Creating a content class

For CMS developers, a task from start to finish. Create a class in the administration, create content with it, give it a full view template through an override, let an editor role create it, and carry it to the next...

Source file: `doc/eng-us/help/cms-developers/creating-a-content-class.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/cms-developers/creating-a-content-class.md

This page is for developers who have a content model on paper and now build the first class. It follows one class, an event, through every step a real project needs: the class, a first object, a template, permissions and a package. Do it on a development installation first (Development environment).

Contents

1. What you need
2. Step 1: create the class
3. Step 2: create an object
4. Step 3: give it a full view template
5. Step 4: let editors create it
6. Step 5: carry the class to other installations
7. Checks
8. When something goes wrong

What you need

- An administrator account on a development installation.
- A design extension of your own for templates, for example `extension/exp_mysite` (Your own design extension).
- The class written down: identifier `exp_event`, attributes `title` (Text line, required), `intro` (Text block), `body` (XML block), `start_date` (Date and time, required), `image` (Image).

Step 1: create the class

1. In the administration, open **Setup** > **Classes** (`/class/grouplist`).
2. Open the class group **Content** and press **New class**.
3. Enter the name `Event` and the identifier `exp_event`.
4. Set **Object name pattern** to `<title>`. Leave **URL alias name pattern** empty: the URL is made from the name.
5. Leave **Container** unticked: events have no children.
6. Under the attribute list, choose **Text line** and press **Add attribute**. Name it `Title`, identifier `title`, tick **Required** and **Searchable**.

7. Add the other attributes the same way: `Intro` (Text block, `intro`), `Body` (XML block, `body`), `Start date` (Date and time, `start_date`, required), `Image` (Image, `image`).
8. Press **OK**. The class is stored and appears in the group's list.

Until you press **OK** the class is a temporary version that nobody else sees; **Cancel** throws it away. The content model and editing content, section 2 has the same steps with screenshots of the fields.

Step 2: create an object

1. Open **Content structure**, select the folder the events go in, choose **Event** in the create menu and press **Create here**.
2. Fill the fields and press **Send for publishing**.
3. Open the new page on the public siteaccess. Without a template of your own it is drawn by the design's generic full view, which lists the attributes.

Fill a few events with real content now: they are your test data for templates and fetches.

Step 3: give it a full view template

Add an override rule to your design extension, for the public siteaccess only. In `extension/exp_mysite/settings/siteaccess/<public siteaccess>/override.ini.append.php`:

INI

```
<?php /* #?ini charset="utf-8"?

[exp_mysite_full_event]
Source=node/view/full.tpl
MatchFile=full/exp_event.tpl
Subdir=templates
Match[class_identifier]=exp_event
Priority=10
*/ ?>
```

Then the template, `extension/exp_mysite/design/<your design>/override/templates/full/exp_event.tpl`:

Template

```
<article class="exp-event">
    <h1>{$node.name|wash()}</h1>
    <p class="exp-event__date">
        {$node.data_map.start_date.content.timestamp|l10n( 'shortdatetime' )}
    </p>
    {if $node.data_map.image.has_content}
        {attribute_view_gui attribute=$node.data_map.image image_class='large'}
    {/if}
    {if $node.data_map.intro.has_content}
        <p class="exp-event__intro">{$node.data_map.intro.content|wash()}</p>
    {/if}
    {attribute_view_gui attribute=$node.data_map.body}
</article>
```

Styles go in your design's stylesheet, loaded through `[StylesheetSettings]`, never in the

template (No inline CSS). Clear the caches that hold settings and override lookups, then reload the page:

Shell

```
php bin/php/ezcache.php --clear-tag=ini --allow-root-user
php bin/php/ezcache.php --clear-id=template-override,content --allow-root-user
```

More on rules: Templates and overrides for CMS developers.

Step 4: let editors create it

1. Open **Users** > **Roles and policies** and edit the editors' role (or create one).
2. **New policy**: module `content`, function `create`. Limit **Class** to `Event` and **Subtree** (or **Node**) to the events folder.
3. Add `content/edit` limited to `Event` and **Owner** if editors should only change their own events.
4. Save and assign the role to the editors' user group.

Sign in as a test editor (with an address at `example.invalid`) and check that **Event** is offered in the events folder and nowhere else. Details: Roles, policies and unactivated users.

Step 5: carry the class to other installations

The class now exists only in this database. Before you build further, put it into files:

1. **Setup** > **Packages** > **Create new package** > **content class export**.
2. Select `Event`, then enter a name, summary, version and license. The package is written to the local repository.
3. Export the package file and commit it to your project, or build a site package that installs it (Site packages and installers).

Templates and settings are already files in your extension; with the package, a new installation gets the whole feature from your repository. Packages explains install and upgrade of packages.

Checks

- ☐ The class has the identifier you planned; every attribute identifier is lower case and reused where it should be.
- ☐ An object publishes, and its page shows your template (with `ShowUsedTemplates` on, the debug output lists `full/exp_event.tpl`; Debug output and logs).
- ☐ A test editor can create events in the right folder only.
- ☐ The class is in a package in your repository.

When something goes wrong

Symptom	Cause and fix
The new template is not used	the rule is in another siteaccess's file, the INI cache was not cleared, or a rule with a lower `Priority` matches first (Override ordering)
The page is empty after adding the template	the template-override cache: clear `template-override` and `content`
"Event" is not in the create menu	the parent is not a container, or the role has no `content/create` for the class there
An image attribute shows nothing	the image alias in `image_class` does not exist in `image.ini` (Images and image aliases)

Related pages

- Content classes and attributes
- View modes
- Tutorial: override a full view: the designer's version
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.