NAME
OVERVIEW
DESCRIPTION
Install HTML::FormHandler
HTML::FormHandler FORM CREATION
  Using FormHandler in your controllers
  Create a Book Form
  Add Action to Display and Save the Form
  Create a Template Page To Display The Form
  Add Link for Create
  Test The HTML::FormHandler Create Form
  Add Constraints
  Try Out the Updated Form
  Create the 'edit' method
  Try Out the Edit/Update Feature
  See additional documentation on FormHandler
AUTHOR

NAME

Catalyst::Manual::Tutorial::09_AdvancedCRUD::09_FormHandler - Catalyst Tutorial - Chapter 9: Advanced CRUD - FormHandler

OVERVIEW

This is **Chapter 9 of 10** for the Catalyst tutorial.

Tutorial Overview

1. Introduction
2. Catalyst Basics
3. More Catalyst Basics
4. Basic CRUD
5. Authentication
6. Authorization
7. Debugging
8. Testing
9. **09_Advanced CRUD::09_FormHandler**
10. Appendices

DESCRIPTION

This portion of the tutorial explores HTML::FormHandler and how it can be used to manage forms, perform validation of form input, and save and restore data to or from the database. This was written using HTML::FormHandler version 0.28001.

See Catalyst::Manual::Tutorial::09_AdvancedCRUD for additional form management options other than HTML::FormHandler.

Install HTML::FormHandler

Use the following command to install HTML::FormHandler::Model::DBIC directly from CPAN:

```
    sudo cpan HTML::FormHandler::Model::DBIC
```

It will install HTML::FormHandler as a prerequisite.

Also, add:

```
    requires 'HTML::FormHandler::Model::DBIC';
```

to your `Makefile.PL`.

HTML::FormHandler FORM CREATION

This section looks at how HTML::FormHandler can be used to add additional functionality to the manually created form from Chapter 4.

Using FormHandler in your controllers

FormHandler doesn't have a Catalyst base controller, because interfacing to a form is only a couple of lines of code.

Create a Book Form

Create the directory `lib/MyApp/Form`. Create `lib/MyApp/Form/Book.pm`:

```
    package MyApp::Form::Book;

    use HTML::FormHandler::Moose;
    extends 'HTML::FormHandler::Model::DBIC';
    use namespace::autoclean;

    has '+item_class' => ( default =>'Books' );
    has_field 'title';
    has_field 'rating' => ( type => 'Integer' );
    has_field 'authors' => ( type => 'Multiple', label_column => 'last_name' );
    has_field 'submit' => ( type => 'Submit', value => 'Submit' );

    __PACKAGE__->meta->make_immutable;
    1;
```

Add Action to Display and Save the Form

At the top of the `lib/MyApp/Controller/Books.pm` add:

```
    use MyApp::Form::Book;
```

Add the following methods:

```
=head2 create

Use HTML::FormHandler to create a new book

=cut

sub create : Chained('base') PathPart('create') Args(0) {
    my ($self, $c ) = @_;

    my $book = $c->model('DB::Book')->new_result({});
    return $self->form($c, $book);
}

=head2 form

Process the FormHandler book form

=cut

sub form {
    my ( $self, $c, $book ) = @_;

    my $form = MyApp::Form::Book->new;
    # Set the template
    $c->stash( template => 'books/form.tt2', form => $form );
    $form->process( item => $book, params => $c->req->params );
    return unless $form->validated;
    # Set a status message for the user & return to books list
    $c->response->redirect($c->uri_for($self->action_for('list'),
        {mid => $c->set_status_msg("Book created")}));
}
```

These two methods could be combined at this point, but we'll use the 'form' method later when we implement 'edit'.

Create a Template Page To Display The Form

Open `root/src/books/form.tt2` in your editor and enter the following:

```
[% META title = 'Create/Update Book' %]

[%# Render the HTML::FormHandler Form %]
[% form.render %]

<p><a href="[% c.uri_for(c.controller.action_for('list')) %]">Return to book list</a></p>
```

Add Link for Create

Open `root/src/books/list.tt2` in your editor and add the following to the bottom of the existing file:

```
...
<p>
  HTML::FormHandler:
  <a href="[% c.uri_for(c.controller.action_for('create')) %]">Create</a>
</p>
```

This adds a new link to the bottom of the book list page that we can use to easily launch our HTML::FormHandler-based form.

Test The HTML::FormHandler Create Form

Press `Ctrl-C` to kill the previous server instance (if it's still running) and restart it:

```
    $ script/myapp_server.pl
```

Login as `test01` (password: mypass). Once at the Book List page, click the new HTML::Formhandler "Create" link at the bottom to display the form. Fill in the following values:

```
    Title  = "Internetworking with TCP/IP Vol. II"
    Rating = "4"
    Author = "Comer"
```

Click the Submit button, and you will be returned to the Book List page with a "Book created" status message displayed.

Note that because the 'Author' column is a Select list, only the authors in the database can be entered. The 'ratings' field will only accept integers.

Add Constraints

Open `lib/MyApp/Form/Book.pm` in your editor.

Restrict the title size and make it required:

```
   has_field 'title' => ( minlength => 5, maxlength => 40, required => 1 );
```

Add range constraints to the 'rating' field:

```
   has_field 'rating' => ( type => 'Integer', range_start => 1, range_end => 5 );
```

The 'authors' relationship is a 'many-to-many' pseudo-relation, so this field can be set to Multiple to allow the selection of multiple authors; also, make it required:

```
   has_field 'authors' => ( type => 'Multiple', label_column => 'last_name',
                            required => 1 );
```

Note: FormHandler automatically strips whitespace at the beginning and end of fields. If you want some other kind of stripping (or none) you can specify it explicitly; see HTML::FormHandler::Manual.

Try Out the Updated Form

Press `Ctrl-C` to kill the previous server instance (if it's still running) and restart it:

```
    $ script/myapp_server.pl
```

Make sure you are still logged in as `test01` and try adding a book with various errors: title less than 5 characters, non-numeric rating, a rating of 0 or 6, etc. Also try selecting one, two, and zero authors.

Create the 'edit' method

Edit `lib/MyApp/Controller/Books.pm` and add the following method:

```
=head2 edit

Edit an existing book with  FormHandler

=cut

sub edit : Chained('object') PathPart('edit') Args(0) {
    my ( $self, $c ) = @_;

    return $self->form($c, $c->stash->{object});
}
```

Update the `root/src/books/list.tt2`, adding an 'edit' link below the "Delete" link to use the FormHandler edit method:

```
<td>
  [% # Add a link to delete a book %]
  <a href="[% c.uri_for(c.controller.action_for('delete'), [book.id]) %]">Delete</a>
  [% # Add a link to edit a book %]
  <a href="[% c.uri_for(c.controller.action_for('edit'), [book.id]) %]">Edit</a>
</td>
```

Try Out the Edit/Update Feature

Press `Ctrl-C` to kill the previous server instance (if it's still running) and restart it:

```
$ script/myapp_server.pl
```

Make sure you are still logged in as `test01` and go to the http://localhost:3000/books/list URL in your browser. Click the "Edit" link next to "Internetworking with TCP/IP Vol. II", change the rating to a 3, the "II" at end of the title to the number "2", add Stevens as a co-author (control-click), and click Submit. You will then be returned to the book list with a "Book edited" message at the top in green. Experiment with other edits to various books.

See additional documentation on FormHandler

HTML::FormHandler::Manual

HTML::FormHandler

```
#formhandler on irc.perl.org

mailing list: http://groups.google.com/group/formhandler

code: http://github.com/gshank/html-formhandler/tree/master
```

AUTHOR

Gerda Shank, `gshank@cpan.org`

syntax highlighting: no syntax highlighting ▼