

Field Type Tutorial

About Field Types

Field Types are the most granular building blocks for content managed by eZ Platform. The system comes with about 30 native types that cover most common needs (Text line, Rich text, Email, Author list, Content relation, Map location, Float, etc.)

Field Types are responsible for:

- storing data, either using the native storage engine mechanisms, or specific means
- validating input data
- making the data searchable (if applicable)
- displaying an instance of the type

Custom Field Types are a very powerful type of extension, since they allow you to hook deep into the content model.

You can find the in-depth [documentation about Field Types and their best practices](#). It describes how each component of a Field Type interacts with the various layers of the system, and how to implement those.

About this tutorial

This tutorial aims at covering the conception and development of a custom eZ Platform Field Type.

We will do this by implementing a Tweet Field Type. It will:

- accept as input the URL of a tweet (<https://twitter.com/<username>/status/<id>>)
- fetch the tweet using the twitter oEmbed API (<https://dev.twitter.com/docs/embedded-tweets>)
- store the tweet's embed contents and URL
- display the tweet's embedded version when displaying the field from a template



Getting the code

The code created throughout this tutorial is available on github: <https://github.com/eZsystems/TweetFieldTypeBundle>.

Steps

The bundle

Field Types, like any other eZ Platform plugin, must be provided as Symfony 2 bundles. This chapter will cover the creation and organization of this bundle

Read more about [Creating the bundle](#) and [structuring the bundle](#).

API

This part will cover the implementation of the eZ Platform API elements required to implement a custom Field Type

Read more about [Implementing the Tweet\Value class](#) and [Implementing the Tweet\Type class](#).

Converter

Storing data from any Field Type into the Legacy Storage Engine requires that your custom data is mapped to the data model.

Read more about [Implementing the Legacy Storage Engine Converter](#).

Templating

Displaying a Field Type's data is done through a Twig template.

Read more about [implementing the Field Type template](#).

PlatformUI integration

Viewing and editing values of the FieldType in PlatformUI requires that you extend PlatformUI, using mostly Javascript.

You should ideally read the general [extensibility documentation for PlatformUI](#). The part about [templating](#) covers view templates. Edit templates are not documented at the time of writing, but [Netgen](#) has published a tutorial that covers the topic: <http://www.netgenlabs.com/Blog/Adding-support-for-a-new-field-type-to-eZ-Publish-Platform-UI>.