# 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

# Data Container Arrays & Models

*Data Container Arrays* (*DCA* for short) are a key concept of the Contao framework. The management of any record within the Contao CMS is usually described via so called Data Container Arrays. It is possible to create custom DCAs as well as extending or altering existing ones.

*Models* are object representations of any DCA record, while *Collections* are object representations of a collection of such Models. Models can be used to create, load and change these records. They usually also provide convenient functions for loading records according to a specific set of parameters.

# Front End Modules

Front end modules are used for any type of complex functionality on specific pages or specific places on a page of your website managed by Contao. Simple examples are login forms, displaying news entries, showing a navigation etc.

# Content Elements

While any data record can hold information for content to be displayed in the front end, another key concept of Contao is the usage of Content elements for arbitrary and more complex content. The content of regular

docs.contao.org/dev/getting-started/starting-development/

# Contao

### Developer Documentation

🔍 Search...                              ✕

# Structure

After a fresh install of Contao, your project will have a certain initial file & directory structure (which is similar to the structure of a pure Symfony project using the `symfony/skeleton` for example).

| File/Directory | Explanation |
| --- | --- |
| `assets/` | JavaScript and CSS assets of the Contao framework and third parties. |
| `config/` | Application configuration files. |
| `files/` | Public or protected files manged by Contao's file manager. |
| `system/` | Legacy folder for Contao 3 compatibility. |
| `templates/` | Customized Contao & Twig templates. |
| `var/` | Transient files like the application cache and log files. |
| `vendor/` | Composer's vendor folder containing all dependencies (including Contao). |
| `web/` | Public entry points; contains symlinks to other public ressources. |
| `composer.json` | `composer.json` of your project defining your dependencies and autoloading. |