

TYPO3 EXT:campus_lingo

© 2024 [Brain Appeal GmbH, Mannheim](#)

This documentation is for EXT:campus_lingo 2.0.0

Last Modified: 2024-11-18

What is EXT:campus_lingo for?

- Auto-translates all records for language fallbacks to default language via Campus Lingo API.
- Respects backend translations.
- Auto-detect updates of default language contents and/or field configuration changes.
- Only re-translates on detection of changes, minimizing requests to the API (and therefore costs) via intelligent caching.

Requirements

- PHP 7.4-8.3
- TYPO3 CMS (Versions 11 LTS and 12 LTS are supported)

How do I get set up?

1. Install Extension TYPO3 via Extension Manager or by composer update
2. Configuration via [Extension Configuration](#)
At least: **Enter your [Campus Lingo API key](#).**
3. Configuration via [Sites backend module](#) (or the site's config.yaml)
At least: **Languages must be configured to fall back to the default language in order to translate them automatically.**
EXT:campus_lingo enriches the languages in the Sites backend module with help depicting configuration problems.
4. Optionally, configure pages that must not be translated via page properties in [page's "Language" register](#).
5. Something isn't working? Have a look at the section [Troubleshooting](#), the [FAQs on academic-hero.de](#) or [contact us](#).

Advanced integration features

For the realization of feature requests that aren't covered by backend interfaces, this document contains a section that is aimed at [TYPO3 integrators or developers](#). You will e.g. find here how to ...

1. [show your visitors, whether a page is automatically translated](#);
2. [customize the automatic translation of content stored in certain tables and fields](#);
3. [automatically translate content saved in FlexForms](#);
4. use `EXT:campus_lingo` [commands](#);
5. [automatically translate an extension's locallang.xlf](#);
6. [translate "arbitrary" content that isn't stored in the TYPO3 database](#).

Support

If you have any questions, please contact us via our [ticket system](#) or your contact person at [Brain Appeal GmbH, Mannheim](#).

In case you don't have a login or contact person you can reach us via [email](#).

Page properties to toggle automatic translation

EXT:campus_lingo adds the form element “Automatic translation” to the page properties on register “Language”.

This form element provides two checkboxes depending on the current activation state:

- **For the current page**

If active: Don't show automatic translated content on this page

If inactive: Enable automatic translated content on this page

- **For subpages of the current page**

If active: Don't show automatic translated content on subpages

If inactive: Enable automatic translated content on subpages

The form element on pages “deeper” in the TYPO3 Pagetree respects settings of their parent pages.

Therefore, (de-)activation for subpages may be activated (or deactivated resp.) for sub-subpages again.

The form element contains an information panel to show, if the automatic translation is activated for the current page and for its subpages.

To avoid confusion, this panel also provides the information of the parent page, where the (de-)activation for the current page is set. A link to edit this parent page directly is also provided.

Configuration

Site Configuration

All languages

- Campus Lingo Language Code `site_language.[].campus_lingo_lang_code`
If set, this code is used for Campus Lingo. Else, the "Two Letter ISO Code" is used.
Examples when using DeepL: EN-GB, EN-US, PT-BR, PT-PT (see [DeepL documentation](#)).

Default language

- Campus Lingo needs "Opt-In" `site_language.0.campus_lingo_opt_in`
If enabled, no automatic translation takes place on this site without configuring the page properties accordingly (see section above).
Automatic translation can be switched on in the page properties for the respective page or subpages.
- Campus Lingo translation service can be deactivated `site_language.0.campus_lingo_translation_service_toggle_enabled`
If enabled, usage of the translation service can be switched off in the page properties' form element "Automatic translation". Automatically translated content in the EXT:campus_lingo cache will still be used. However, no new translations are obtained from the Campus Lingo API.
- Hide untranslated pages `site_language.0.campus_lingo_hidePageIfNotTranslated`
Hides pages that are neither translated manually nor automatically.
If activated, the checkbox "Hide page if no translation for current language exists" is removed from page properties.

Additional languages (i.e. translations)

- Fallback Type `site_language.[].fallbackType`
Must be set to `Strict` OR `Fallback`.
Attention The Fallback Type `Free` mode isn't supported. It isn't planned to support it.

Extension Manager Configuration

Campus Lingo API

- `campusLingoApiKey`

Campus Lingo API key: If you don't have a Campus Lingo API key, visit academic-hero.de/produkte/campus-lingo

- `campusLingoApiEndpoint = https://api.campus-lingo.com/api.php`
API endpoint

- `feature_disableSubscription = 0`

Disable automatic subscription:EXT:campus_lingo by default automatically subscribes its endpoints at Campus Lingo API. If these endpoints aren't accessible (e.g. due to firewall or password protection), you may want turn off this automatic subscription. For the features provided by these endpoints (e.g. automatic updates of glossary entries), you must manually configure those endpoints in Campus Lingo API.

Translation

- `formality = (default|more|less)`

Formality: Sets whether the translated text should lean towards formal or informal language. This feature currently only works for target languages "DE", "FR", "IT", "ES", "NL", "PL", "PT-PT", "PT-BR" and "RU"

- `translateFields_allowLanguageSynchronization`

Translate optional fields: If set, fields that are marked as optionally translatable are automatically translated, too. Technical background: Activating this option will translate fields with TCA "allowLanguageSynchronization". If not activated, those fields aren't translated.

- `dontTranslateWrap = ~~~`

Don't translate wrap: If set, this string can be used to "wrap" text or words (i.e., prepend and append the string) that won't be translated automatically. The wrap is removed before a page is sent to cache or outputted. E.g.: The default setting ~~~ will prevent the translation of ~~~ This text isn't translated~~~ and will eventually be outputted as This text isn't translated.

- ignoreTags = nottranslate

Ignore tags: Content within these tags is to be ignored by automatic translation.

Frontend

- translationInformationPlaceholder = <!-- CAMPUS_LINGO_AUTOTRANSLATION_INFORMATION -->

Translation information placeholder: Placeholder for templates, replaced with information if a page is (partially) translated. If some auto translated content is on the current page, this placeholder will be replaced with information about auto translated content (for modifying the HTML, see TypoScript lib.campusLingo.translationInformation).

- excludeTablesFromAutotranslationRegisterCount = pages, sys_file_metadata
Excluded tables from auto-translation register count: Comma separated list of database tables that aren't counted in TSFE register CAMPUS_LINGO_AUTOTRANSLATED_CONTENT_RECORD_COUNT. The value of this register is used to determine the content outputted in the translation information placeholder (see above).
- strict_excludeTablesFromFallback
Tables that mustn't fall back: Comma separated list of database tables that aren't allowed to fallback for auto translation when language has Fallback Type set to "Strict".

Backend

- feature_disableChoicesNoTransS = 0

Deactivate "disable translation service": Deactivates the possibility to configure the usage of the "Campus Lingo translation service can be deactivated" in sites configuration. If checked, disabling the usage of translation service while still using EXT:campus_lingo translation cache isn't possible at all. If not checked, disabling the usage of translation service while still using EXT:campus_lingo translation cache via page properties may be enabled via sites configuration.

Debug

- debugOutputTimes

Output times: Outputs summed time in seconds needed for API calls and getting translations from cache. Times are outputted as HTML comment before the closing </body>.

- `debugAdditionalTranslationInformation`
Debug Output of additional translation information: Outputs additional information about translated content as HTML comment below the Translation Information Placeholder.
- `debugLogRepeatingMessages`
Log repeating messages to System > Log: If enabled, the same log message of `campus_lingo` is written to the table `sys_log` (System > Log) on each occurrence. If disabled, the same log message is only written to `sys_log`, if its last occurrence is older than an hour.

Development

Please see section [Translate locallang.xlf files](#).

Troubleshooting / incompatible configurations

If (some) pages aren't translated automatically after installing EXT:campus_lingo, ensure to disable the following settings: Under certain circumstances, they may hide untranslated pages before campus_lingo is able to translate them.

Disable inverse control of hidden translations

Ensure that the following setting is not set, since campus_lingo isn't fully tested with this setting turned on:

```
$GLOBALS['TYPO3_CONF_VARS']['FE']['hidePagesIfNotTranslatedByDefault'] = 1;
```

See [Inverse control of hidden translations at "Frontend Localization Guide" on docs.typo3.org](#)

Field "Localization" of pages

Ensure that the field "Localization" of table pages `pages.l18n_cfg` isn't set to 2, i.e. "Hide page if no translation for current language exists" shouldn't be checked. Furthermore, ensure that `pages.l18n_cfg = 2` isn't set by default (e.g. by `TCAdefaults.pages.l18n_cfg = 2`).
campus_lingo works best without this setting, since it isn't fully tested for compatibility with `l18n_cfg=2`.

For Developers & Integrators

Output information about auto-translated content in frontend

Insert the following placeholder to your template

```
<!-- CAMPUS_LINGO_AUTOTRANSLATION_INFORMATION -->
```

If there is auto translated content on the current page, the placeholder will show the information, that (some) content on the current page is auto translated.

If no auto translated content is used, the placeholder is removed.

To modify HTML and contents shown by the placeholder, modify the TypoScript object `lib.campusLingo.translationInformation`.

TSFE Registers

EXT:campus_lingo sets the following TSFE register values:

- (int)
CAMPUS_LINGO_AUTOTRANSLATED_CONTENT_RECORD_COUNT
Number of auto translated records used on current page (expect page records)
- (int)
CAMPUS_LINGO_MANUAL_TRANSLATED_CONTENT_RECORD_COUNT
Number of manually translated records used on current page (expect page records)
- (bool) CAMPUS_LINGO_PAGE_IS_AUTOTRANSLATED
Set if page record is auto translated
- (bool) CAMPUS_LINGO_PAGE_USES_AUTOTRANSLATED_PAGES
Set if any other page record is auto translated (e.g. for menus shown on the page)
- (array) CAMPUS_LINGO_AUTOTRANSLATED_CONTENT_RECORDS
An array of elements in format `table.uid` representing auto translated records
- (array)
CAMPUS_LINGO_MANUAL_TRANSLATED_CONTENT_RECORDS
An array of elements in format `table.uid` representing manually translated records

Customize tables and fields

1) Add option to disable auto-translation for records of a certain table `your_table`

In Configuration/TCA/Overrides/your_table.php add

```
if (\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::isLoaded('campus_lingo')) {
    \BrainAppeal\CampusLingo\Utility\TCA\ApiRecordTranslations::addAutoTranslationSettings(
        'your_table'
    );
}
```

2) Enforce auto-translation of a certain field `your_table.your_field`

In Configuration/TCA/Overrides/your_table.php add

```
$GLOBALS['TCA']['your_table']['columns']['your_field']['autotranslate_enforce'] = true;
```

3) Prohibit auto-translation of a certain field `your_table.your_field`

Beware: `autotranslate_enforce` has priority.

In Configuration/TCA/Overrides/your_table.php add

```
$GLOBALS['TCA']['your_table']['columns']['your_field']['autotranslate_forbidden'] = true;
```

4) Preprocess/Postprocess records - PSR-14 BrainAppeal

As an example, have a look at the EventListener `BrainAppeal\CampusLingo\EventListener\CampusLingo\TranslationServiceProcessRowFlexFormEventListener` that implements auto-translation of FlexForm properties.

Following this example, this Event should enable you to implement the auto-translation of data stored sort of “unstructured” to database as being the case for the FlexForms’ XML.

5) Set/modify TCA on the fly: PSR-14 BrainAppeal

As an example, have a look at the EventListener `BrainAppeal\CampusLingo\EventListener\CampusLingo\TranslationServiceModifyTCAEventListener`

The following snippets ready for copy&paste ensure auto-translation of fields without the need of modifying (or defining) `$GLOBALS['TCA']` for a table.

`your_extension/Configuration/Services.yaml`

```
services:
  Vendor\YourExtension\EventListener\CampusLingo\TranslationServiceModifyTCAEventListener:
    tags:
      - name: event.listener
        identifier: 'yourExtension/campusLingo/translationServiceModifyTCA'
        event: BrainAppeal\CampusLingo\Event\TranslationServiceModifyTCAEvent
```

`your_extension/Classes/EventListener/CampusLingo/TranslationServiceModifyTCAEventListener.php`

`<?php`

```
namespace Vendor\YourExtension\EventListener\CampusLingo;
use BrainAppeal\CampusLingo\Event\TranslationServiceModifyTCAEvent;

class TranslationServiceModifyTCAEventListener
{
    /**
     * sets or modifies TCA for the auto-translation process on table 'your_table'
     * to ensure that fields 'field_a' and 'field_b' are auto-translated
     * @param TranslationServiceModifyTCAEvent $event
     */
    public function __invoke(TranslationServiceModifyTCAEvent $event): void
    {
        if ('your_table' === $event->getTable()) {
            $fieldsToTranslate = ['field_a', 'field_b'];
            $tcaOverride = [];
            foreach ($fieldsToTranslate as $fieldToTranslate) {
                $tcaOverride['columns'][$fieldToTranslate]['autotranslate_enforce'] = true;
            }
            $event->setTCA(array_replace_recursive(
                ($event->getTCA() ?? []),
                $tcaOverride
            ));
        }
    }
}
```

6) Configure auto-translation for FlexForm fields

To enable auto-translation of content stored in FlexForms, additional configuration is required. You must configure, which properties of a FlexForm shall be auto-translated.

Configuration is registered in TCA. For ease of use, the following API method exists which will ensure the needed TCA:

In Configuration/TCA/Overrides/your_table.php add

```
if (\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::isLoaded('campus_lingo')) {
    \BrainAppeal\CampusLingo\Utility\TCA\ApiRecordTranslations::addAutoTranslationForFlexFormField(
        $table = 'your_table',

        // field containing FlexForm data (e.g. pi_flexform for table tt_content)
        $fieldName = 'pi_flexform',

        // array of FlexForm properties to auto-translate
        // or
        // "compact notation": An associative array with
        // - keys representing the dataStructureKey
        // - values of keys are arrays of FlexForm properties to auto-translate
        $flexFormProperties = [
            'settings.yourFlexformKey'
        ],

        // $dataStructureKey isn't needed for the compact notation.
        // It is optional recommended for specific $flexFormProperties
        // auto-translation behaviour depending on the given data structure.
        $dataStructureKey = '*'
    );
}
```

Examples

The following example for the infamous EXT:powermail will auto-translate emails to sender (“Mail to User”) and the thank you text (“Text on submit page”):

```
if (\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::isLoaded('campus_lingo')) {
    \BrainAppeal\CampusLingo\Utility\TCA\ApiRecordTranslations::addAutoTranslationForF
        lexFormField(
            'tt_content',
            'pi_flexform',
            [
                'settings.flexform.sender.subject',
                'settings.flexform.sender.body',
                'settings.flexform.thx.body'
            ],
            '*',powermail_pi1'
        );
}
```

Using the compact notation (the result is equivalent with the above)

```
if (\TYPO3\CMS\Core\Utility\ExtensionManagementUtility::isLoaded('campus_lingo')) {
    \BrainAppeal\CampusLingo\Utility\TCA\ApiRecordTranslations::addAutoTranslationForF
        lexFormField(
            'tt_content',
            'pi_flexform',
            [
                '*',powermail_pi1' => [
                    'settings.flexform.sender.subject',
                    'settings.flexform.sender.body',
                    'settings.flexform.thx.body'
                ]
            ]
        );
}
```

Commands

campuslingo:delete-manual-translations

The console command `campuslingo:delete-manual-translations` can be used to delete existing editorial translations for pages or subtrees. This command may be handy in order to delete certain editorial translations that are henceforth ought to be translated automatically (e.g. due to “stuck” or outdated editorial translations).

Use `./vendor/bin/typo3 campuslingo:delete-manual-translations --help` for more information.

Translate locallang.xlf files

Prerequisite: The applicationContext of the TYPO3 system must be set to development.

1. Open the Extension Configuration of EXT:campus_lingo and switch to register “Development”.
2. Enter the keys of the extensions, for which you want automatically translate the locallang.xlf file.
3. If the target locallang.xlf file exists (e.g. de.locallang.xlf for the German translation file) and you want to overwrite it, check the corresponding checkbox.
4. Switch to the backend module “Maintenance” and click to “Manage Languages”
5. You may need to add the target language if it isn’t already there.
6. Download the language of the extension(s) of step 2. Or simply click “Update all”.
7. The translations for the extensions of step 2 are fetched based on EXT:
[extKey]/Resources/Private/Language/locallang.xlf,
and are saved to EXT:[extKey]/Resources/Private/Language/
[target_language].locallang.xlf

Your extension is now ready for use in the target language.

Translation of arbitrary content

The API Method `\BrainAppeal\CampusLingo\Service\LanguageOverlayService::autoTranslateRow` allows automatic translation of literally “arbitrary” content given as an associative array (e.g. some database row from an external database).

Features

- Respects the requested language on frontend request.
- Uses site’s default language as source language by default. The source language may be given as parameter, allowing for automatic translation of content given in languages different from the site’s default language.

Note: This allows for automatic translation to the site’s default language: If the passed source language is different from the site’s default language, the API method will even trigger automatic translation, if the default language is requested.

- Respects (de)activation of the auto-translation via page properties.
- The placeholder `<!-- CAMPUS_LINGO_AUTOTRANSLATION_INFORMATION -->` reflects automatic translations via this API method.
- Multiple automatic translation of the same content is avoided by using the translation cache of `EXT:campus_lingo`. Changes of the same content source trigger automatic retranslation, whereas the content source is identified by two parameters (id and identifier).
- On errors, the configured language fallbacks of the site’s language are respected.

Configuration

Apart from the method’s parameters, you may need (or want) to configure the automatic translation by establishing an event listener for the event `\BrainAppeal\CampusLingo\Event\TranslationServiceModifyTCAEvent` (see above)

Example

```
/**
 * @param array $row
 * @param string $identifier
 * @param string $idField
 * @return array
 */
function autoTranslateRow(array $row, string $identifier, string $idField = 'id'): array
{
    /**
     * for all parameters:
     * @see \BrainAppeal\CampusLingo\Service\LanguageOverlayService::autoTranslateRow
     *
     * you may need (or want) to configure, which keys of $row are automatically
     * translated
     * by establishing an event listener for
     * @see \BrainAppeal\CampusLingo\Event\TranslationServiceModifyTCAEvent
     */
    return \BrainAppeal\CampusLingo\Service\LanguageOverlayService::getInstance()-
        >autoTranslateRow(
            $row,
            $identifier,
            $idField
        );
}

$translatedRow = autoTranslateRow(
    [
        'id' => 12345,
        'textProperty' => 'Please translate this text automatically.'
    ],
    'someIdentifierToEnsureThatIdIsUnique',
    'id'
);
```