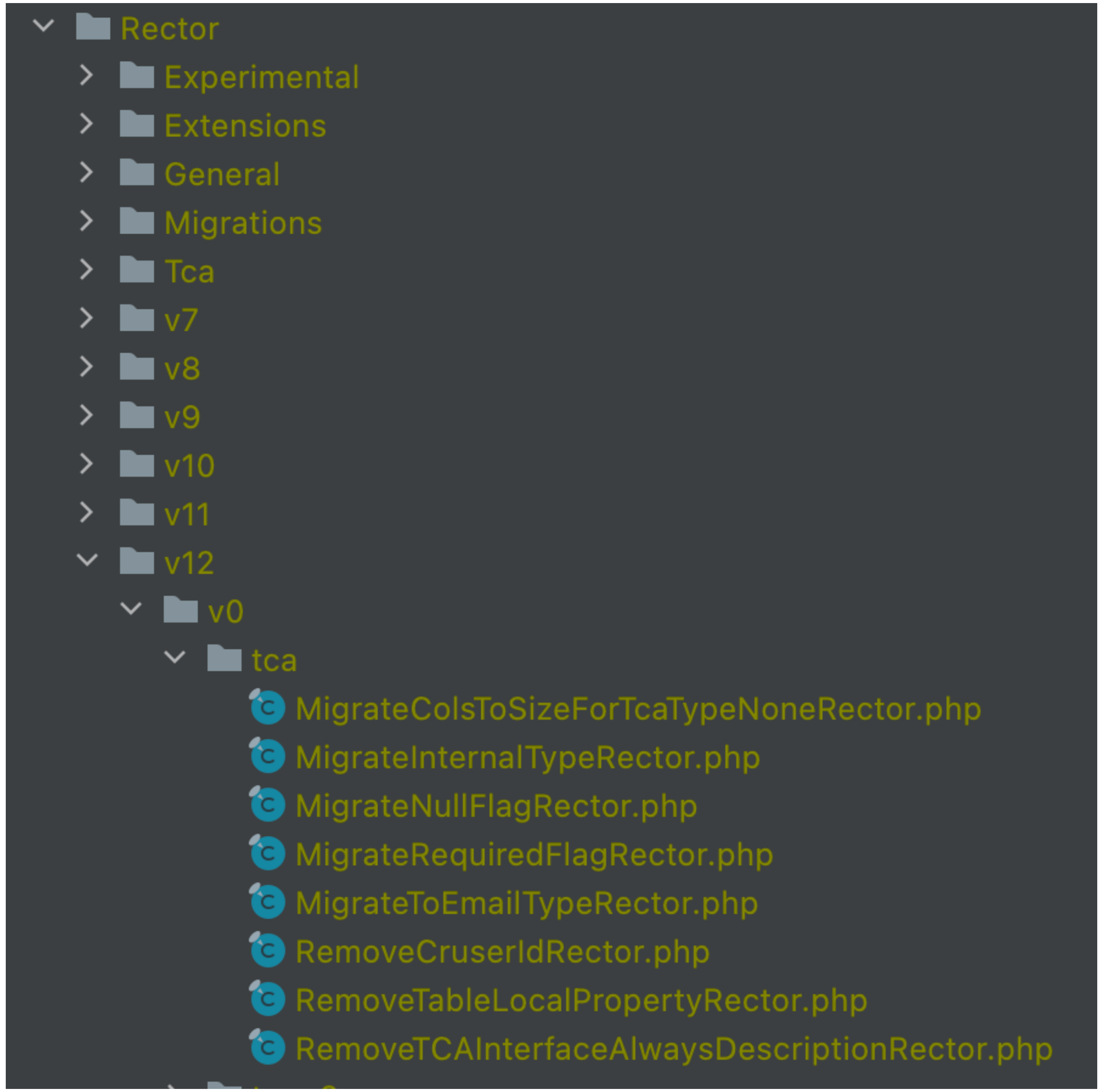
Vorbereitungen

Zuerst überprüfen, ob es die Rule schon gibt:

```
vendor/ssch/typo3-rector/src/Rector
vendor/rector/rector/rules
```

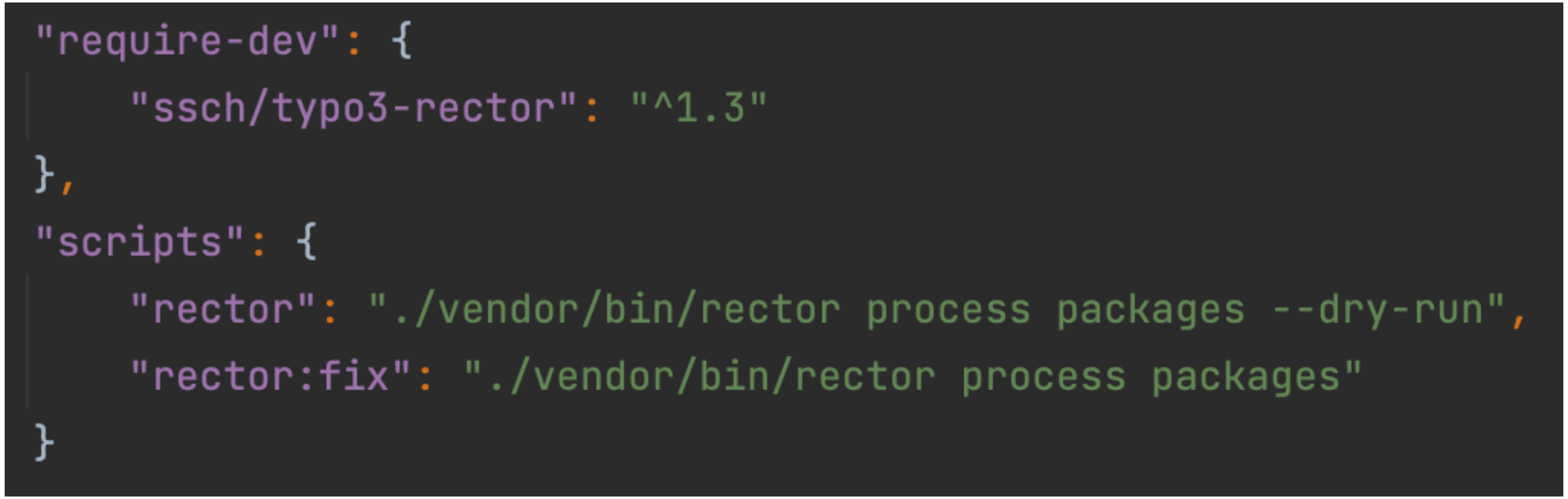
Bzw. in den Pull-Requests und Forks:

https://github.com/sabbelasichon/typo3-rector/pulls
https://github.com/sabbelasichon/typo3-rector/forks



```
Rector
  Experimental
  Extensions
  General
  Migrations
  Tca
  v7
  v8
  v9
  v10
  v11
  v12
    v0
      tca
        MigrateColsToSizeForTcaTypeNoneRector.php
        MigrateInternalTypeRector.php
        MigrateNullFlagRector.php
        MigrateRequiredFlagRector.php
        MigrateToEmailTypeRector.php
        RemoveCruserIdRector.php
        RemoveTableLocalPropertyRector.php
        RemoveTCAInterfaceAlwaysDescriptionRector.php
```

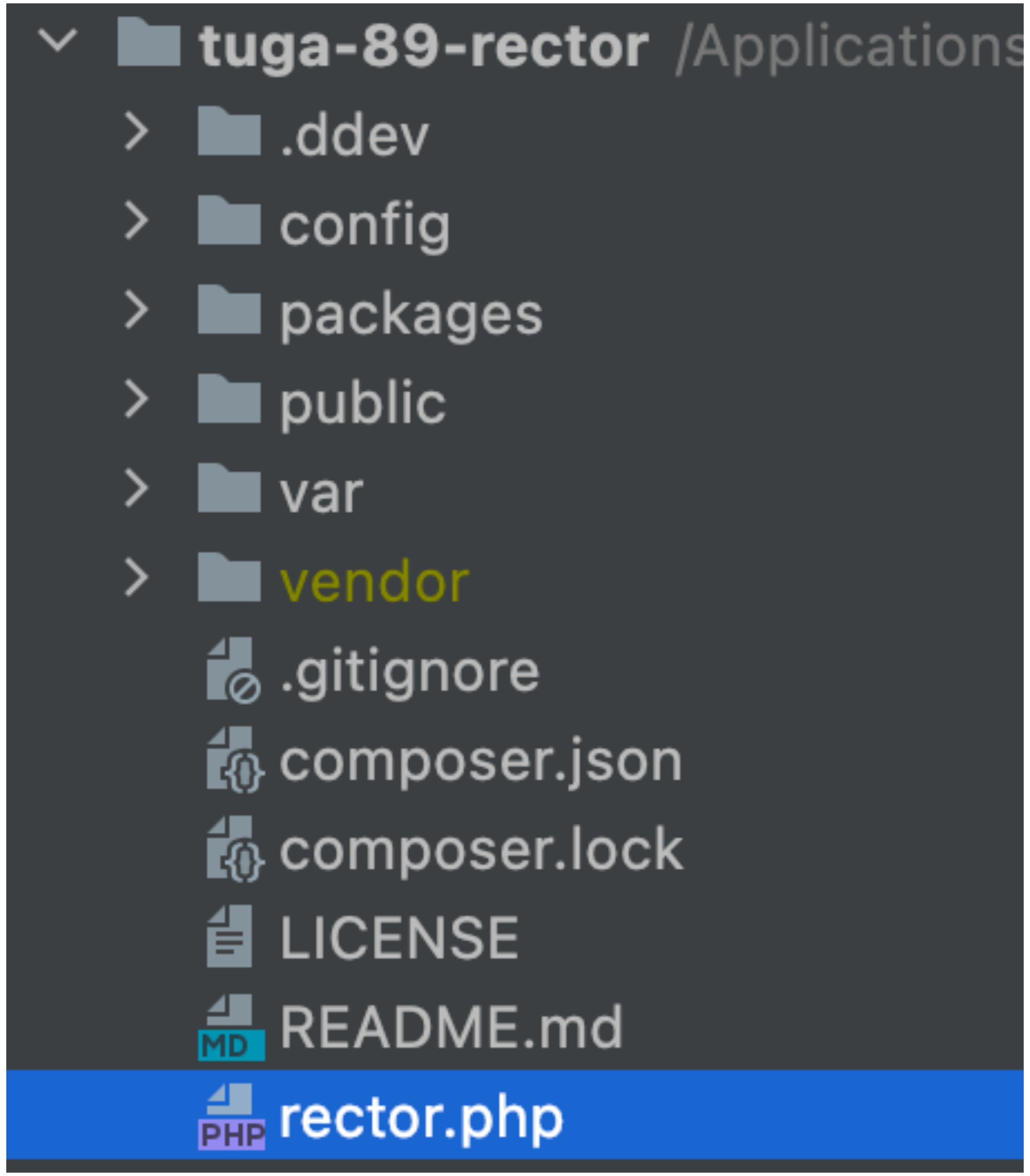
Package + Scripts in composer.json



```
"require-dev": {
    "ssch/typo3-rector": "^1.3"
},
"scripts": {
    "rector": "./vendor/bin/rector process packages --dry-run",
    "rector:fix": "./vendor/bin/rector process packages"
}
```

Mit `ddev composer run rector` werden die Changes vorerst nur in der Console ausggeben.

Rector-Config: rector.php



```
tuga-89-rector /Applications
  .ddev
  config
  packages
  public
  var
  vendor
  .gitignore
  composer.json
  composer.lock
  LICENSE
  README.md
  rector.php
```

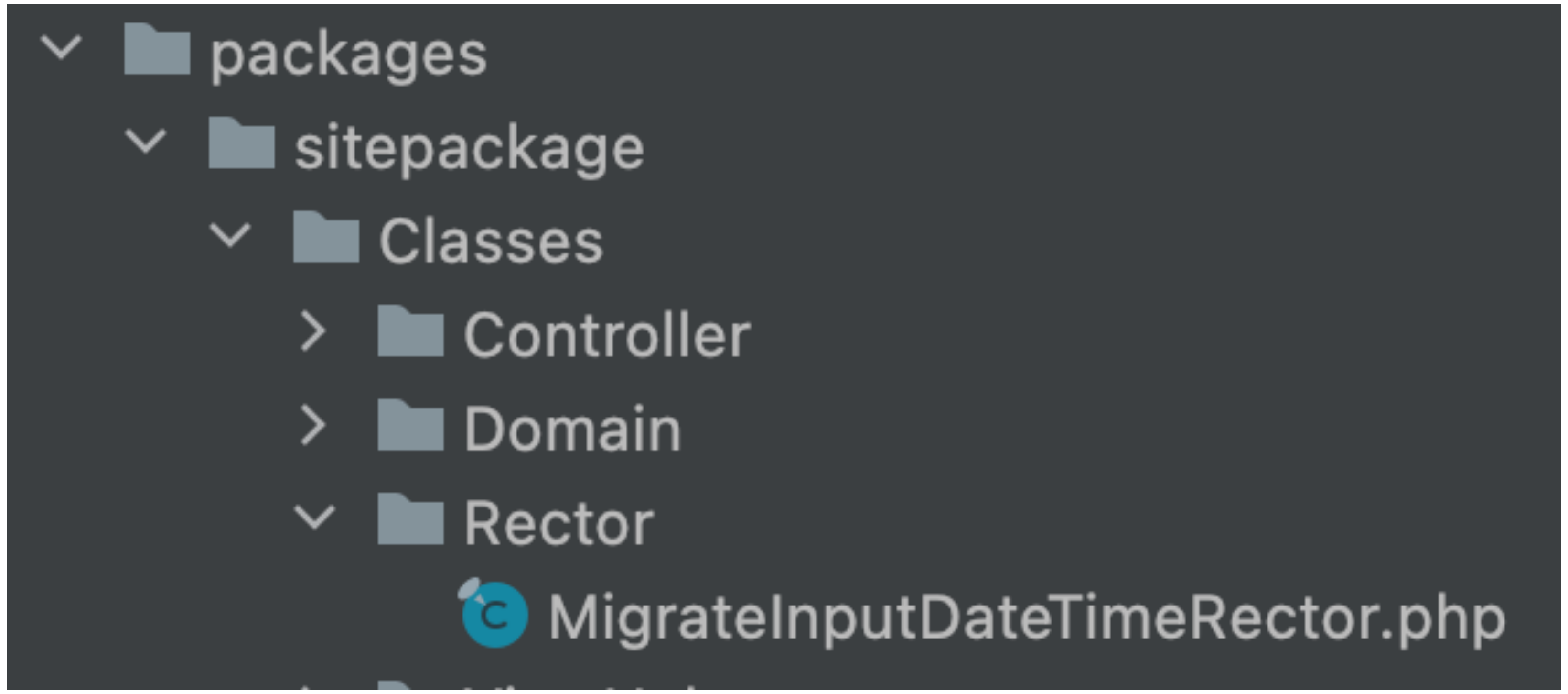
Per Default wird im Projektroot nach der Datei `rector.php` gesucht.

Optional kann der Parameter `--config` verwendet werden.

Beispiel:
https://github.com/sabbelasichon/typo3-rector/blob/main/docs/configuration_and_processing.md

Rector-Config: Regel definieren in rector.php

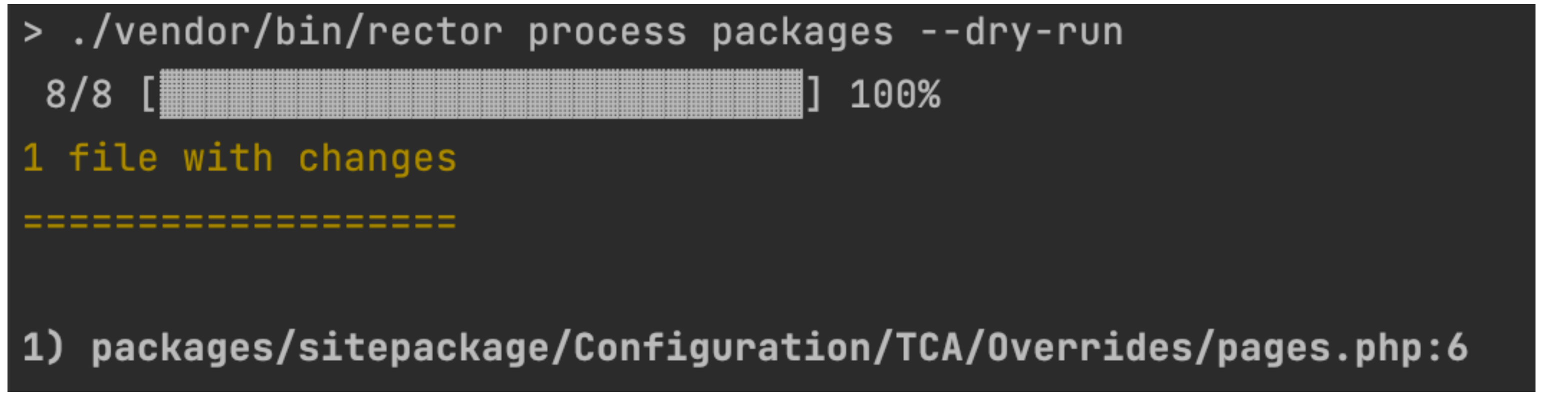
```
// register a single rule
$rectorConfig
->rule(\Vendor\Sitepackage\Rector\
MigrateInputDateTimeRector::class);
```



```
packages
  sitepackage
    Classes
      Controller
      Domain
      Rector
        MigrateInputDateTimeRector.php
```

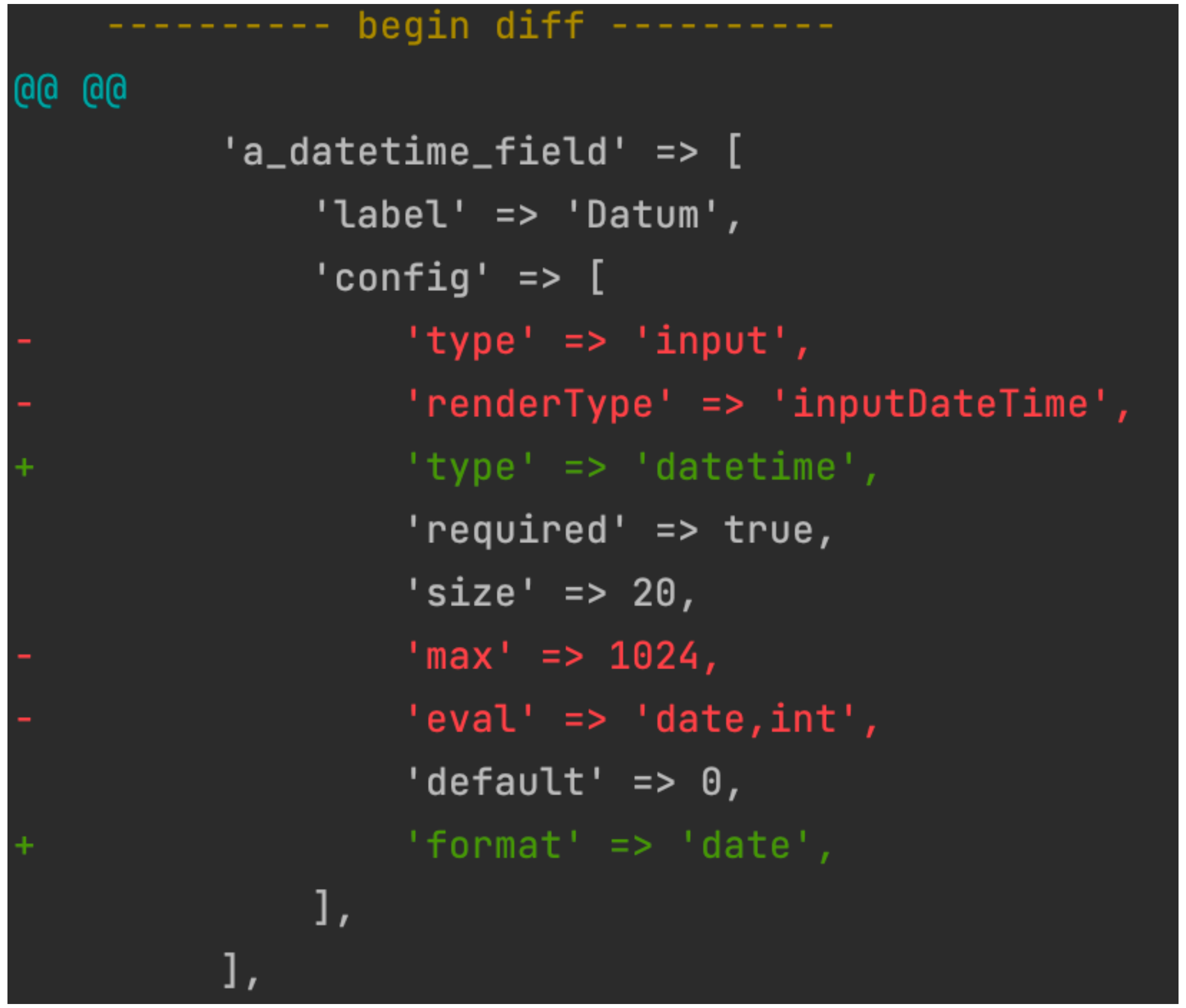
1, 2, … dry-run

```
ddev composer run rector
```

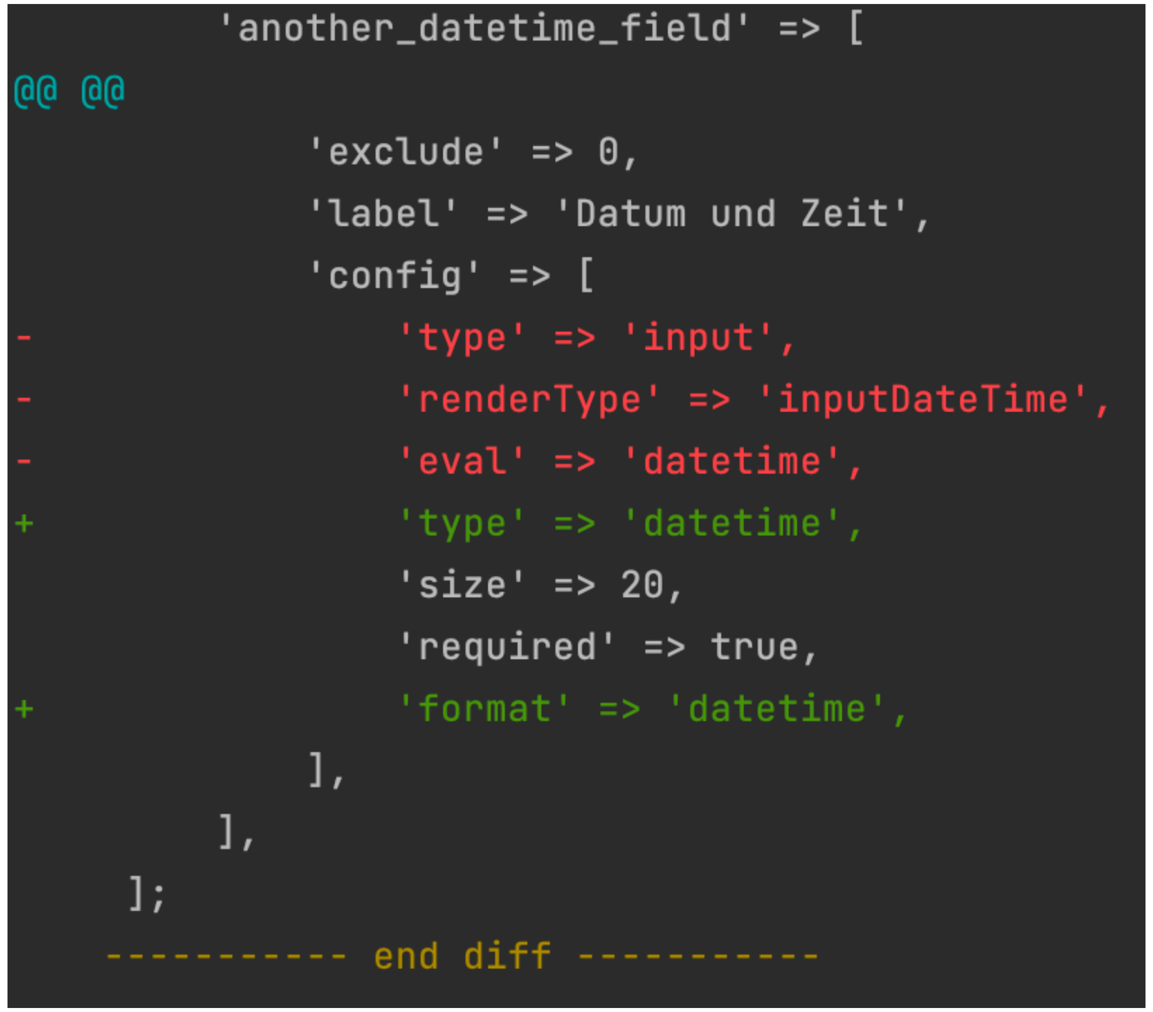


```
> ./vendor/bin/rector process packages --dry-run
 8/8 [▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓] 100%
1 file with changes
===================

1) packages/sitepackage/Configuration/TCA/Overrides/pages.php:6
```



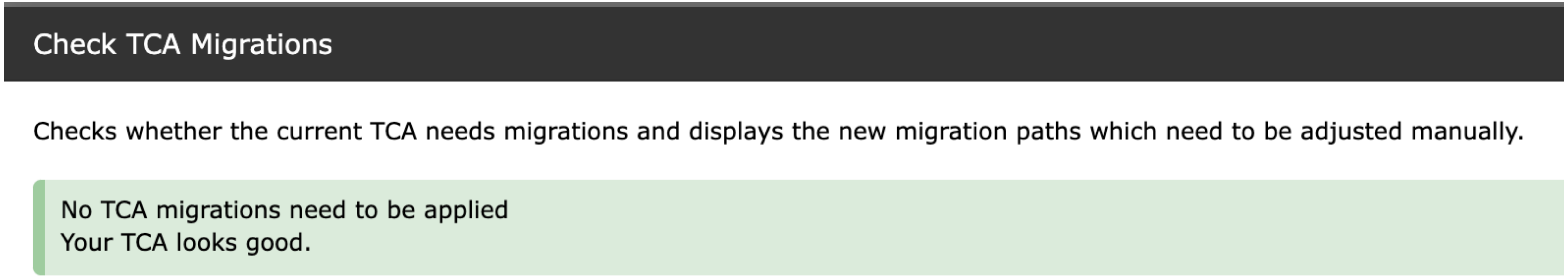
```
    ---------- begin diff ----------
@@ @@
         'a_datetime_field' => [
             'label' => 'Datum',
             'config' => [
-                'type' => 'input',
-                'renderType' => 'inputDateTime',
+                'type' => 'datetime',
                 'required' => true,
                 'size' => 20,
-                'max' => 1024,
-                'eval' => 'date,int',
                 'default' => 0,
+                'format' => 'date',
             ],
         ],
```



```
         'another_datetime_field' => [
@@ @@
             'exclude' => 0,
             'label' => 'Datum und Zeit',
             'config' => [
-                'type' => 'input',
-                'renderType' => 'inputDateTime',
-                'eval' => 'datetime',
+                'type' => 'datetime',
                 'size' => 20,
                 'required' => true,
+                'format' => 'datetime',
             ],
         ],
     ];
    ----------- end diff -----------
```

Und dann gscheit…

```
ddev composer run rector:fix
```



Check TCA Migrations

Checks whether the current TCA needs migrations and displays the new migration paths which need to be adjusted manually.

No TCA migrations need to be applied
Your TCA looks good.

Demo-Projekt: https://github.com/m11r86/tuga-89-rector

Thanks,
bye 👋