## Adding the create/edit/delete forms (CRUD)

👉 This section highlights the entity CRUD (create, read, edit, delete) process. At the end you will have learnt the ins and outs of how to show a user an **entity form** for **creation, editing and deleting** of a **custom entity**.

For our CRUD operations, the system needs to know where to put them. Let’s update our **modules/custom/offer/src/Entity/offer.php** once again by adding the forms and links:

```
*   handlers = {
*     "access" = "Drupal\offer\OfferAccessControlHandler",
*     "form" = {
*       "add" = "Drupal\offer\Form\OfferForm",
*       "edit" = "Drupal\offer\Form\OfferForm",
*       "delete" = "Drupal\offer\Form\OfferDeleteForm",
*     },
*   },
*   links = {
*     "canonical" = "/offers/{offer}",
*     "delete-form" = "/offer/{offer}/delete",
*     "edit-form" = "/offer/{offer}/edit",
*     "create" = "/offer/create",
*   },
```

You may understand what we’re adding here. The links are the routes we want to use for CRUD operations on our offer entity. Inside the handlers directory we placed the add/edit and delete form statements.

These routes to not exist yet, so we define them in **modules/custom/offer/offer.routing.yml**

```
offer.add:
 path: '/offers/create'
 defaults:
   _entity_form: offer.add
   _title: 'Add offer'
 requirements:
   _entity_create_access: 'offer'

entity.offer.edit_form:
 path: '/offers/{offer}/edit'
 defaults:
   _entity_form: offer.edit
   _title: 'Edit offer'
 requirements:
   _entity_access: 'offer.edit'

entity.offer.delete_form:
 path: '/offers/{offer}/delete'
 defaults:
   _entity_form: offer.delete
   _title: 'Delete offer'
 requirements:
   _entity_access: 'offer.delete'

entity.offer.canonical:
 path: '/offer/{offer}'
 defaults:
   _entity_view: 'offer'
   _title: 'Offer'
 requirements:
   _entity_access: 'offer.view'
```

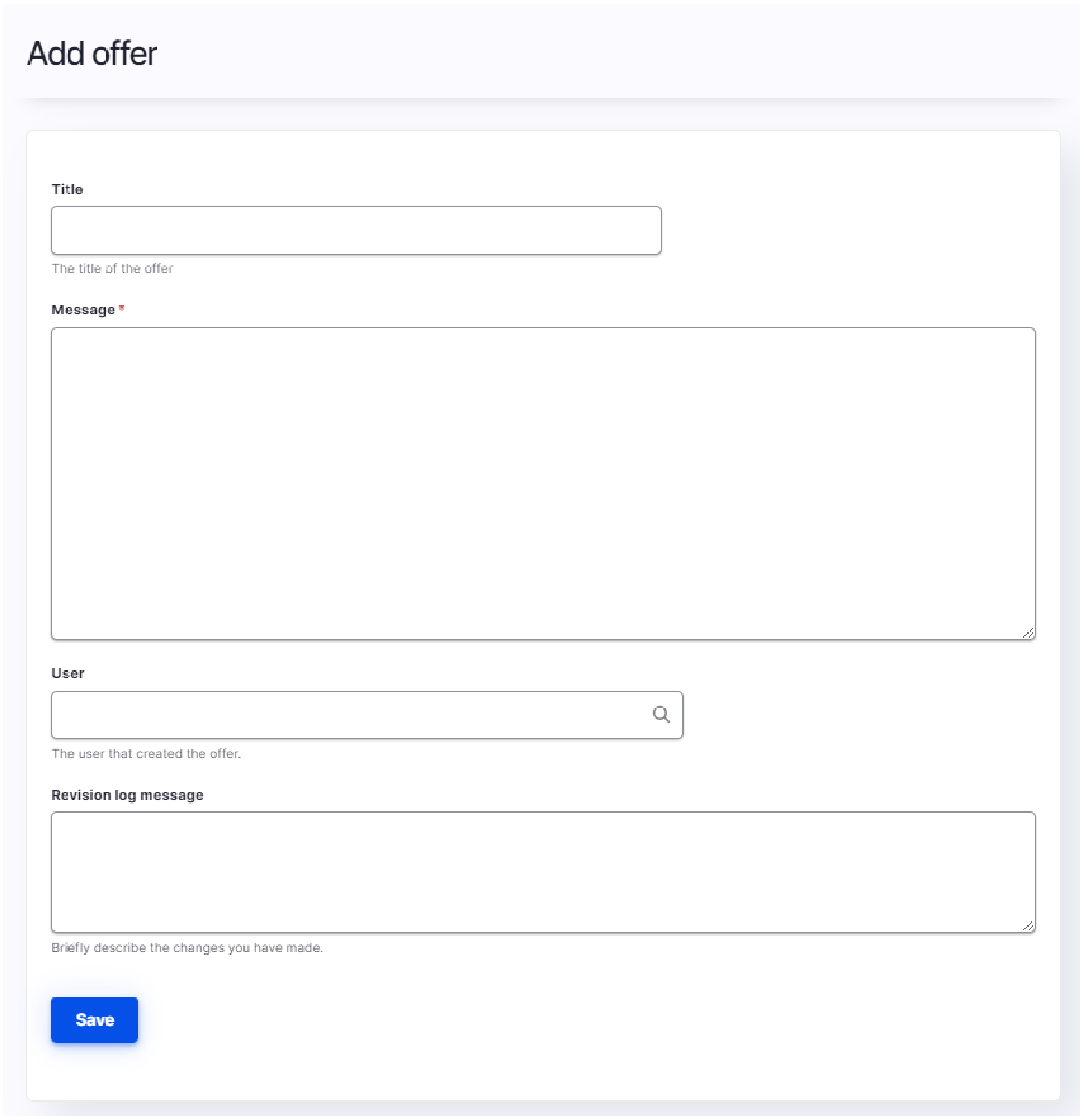
Check the _*entity_access* parameters. These will check the access conditions inside **OfferAccessControlHandler** (checkAccess and checkCreateAccess). By this, we control exactly who can access the /offers/create form for example.

Now we'll add a generic form for adding and editing. It inherits from *ContentEntityForm:* **custom/offer/src/Form/OfferForm.php**:

```
<?php
/**
* @file
* Contains Drupal\offer\Form\OfferForm.
*/

namespace Drupal\offer\Form;

use Drupal\Core\Entity\ContentEntityForm;
use Drupal\Core\Form\FormStateInterface;

/**
* Form controller for the offer entity edit forms.
*
```

```
* @ingroup content_entity_example
*/
class OfferForm extends ContentEntityForm {

 /**
  * {@inheritdoc}
  */
 public function buildForm(array $form, FormStateInterface $form_state)
{
   /* @var $entity \Drupal\offer\Entity\Offer */
   $form = parent::buildForm($form, $form_state);
   return $form;
 }

 /**
  * {@inheritdoc}
  */
 public function save(array $form, FormStateInterface $form_state) {
   // Redirect to offer list after save.
   $form_state->setRedirect('entity.offer.collection');
   $entity = $this->getEntity();
   $entity->save();
 }

}
```

Clear cache and head to /offers/create. There it is! We've gotten ourselves access to the create and edit mode of this entity.


Add offer

Title

The title of the offer

Message *

User

The user that created the offer.

Revision log message

Briefly describe the changes you have made.

💻 If you get an “Access denied”, make sure authenticated users have permission to ‘administer own offers’ via *admin/people/permissions*.

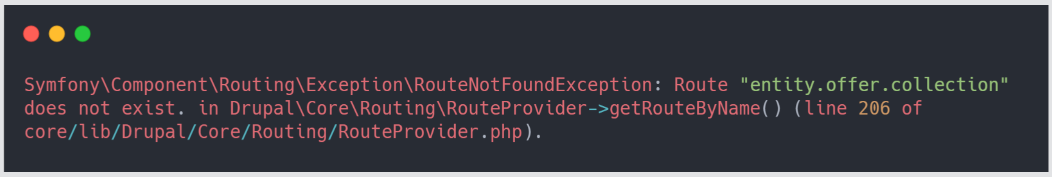
Saving an entity will not work at the moment. We get this error:

```
The website encountered an unexpected error. Please try again later.
```

This is not a readable error when developing. Add this to your **settings.php**:

```
$config['system.logging']['error_level'] = 'verbose';
```

Refresh the page to see the error:



```
Symfony\Component\Routing\Exception\RouteNotFoundException: Route "entity.offer.collection"
does not exist. in Drupal\Core\Routing\RouteProvider->getRouteByName() (line 206 of
core/lib/Drupal/Core/Routing/RouteProvider.php).
```

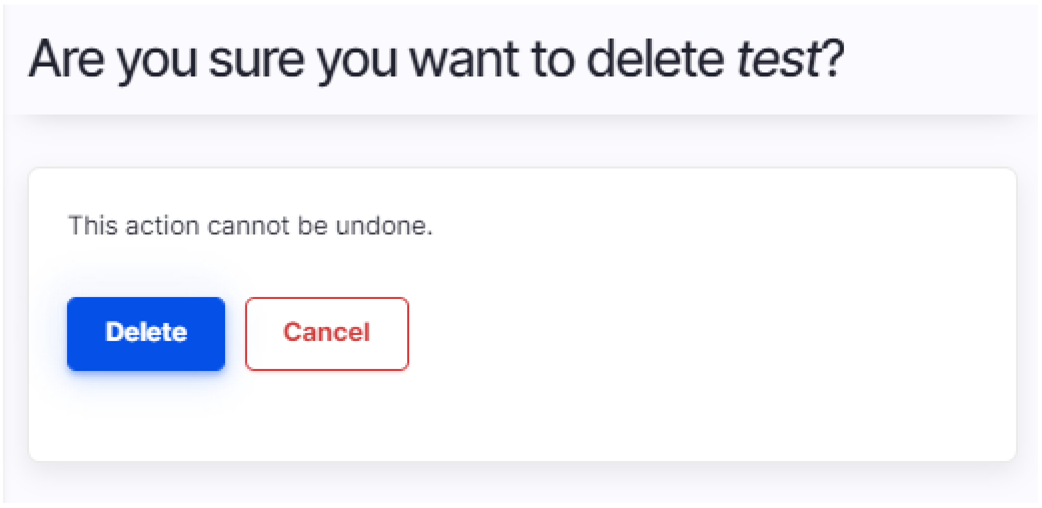
This tells us the page we redirect to after saving does not exist yet, We will fix this in the views integration chapter.

At /offers/1/edit you should see your created entity in an edit form with your submitted values filled in.

Apart from the overview page, a missing step towards full CRUD operations is the delete form. We've already added the definition in our entity and routing, so now add **custom/offer/src/Form/OfferDeleteForm.php**:

```
<?php

/**
* @file
* Contains \Drupal\offer\Form\OfferDeleteForm.
*/

namespace Drupal\offer\Form;

use Drupal\Core\Entity\ContentEntityConfirmFormBase;
use Drupal\Core\Form\FormStateInterface;
use Drupal\Core\Url;

/**
* Provides a form for deleting a content_entity_example entity.
*
* @ingroup offer
*/
class OfferDeleteForm extends ContentEntityConfirmFormBase {

 /**
   * {@inheritdoc}
   */
 public function getQuestion() {
   return $this->t('Are you sure you want to delete %name?',
array('%name' => $this->entity->label()));
 }

  /**
   * {@inheritdoc}
   *
   * If the delete command is canceled, return to the offer.
   */
  public function getCancelUrl() {
    return Url::fromRoute('entity.offer.edit_form', ['offer' =>
$this->entity->id()]);
  }


 /**
  * {@inheritdoc}
  */
 public function getConfirmText() {
   return $this->t('Delete');
 }

 /**
  * {@inheritdoc}
  *
  * Delete the entity
  */
 public function submitForm(array &$form, FormStateInterface
$form_state) {
   $entity = $this->getEntity();
   $entity->delete();

   $this->logger('offer')->notice('deleted %title.',
     array(
       '%title' => $this->entity->label(),
     ));
   // Redirect to offer list after delete.
   $form_state->setRedirect('entity.offer.collection');
 }

}
```

We are now able to delete an offer with a confirmation form before deleting. If you tried creating an entity via the create form, you will now be able to visit offers/1/delete.


Are you sure you want to delete *test*?

This action cannot be undone.

We get an error after deletion because again, we get redirected towards the collection of this entity:

```
$form_state->setRedirect('entity.offer.collection');
```

The error happens because the entity has not defined a collection ("listing") class yet. Typically, drupal offers an entity listing functionality for entities that get defined in a list builder. Personally I think there is a better option because the ux and flexibility are not that great. Also, there is a much more powerful option for this. We will use the core Views module to offer a listing of our entities in the next chapter to fix the error and finalize our CRUD operations.