

Symfony is *fantastic* for building very quick snappy applications; especially CRUD apps. However, sometimes we want to quickly alter the default CRUD forms with some business logic. For example, if we have **Products** and **Categories** and want to be able to create new *product* entities with a specified *category* but somehow restrict which categories are accessible in the product form.

## Altering the FormType Class

```
<?php

namespace MyBundle\Form;

use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolverInterface;
use Symfony\Component\Form\FormEvent;
use Symfony\Component\Form\FormEvents;
use Doctrine\ORM\EntityManager;

class ProductType extends AbstractType
{
    private $em;

    public function __construct(EntityManager $em)
    {
        $this->em = $em;
    }

    /**
     * @param FormBuilderInterface $builder
     * @param array $options
     */
```

```

    public function buildForm(FormBuilderInterface $builder, array
$options)
    {
        $builder
            ->add('name')
            ->add('price')
        ;

        $builder->addEventListener(FormEvents::PRE_SET_DATA, function
(FormEvent $event) {

            $form = $event->getForm();

            $entities = array();
            $categories =
$this->em->getRepository('MyBundle:Category')->findAll();

            /**
             * Here you do your logic to filter out categories you
don't
             * want to be selectable in this Form.
             */
            foreach($categories as $delta => $category) {

                if($category->getType() == 'FREE') {
                    unset($categories[$delta]);
                }

            }

        }
    }

```

```

        $form->add('category', 'entity', array(
            'class'          => 'MyBundle\Entity\Category',
            'placeholder' => '',
            'choices'       => $categories,
        ));

    });

}
/**
 * @param OptionsResolverInterface $resolver
 */
public function setDefaultOptions(OptionsResolverInterface
$resolver)
{
    $resolver->setDefaults(array(
        'data_class' => 'MyBundle\Entity\Product'
    ));
}

/**
 * @return string
 */
public function getName()
{
    return 'mybundle_question';
}
}

```

## Define the Form as a Service

```
my_bundle.form.question_type:  
    class: MyBundle\Form\QuestionType  
    arguments:  
        em: "@doctrine.orm.entity_manager"
```

## Usage in a Controller

```
<?php
```

```
namespace MyBundle\Controller;  
  
use Symfony\Component\HttpFoundation\Request;  
use Symfony\Bundle\FrameworkBundle\Controller\Controller;  
  
use MyBundle\Entity\Product;  
use MyBundle\Form\ProductType;  
  
/**  
 * Product controller.  
 */  
class ProductController extends Controller  
{  
  
    /**  
     * Creates a form to create a Product entity.  
     *  
     * @param Product $entity The entity  
     *  
     * @return \Symfony\Component\Form\Form The form
```

```
*/
private function createCreateForm(Product $entity)
{
    $form = $this->createForm(new
ProductType($this->getDoctrine()->getManager()), $entity, array(
        'action' => $this->generateUrl('products_create'),
        'method' => 'POST',
    ));

    $form->add('submit', 'submit', array('label' => 'Create'));

    return $form;
}
}
```

Share this:

- [LinkedIn](#)
- [Twitter](#)
- [Email](#)
- [Print](#)