Registering the FieldType as a service

To complete the implementation, we must register our FieldType with Symfony, by creating a service for it.

Services are by default declared by bundles in `Resources/config/services.yml`.

Using a dedicated file for the FieldType services
In order to be closer with the kernel best practices, we could declare our FieldType services in a custom `fieldtypes.yml` file.

All we have to do is instruct the bundle to actually load this file in addition to `services.yml` (or instead of `services.yml` !). This is done in the extension definition file, `DependencyInjection/EzSystemsTweetFieldTypeExtension.php`, in the `load()` method.

Inside this file, file, find this line:

```
$loader->load('services.yml');
```

This is where our bundle tells Symfony that when parameters are loaded, `services.yml` should be loaded from `Resources/config/` (defined above). Either replace edit the line, or add a new one with:

```
$loader->load('fieldtypes.yml');
```

Like most API components, FieldTypes use the Symfony 2 service tag mechanism.

The principle is quite simple: a service can be assigned one or several tags, with specific parameters. When the dependency injection container is compiled into a PHP file, tags are read by `CompilerPass` implementations that add extra handling for tagged services. Each service tagged as `ezpublish.fieldType` is added to a registry using the alias argument as its unique identifier (`ezstring`, `ezxmltext`...). Each FieldType must also inherit from the abstract ezpublish.fieldType service. This ensures that the initialization steps shared by all fieldtypes are executed.

Here is the service definition for our Tweet type:

Resources/config/services.yml

```
parameters:
    ezsystems.tweetfieldtypebundle.fieldType.eztweet.class:
EzSystems\TweetFieldTypeBundle\eZ\Publish\FieldType\Tweet\Type

services:
    ezsystems.tweetfieldtypebundle.fieldType.eztweet:
        parent: ezpublish.fieldType
        class: %ezsystems.tweetfieldtypebundle.fieldType.eztweet.class%
        tags:
            - {name: ezpublish.fieldType, alias: eztweet}
```

We take care of namespacing our fieldtype with our vendor and bundle name to limit the risk of naming conflicts.