Create the bundle

Custom Field Types follow the Symfony 2 extension mechanism: bundles. We can get started with a bundle using the built-in Symfony 2 bundle generator.

Generating the bundle

From the eZ Platform root, run the following:

```
php app/console generate:bundle
```

First, we are asked for the namespace. As the vendor, we will use EzSystems as the root namespace. This must of course be changed to whatever identifies you as a vendor (your name, company name, etc). We then choose a preferably unique name for the Field Type itself, and make the name end with Bundle. Here we've chosen TweetFieldTypeBundle.

```
Bundle namespace: EzSystems/TweetFieldTypeBundle<enter>
```

We must next input the bundle's name. Nothing exotic here: we concatenate the vendor's namespace and the bundle's namespace, which is the default. Just hit Enter:

```
Bundle name [EzSystemsTweetFieldTypeBundle]:<enter>
```

We are then asked for the target directory. We will begin within the `src` folder, but we could (and should!) version it and have it moved to `vendor` at some point. Again, this is the default, and we just hit Enter.

```
Target directory
[/home/bertrand/www/ezpublish5/src]:<enter>
```

We must then specify which format the configuration must be generated as. We will use yml, since it is what we use in eZ Platform itself. Of course, any other format could be used.

```
Configuration format (yml, xml, php, or annotation):
yml<enter>
```

The generator will then offer to create the whole directory structure for us. Since our bundle isn't really a standard Symfony full stack bundle, we decline.

```
Do you want to generate the whole directory structure
[no]? no<enter>
```

We then get a summary of what will be generated:

```
You are going to generate a
"EzSystems\TweetFieldTypeBundle\EzSystemsTweetFieldTypeB
undle" bundle
in "/home/bertrand/www/ezpublish5/src/" using the "yml"
format.
Do you confirm generation [yes]?<enter>
```

After generation, the wizard will offer to update the kernel with our bundle (answer "yes"), and to update the app's routing with our bundle's route file (answer "no").

```
Generating the bundle code: OK
Checking that the bundle is autoloaded: OK
Confirm automatic update of your Kernel [yes]? <enter>
Enabling the bundle inside the Kernel: OK
Confirm automatic update of the Routing [yes]? no
<enter>
```

Our bundle should now be generated. Navigate to `src/EzSystems/EzSystemsTweetBundle` and you should see the following structure:

```
$ ls -l src/EzSystems/TweetFieldTypeBundle
Controller
DependencyInjection
EzSystemsTweetFieldTypeBundle.php
Resources
Tests
```

Feel free to delete the Controller folder, since we won't use it in this tutorial. It could have been useful, had our Field Type required an interface of its own.