



TYPO3 extension:  
we\_cookie\_consent 1.3.0  
Documentation

**websedit AG**

Seestr. 35  
88214 Ravensburg

Telefon: +49 751 35 41 04-0  
Telefax: +49 751 35 41 04-42

[extensions@websedit.de](mailto:extensions@websedit.de)  
[www.websedit.de](http://www.websedit.de)

## 1. Introduction

### 1.1. How is the extension?

Due to the legal decision of the European Court of Justice (ECJ), website operators must obtain the active consent of users to the use and storage of cookies or scripts on the user's computer. It is therefore not sufficient to simply point them out or to use them before actively agreeing to them. With the `we_cookie_consent` Extension you can integrate cookies or scripts on your website in a data protection compliant way.

### 1.2. Properties

- Activates cookies only after approval
- Data protection-compliant content solution
- Providing the user settings
- Simple Google Tag Manager export
- Provision of cookie information in the privacy policy
- Preconfigured modules for Google Analytics, Google Tag Manager and Facebook
- Arbitrary expandability of tracking scripts

### 1.3. Compatibility

TYPO3 7.6

TYPO3 8.7

TYPO3 9.5

TYPO3 10.4

### 1.4. Demo

<https://consent.websedit.de/>

## 2. Setup

### 2.1. Requirements

The extension uses jQuery in the frontend. If the jQuery library is already in use, no adjustment is necessary. Otherwise, you must include jQuery. Future versions of the extension should work without jQuery.

### 2.2. Installation

The extension is installed like any TYPO3 extension. There are 4 sources of supply.

### 2.3. Get the extension

#### 2.3.1. About the Extension Manager in the TYPO3 Backend

- 1) As a user with admin rights, navigate to the **Extensions** module
- 2) In the select box, navigate to the **Get Extensions** sub-module.
- 3) Click on **"Update now"**
- 4) Enter the key **we\_cookie\_consent** in the search field
- 5) Click the **cloud icon** in the hit list to import and install the extension.

#### 2.3.2. From the TYPO3 Extension Repository. (<https://extensions.typo3.org/>)

- 1) Search for the key **we\_cookie\_consent**
- 2) Download the **ZIP file**
- 3) **Unpack** the ZIP file and move the content into the **TYPO3 extension directory** (`typo3conf/ext/`)

- 4) Navigate in the TYPO3 Backend as user with admin rights into the module **Extensions**
- 5) Look for the extension **we\_cookie\_consent**
- 6) Click on the **cube icon with the plus** to install the extension

#### 2.3.3. Via Composer

The installation via Composer is possible via the TYPO3 own repository (<https://composer.typo3.org/satis.html>)

Three steps are necessary:

- 1) Publish the TYPO3 repository in composer.json:  
composer config repositories.0 composer <https://composer.typo3.org/>
- 2) Install the extension with the package name typo3-ter/we-cookie-consent  
composer req typo3-ter/we-cookie-consent
- 3) Activate the extension via the extension manager in the backend

#### 2.3.4. Via GIT

– Needs to be done. –

### 2.4. Integration

#### 2.4.1. Installation of the supplied static template

- 1) As a user with admin rights, navigate to the **Template** module
- 2) Navigate in the page tree **to the page where you have stored your templates**
- 3) In the select box, select the sub-module **Info/Modify** (usually preselected)
- 4) Switch to the Edit Mode by clicking the **Edit the whole template record** button
- 5) Click the **Includes** tab
- 6) Select Cookie Consent (**we\_cookie\_consent**) in the Available Objects area on the right side
- 7) This moves to the left to the **Include Static (from extensions)**
- 8) Click **Save**
- 9) The integration is completed

#### 2.4.2. Linking the record storage locations

- 1) As a user with admin rights, navigate to the **Template** module
- 2) Navigate in the page tree to the **root page (globe)**
- 3) In the select box, select the **Constant Editor** sub-module
- 4) Select **PLUGIN.TX\_WECOOKIECONSENT\_PI1** in the selectbox
- 5) Enter the **UID of your privacy page** as the Privacy Page
- 6) Enter the **UID of the Cookie Consent data set folder** under Default storage PID, see chapter 3

### 3. Configuration

#### Creating a Record Folder

- 1) In TYPO3, navigate to the **List** module
- 2) Create a **new folder** in the page tree
- 3) Give the folder the name **Cookie Consent**
- 4) Activate the folder

#### 4. Maintenance

##### 4.1. Find out which cookies are used on your website

- 1) For example, you can visit the URL <https://www.cookiemetrix.com/> and analyse your website. Soon we will also offer a cookie analysis tool.
- 2) Check your cookies and identify the cookies that need to be marked

##### 4.2. Maintenance of the service without Google Tag Manager

###### 4.2.1. Maintenance of an integrated service

The `we_cookie_consent` extension offers a simplified integration for certain services. These are Google Analytics, Google Analytics Universal, Facebook Pixel and Matomo. For these scripts you only need the ID of the service and not the complete tracking code.

- 1) Navigate to the **List** module
- 2) Click on the **Cookie Consent folder** in the page tree
- 3) To create a record for the first time, click the "+" in the content area. Select under the item GDPR: Cookie Consent --> Service.
- 4) To create all further records, click on the "+" next to Provider
- 5) Under "**Provider**" select the service you want to integrate (Google Analytics, Google Analytics Universal, Facebook or Matomo)
- 6) Enter a name under **Service or Script Title**
- 7) In the **Description field**, add a description (explanation of how to use the service or script)
- 8) Navigate to the Cookies tab and click on + **Add another cookie**
- 9) Add the **cookie name**, **cookie purpose** and **cookie lifetime** according to the GDPR specifications (In the Cookie lifetime input field, you can enter minutes, days or hours. It is important that you write the time unit behind it)
- 10) In the **Settings tab**, make the desired setting on how the script should be loaded
- 11) In the **Authentication tab**, insert the script for the desired service
- 12) Click **Save**

Your service is now displayed in the Cookie Consent overview

###### 4.2.2. Maintenance of an individual service

- 1) Navigate to the **List** module
- 2) Click on the **Cookie Consent folder** in the page tree
- 3) To create a record for the first time, click the "+" in the content area. Select under the item GDPR: Cookie Consent Service.
- 4) To create all further records, click on the "+" next to Provider
- 5) Under "Provider", select **Other Service**
- 6) Enter a name under **Service or Script Title**
- 7) In the **Description field**, add a description (explanation of how to use the service or script)
- 8) Navigate to the Cookies tab and click + **Add another cookie**
- 9) Add the **cookie name**, **cookie purpose** and **cookie lifetime** according to the DS-GVO specifications
- 10) In the **Settings tab**, make the desired setting on how the script should be loaded
- 11) In the **tab Development** insert the script for the desired service
- 12) Click **Save**

Your service is now displayed in the Cookie Consent overview

#### 4.3. Maintenance of the services with Google Tag Manager

##### 4.3.1. Creating the Google Tag Manager in the Cookie Consent Extension

- 1) Navigate to the **List** module
- 2) Select **Cookie Consent** in the page tree
- 3) Click on the "+" to create a new record
- 4) Under "Provider" select **Google Tag Manager**
- 5) Enter a name under **Service or Script Title**
- 6) Add a description in the **Description field**
- 7) Navigate to the Cookies tab and click on + **Add another cookie**
- 8) Add a **cookie name, cookie purpose** and **cookie lifetime**
- 9) In the **Settings tab**, make the desired setting on how the script should be loaded
- 10) In the **Authentication tab**, insert the **Google Tag Manager ID** in the format **GTM-XXXXXXX**
- 11) Click **Save**

##### 4.3.2. Create a service or script in the extension, which is integrated via the Google Tag Manager

- 1) Navigate to the **List** module
- 2) Select **Cookie Consent** in the page tree
- 3) Click on the "+" to create a new record
- 4) Under "Provider" select **Google Tag Manager – Service**
- 5) Insert a name in the **Title of service or script field**
- 6) Add a description in the **Description field**
- 7) Navigate to the Cookies tab and click on + **Add another cookie**
- 8) Add a **cookie name, the cookie purpose** and the **cookie lifetime** of the service
- 9) In the **Settings tab**, make the desired setting on how the script should be loaded
- 10) In the **Authentication tab**, insert the **ID of the desired service**
- 11) Navigate to the tab "**Google Tag Manager**". **Fill out all fields** here. You will need these names again in the Google Tag Manager interface
- 12) Click **Save**

##### 4.3.3. Create the service in the Google Tag Manager

(<https://www.google.com/intl/de/tagmanager/>)

- 1) Log in (<https://www.google.com/intl/de/tagmanager/>)
- 2) Click **Variables** in the left pane
- 3) Create a **new custom variable** in Google Tag Manager
- 4) Give the variable a **name**
- 5) Select the variable type **Data layer Variable**
- 6) Give the **data layer variable** the **same name** that you selected in the **Cookie Consent Extension** for the corresponding service.
- 7) **Save** the variable
- 8) Create a **new trigger** in Google Tag Manager
- 9) Give the trigger a **name**
- 10) Select the trigger type **Custom event**
- 11) Give the **trigger** the **same event name** that you selected in the **Cookie Consent Extension** for the corresponding service.
- 12) Select **Some Custom Events** at "**This Trigger fires on**"

- 13) **Select** at "Fire this trigger when an event occurs and all these conditions are true", **the created variable for this service equals true**
- 14) Create a **tag** for the corresponding service and select the **corresponding trigger** that you have previously created
- 15) Click on the "**Submit**" button in the top left corner of the Google Tag Manager interface
- 16) Enter a version name and a version description
- 17) In the Google Tag Manager interface, click on "**Publish**" in the upper left corner

#### 4.3.4. Use of the Google Tag Manager export function within the Cookie Consent Extension - Needs to be done. –

5. Adding the cookie overview in the privacy policy
  - 1) Navigate to the module **Page**
  - 2) Navigate in the page tree to the **page with your privacy policy**
  - 3) Create a new content element by clicking on **+ Content**
  - 4) In the "Create New Content Element window", navigate to the **Plugins tab**
  - 5) Click **Cookie Liste**
  - 6) Click on the **Plugin tab**
  - 7) In the Services & Scripts area, click on the **folder icon** on the left
  - 8) The data record selector opens
  - 9) Select the **folder** where you created your cookies
  - 10) Select the desired **record**
  - 11) Under **Provider**, add any cookies that you want to be displayed later in the privacy statement
  - 12) Click **Save** after you have added all cookies
  
6. Additional
  - 6.1. Adjusting the TYPO-Script parameters
    - 1) **Default Storage:** The ID of the data record folder is stored here. See point 2.4.2.
    - 2) **Privacy Page:** In the input field Privacy Page you must enter the UID of your privacy page. See point 2.4.2.
    - 3) **Style Prefix:** In the Style Prefix input field you can adjust the positioning of the cookie banner.  
Possible entries:  
Display at the bottom of the screen: `klaro we_cookie_consent` (default)  
Display at the top of the screen: `klaro we_cookie_consent notice--top`  
Display in the middle with the buttons below each other: `klaro we_cookie_consent notice--center`  
Display in the middle with the buttons below and next to each other: `klaro we_cookie_consent notice--center-floated`
    - 4) **Storage Method:** Here you can set whether the user settings should be stored in a cookie or in the local storage of the browser. (default: cookie)
    - 5) **Cookie Name:** In the input field Cookie Name you can change the name of your content cookie. This was designed for the use of different language versions, with different services provided via directories and not via different Top Level Domains.

- 6) Lifetime: In the Lifetime input field, you can determine the validity period of the cookie. After this time, the user will be asked again to accept the cookies. The default setting is 365 days.
- 7) State: In the input field State you can set whether your cookies are activated or deactivated in the cookie banner. Possible entries: false (default), true (Not DS-GVO compliant!)
- 8) Must Consent: In the input field Must Consent you can set whether users can use your website without interaction with the cookie banner. Possible entries: false (default), true (Not DS-GVO compliant!)
- 9) Hide Decline all: Hier können Sie einstellen, ob der „Ablehnen“ Button angezeigt werden soll. (Standardmäßig wird der Button ausgegeben)
- 10) Hide Learn more: Hier können Sie einstellen, ob der Button „Einstellungen bearbeiten“ angezeigt werden soll. (Standardmäßig wird der Button ausgegeben)
- 11) Language (obsolete): Dieser Wert sollte nicht verändert werden. In einer der nächsten Versionen wird diese Einstellung entfernt. Weitere Informationen zur Steuerung der Sprachen und der Mehrsprachigkeit siehe Kapitel 6.2.

## 6.2. Localisation

Do not enter a value for the Language constant editor settings and/or leave them set to Default (en).

All labels are stored in the locallang.xlf files. These can be overwritten e.g. by TypoScript. This could look like this:

```
#Override locallang.xlf Labels
config.tx_extbase._LOCAL_LANG {
    default {
        klaro\consentNotice\description = My custom Cookie message
    }
    de {
        klaro\consentNotice\description = Mein eigener Cookie Text
    }
}
```

## 6.3. Adaptation to your design

- CSS... Needs to be done. –

## 7. Don't forget!

After maintaining the Cookie Consent Extension, make sure that each tracking code appears only once on your website. If the case occurs that you have maintained the same tracking code several times on your site, the data collection may be falsified.

## 8. Acknowledgement

We use open source software.

We would like to thank the people involved in the Klaro! project!

<https://github.com/KIProtect/klaro>