



Creating an Extension :: Contao

docs.contao.org/dev/getting-started/extension/

☆ ⓘ 🔒 🔥 | 

Contao

Developer Documentation

Search...

Getting Started

Initial Setup

Core Concepts

Starting your Development

Adjusting the DCA

Changing Translations

Implementing Hooks

Content Elements & Modules

Creating an Extension

Framework

Reference

Guides

Internals

Contao Developer Documentation > Getting Started > Creating ...

# Creating the Bundle

Now it is time to do some ground work for the extension:

1. Create a bundle class.
2. Create a Contao Manager Plugin to load the bundle within a Contao Managed Edition.
3. Configure the `composer.json` for the Contao Manager Plugin.

Creating the bundle class is simple enough. The name of the bundle class can be freely choosen - typically it will have the same name as your top-level subnamespace, or even a combination of your complete top-level namespace. For example:

```
// src/ContaoExampleBundle.php
namespace SomeVendor\ContaoExampleBundle;

use Symfony\Component\HttpKernel\Bundle\Bundle;

class ContaoExampleBundle extends Bundle
{
}
```

The bundle class can be empty, but could contain additional bundle configurations (see

Models :: Contao Developer Do

docs.contao.org/dev/framework/models/

Contao

Developer Documentation

Search...

Getting Started

Framework

Data Container Array

Models

Collections

Custom Models

Back end modules

Caching

Content Elements

Cron

Front End Modules

Hooks

Insert Tags

Manager Plugin

Contao Developer Documentation > Framework > Models

# Fetching a row

Each class extending `Model` has some *static* methods you can use:

```
use Contao\PageModel;

// get a record by its primary key
$page = PageModel::findByPk(2);

// find a record by ID or alias (the parameter you pass can be both)
$page = PageModel::findByIdOrAlias('index');

// find the record where adminEmail is "admin@example.com"
$page = PageModel::findOneBy('adminEmail', 'admin@example.com');

// count the records where language is "de"
$total = PageModel::countBy('language', 'de');
```

If no row in the table matches your query, the return value is `null`.

Also, `Model` implements the `__callStatic()` method from PHP. This way you can attach the field name of the table to the method name and omit the first parameter.

At the moment, the methods `findOneBy()`, `findBy()` and `countBy()` are supported: