

Using form inheritance in Symfony 2

 devvness.blogspot.com/2013/11/using-form-inheritance-in-symfony-2.html

During the development of a web app it's fairly common to need forms sharing most of their fields. For example we often need a "new" and an "edit" form for some of the entities in our app. Normally we only need to add or remove some fields to get the new form and we don't want to replicate code as it will make our app harder to maintain and more error-prone.

Let's say we have an Event entity and we need two forms: an EventNew form to create a new Event and another form that we will call EventEdit in order to edit an Event entity. We want our New form to have the following fields:

- Name
- Category
- Subcategory
- Datetime

On the other hand, we want the EventEdit form to have the same fields except for Datetime (for some reason we don't want the user to be able to change the Datetime of the event once it's created). In addition, we have the Edit form to have a textarea field named "description" (we didn't want to annoy the user requiring a description for the event when it's created). So the list of fields for the edit form would be as follows:

- Name
- Category
- Subcategory
- Description

We have at least two ways to solve this problem. We can create a BaseEventType class including the common fields for both forms and then inherit from it adding/removing fields as needed or we can create just two forms (the "new" one and the "edit" one) and make the EventEditType inherit from EventNewType. If you're sure you won't need more Event forms in the future you can just stick to option two. If we take option one we should create a class named BaseEventType containing all the common fields:

Now we need an EventNewType for our new event form and we'll add the datetime field. Note that we need to call to the method buildForm of parent class (BaseEventType, which we extend).

Last but not least we create our EventEditType and we already have our edit event form:

If we decided to take option 2 we should create a `EventNewType` including all 4 fields (name, category, subcategory, datetime). Then we would create `EventEditType` inheriting from `EventNewType` and add/remove fields as necessary: