

Rails Lab 3

Step 1

The owners of the bookstore are very happy with their new web site, but they would like us to update it to allow users add their second hand books to the site to be sold! The new functionality will allow a user post their own books for sale.

The add book page will have the same data structure as our Show page.

name	string
description	string
price	decimal
Seller Id	integer
email	string

The functionality created to display the second hand books was added to the application in the previous lab. So we already know how to publish data from the database. The books currently only go one way in the current application. The book records are read from the database by the model, which converts them into book objects that are sent to the views by the controller.

Saving data to the database is similar to publishing data from the database, except it works the opposite way around!

We will need a view(page template) to submit the book data rather than display it.

The controller will need to read in the data from the page and convert it into an object.

The model will need to write the object to the database (create new record) for us instead of reading data for display.

Step 2

A form will be needed to submit data and an action will be needed to save it.

Create a new page template and save it in the apps\Views folder (new.html.erb).

The new page will appear at: <http://0.0.0.0:3000/ads/new>

When the form is being submitted it will be sent to <http://0.0.0.0:3000/ads/create>

Step 3

A new route will be needed for this in routes.rb

Add the following routes to the top of the routes.rb file.

```
match '/ads/new', :controller=>'ads', :action=>'new'
```

```
match '/ads/create', :controller=>'ads', :action=>'create'
```

By placing them before the show route eliminates any possible mixup with the routes.

Step 4

Rails will create the form for with the associated model object.

Rails can use a model object to create a form.

```
<h1>New Book</h1>

<% form_for(@ad, :url=>{:action=>'create'}) do |f| %>

<p><b>Name:</b><br><%= f.text_field :name %></p>

<p><b>Description:</b><br><%= f.text_area :description %></p>

<p><b>Price:</b><br><%= f.text_field :price %></p>

<p><b>Seller:</b><br><%= f.text_field :seller_id %></p>

<p><b>Email:</b><br><%= f.text_field :email %></p>

<p><%=f.submit "Create" %></p>

<% end %>
```

Step 5

Now we need to update the ads controller

Add the following to the controller:

```
def new

  @ad = Ad.new

end
```

Now you have the controller creating the new book object before the new.html.erb page template runs. See if this displays. Run the server and open: <http://0.0.0.0:3000/ads/new>

Step 6

The form page should display. But if you try to enter some data, an error will occur!

Rails is telling us to make a create action. This will read in the data from the client and create a new book object.

In the ads controller:

```
def create
  @ad=Ad.new(params[:ad])
  @ad.save
end
```

Step 7

Now we should be able to view our new insert page and add a new book.

But when the create button is clicked, we get an error message:

Missing template ads/create.html.erb

Rails works in pairs, with both requests and responses.

We need to write the create.html.erb

```
<h1>Book Added to the system!</h1>
```

```
View your book <a href="/ads/<%= @ad.id %>">here</a>
```