A Many-to-Many tutorial for Rails

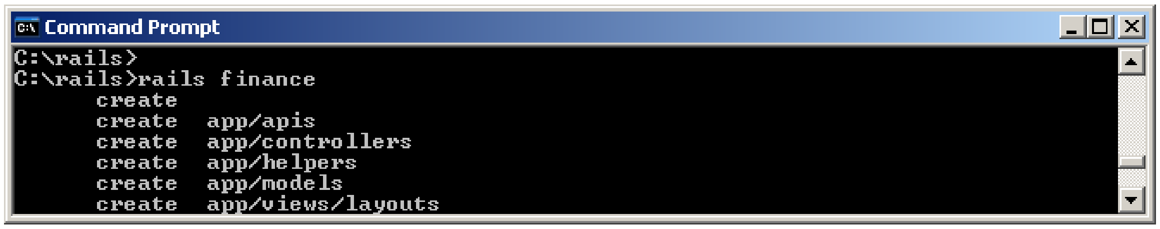
September 4th, 2005

Originally by Jeffrey Hicks @ http://jrhicks.net

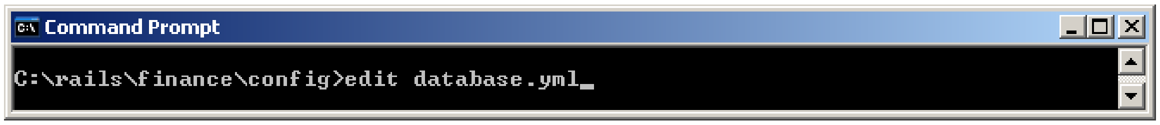
Introduction

This brief tutorial is a start-to-finish example of the Model, View, and Controller required for a many-to-many relationship.

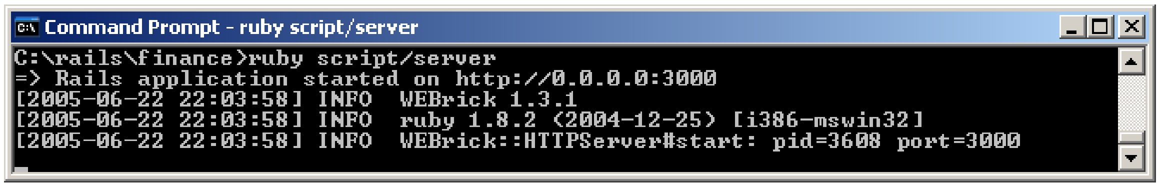
This is a follow-along tutorial for a *finance* application so go ahead and create your rails app, configure your database.yml, and start your server.



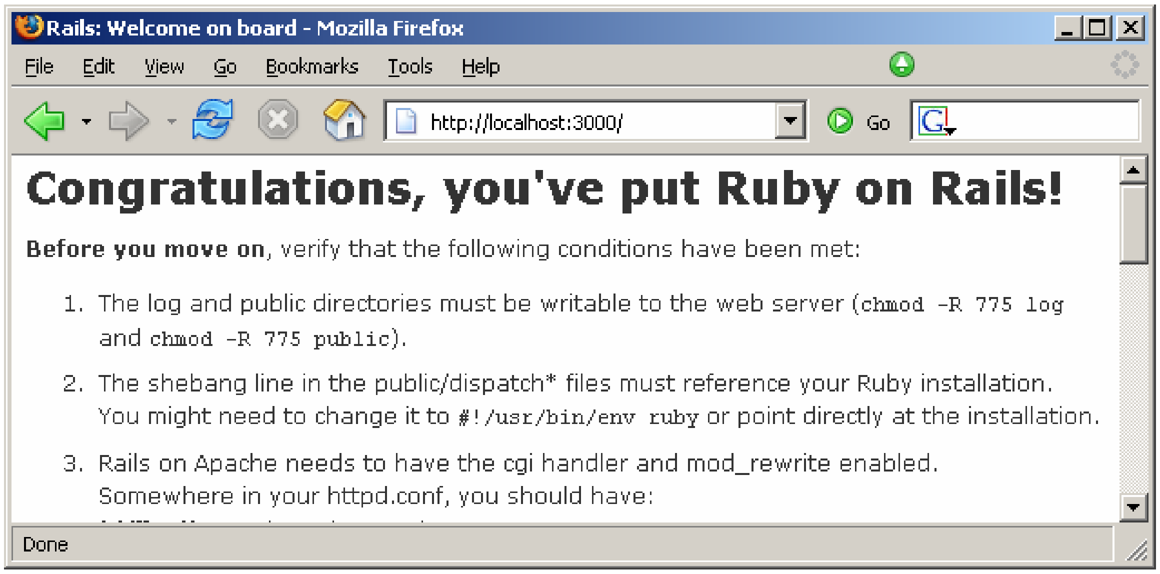
```
C:\rails>
C:\rails>rails finance
      create
      create  app/apis
      create  app/controllers
      create  app/helpers
      create  app/models
      create  app/views/layouts
```



```
C:\rails\finance\config>edit database.yml_
```



```
C:\rails\finance>ruby script/server
=> Rails application started on http://0.0.0.0:3000
[2005-06-22 22:03:58] INFO  WEBrick 1.3.1
[2005-06-22 22:03:58] INFO  ruby 1.8.2 (2004-12-25) [i386-mswin32]
[2005-06-22 22:03:58] INFO  WEBrick::HTTPServer#start: pid=3608 port=3000
```



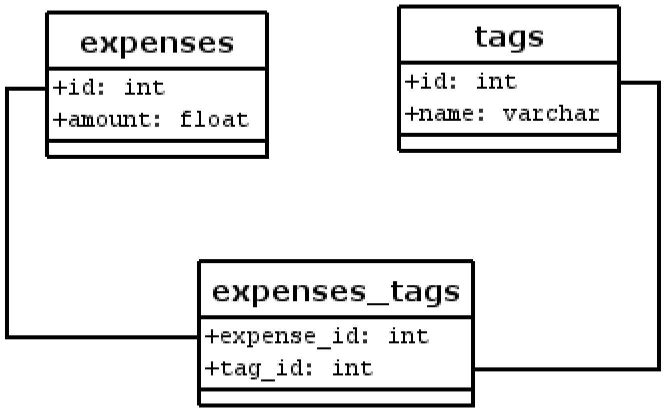
Congratulations, you've put Ruby on Rails!

Before you move on, verify that the following conditions have been met:

1. The log and public directories must be writable to the web server (`chmod -R 775 log` and `chmod -R 775 public`).
2. The shebang line in the public/dispatch* files must reference your Ruby installation. You might need to change it to `#!/usr/bin/env ruby` or point directly at the installation.
3. Rails on Apache needs to have the cgi handler and mod_rewrite enabled. Somewhere in your httpd.conf, you should have:

I. Model

Our example application will model financial *expenses* and *tags*. To work with rail's default expectations we follow a strict naming convention for the database table names and fields.



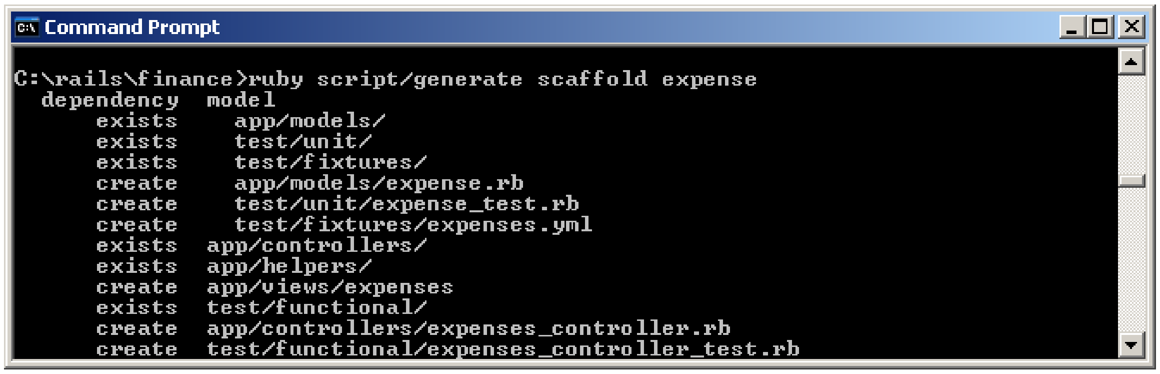
Notice the required naming conventions.

- *expenses* is the plural form of expense
- *tags* is the plural form of tag
- both primary fields use the lowercase *id*
- the relating table is in alphabetical order *expenses_tags*
- the field relating to a tag's id is *tag_id*
- the field relating to an expense's id is *expense_id*

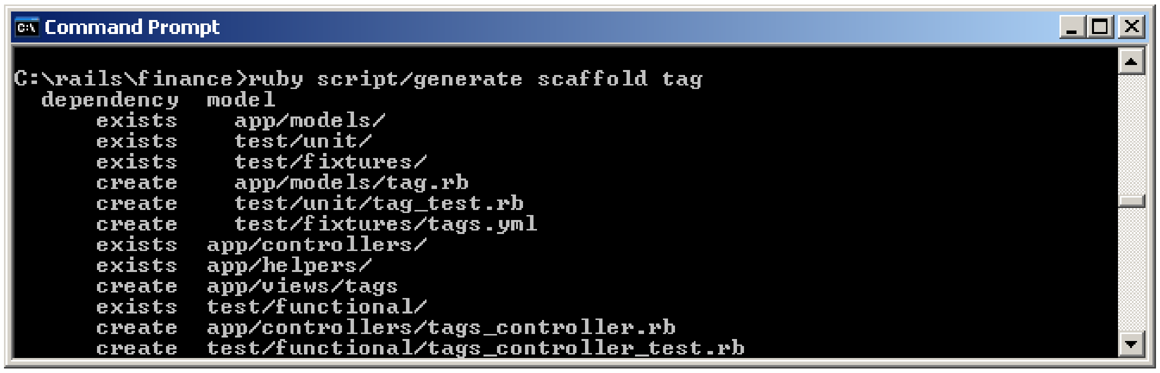
Create your database using the following schema.

```
CREATE TABLE `expenses` (
  `id` int(11) NOT NULL auto_increment,
  `amount` float NOT NULL default '0',
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `tags` (
  `id` int(11) NOT NULL auto_increment,
  `name` varchar(100) NOT NULL default '',
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `expenses_tags` (
  `expense_id` int(11) NOT NULL default '0',
  `tag_id` int(11) NOT NULL default '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Generate your Scaffold for the *expense* and *tag* models.

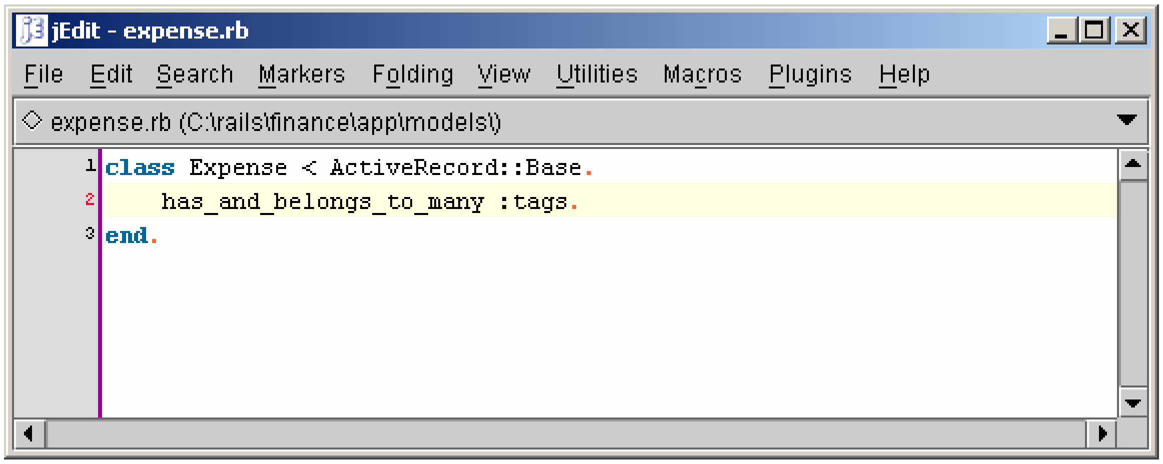


```
C:\rails\finance>ruby script/generate scaffold expense
  dependency  model
      exists    app/models/
      exists    test/unit/
      exists    test/fixtures/
      create    app/models/expense.rb
      create    test/unit/expense_test.rb
      create    test/fixtures/expenses.yml
      exists  app/controllers/
      exists  app/helpers/
      create  app/views/expenses
      exists  test/functional/
      create  app/controllers/expenses_controller.rb
      create  test/functional/expenses_controller_test.rb
```



```
C:\rails\finance>ruby script/generate scaffold tag
  dependency  model
      exists    app/models/
      exists    test/unit/
      exists    test/fixtures/
      create    app/models/tag.rb
      create    test/unit/tag_test.rb
      create    test/fixtures/tags.yml
      exists  app/controllers/
      exists  app/helpers/
      create  app/views/tags
      exists  test/functional/
      create  app/controllers/tags_controller.rb
      create  test/functional/tags_controller_test.rb
```

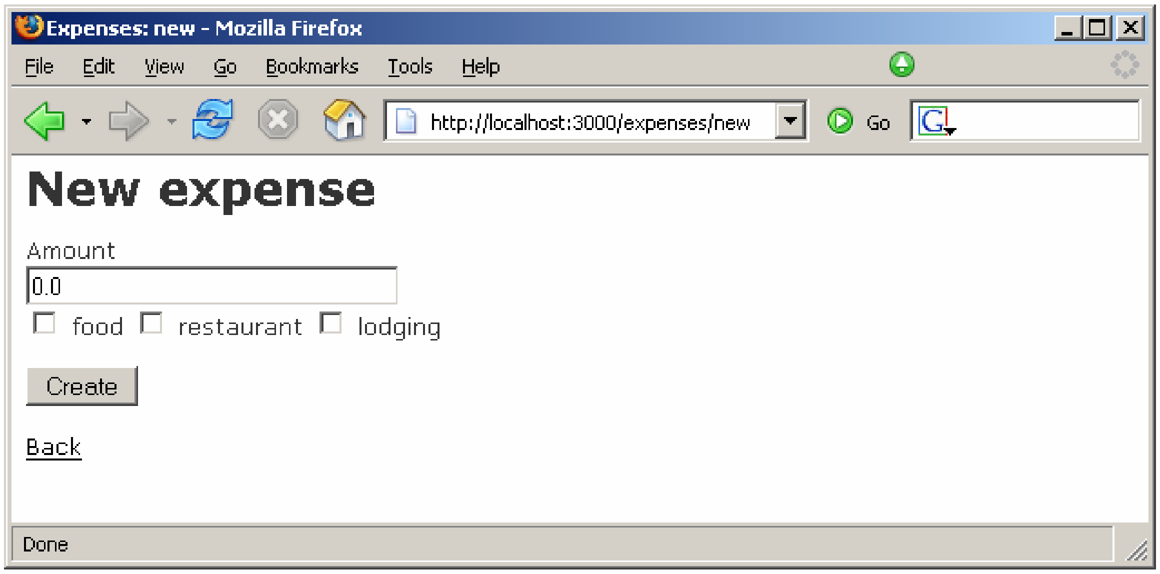
Edit expense.rb to tell your expense model that it **has_and_belongs_to_many** :tags


jEdit - expense.rb
File Edit Search Markers Folding View Utilities Macros Plugins Help
expense.rb (C:\rails\finance\app\models\)

```
class Expense < ActiveRecord::Base
    has_and_belongs_to_many :tags
end
```

2. View

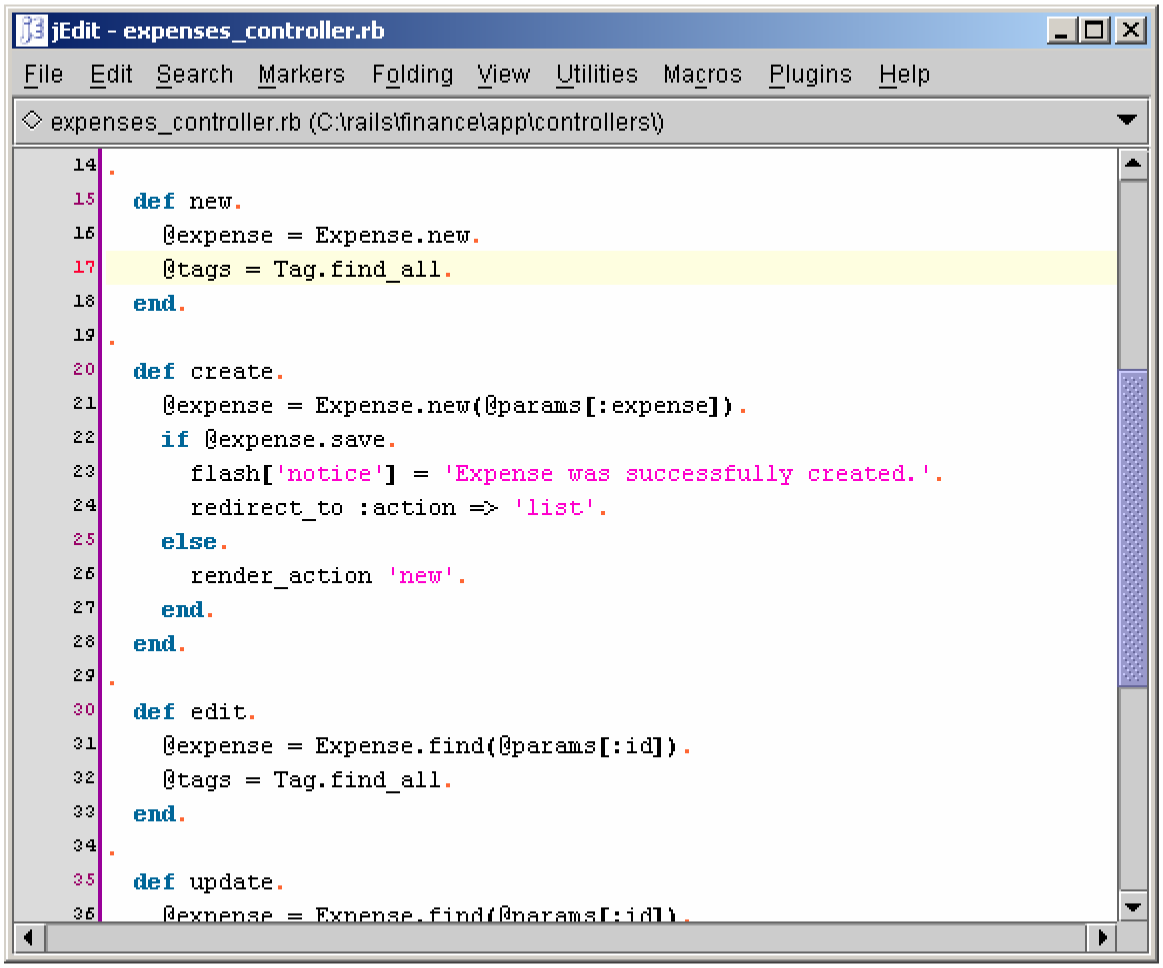
To allow our web visitors to relate an expense with many tags, we are going to use multiple checkboxes. This is how it will look.


Expenses: new - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:3000/expenses/new
Go
New expense
Amount
0.0
food
restaurant
lodging
Create
Back
Done

Our form will generate dynamically from the tags in the database. Execute the following SQL to populate our example database.

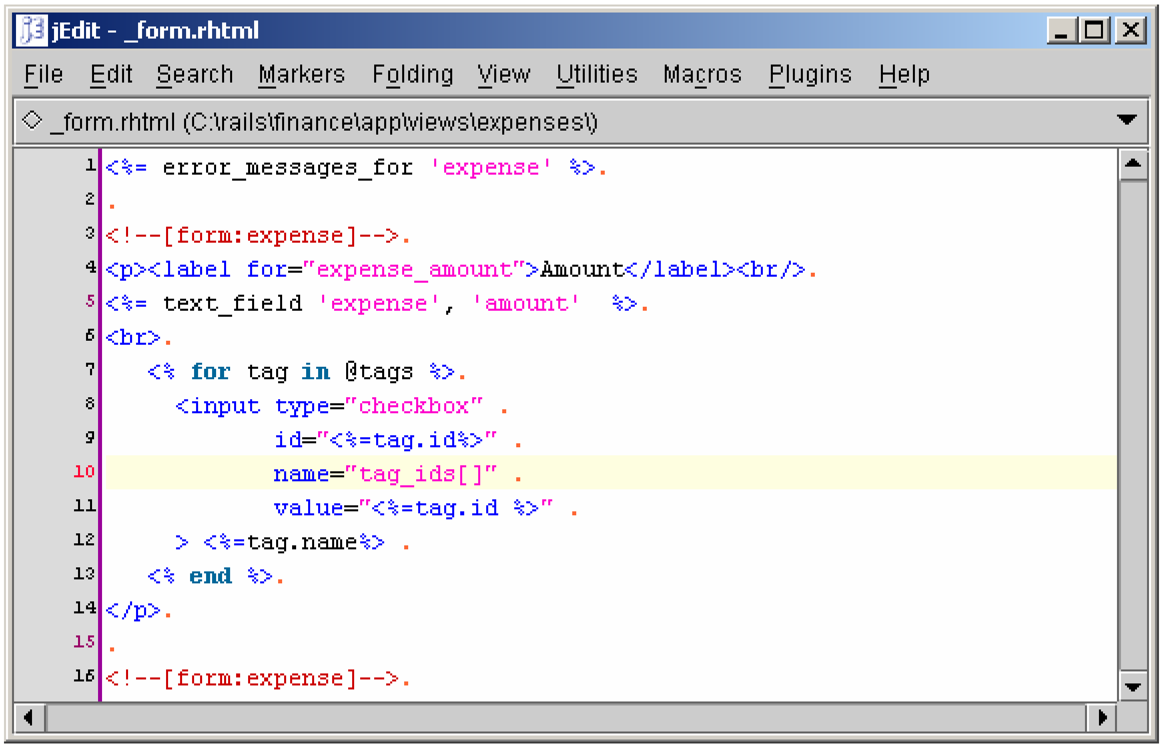
```
Insert into tags(name) values ('food');
Insert into tags(name) values ('restaurant');
Insert into tags(name) values ('lodging');
```

Our view should depend on the *expenses_controller* to load the tags. Add the *@tags=Tag.find_all* line to both the *new* and *edit* methods as depicted in line 17 & 32 below.


jEdit - expenses_controller.rb
File Edit Search Markers Folding View Utilities Macros Plugins Help
expenses_controller.rb (C:\rails\finance\app\controllers\)

```
14
15  def new
16    @expense = Expense.new
17    @tags = Tag.find_all
18  end
19
20  def create
21    @expense = Expense.new(@params[:expense])
22    if @expense.save
23      flash['notice'] = 'Expense was successfully created.'
24      redirect_to :action => 'list'
25    else
26      render_action 'new'
27    end
28  end
29
30  def edit
31    @expense = Expense.find(@params[:id])
32    @tags = Tag.find_all
33  end
34
35  def update
36    @expense = Expense.find(@params[:id])
```

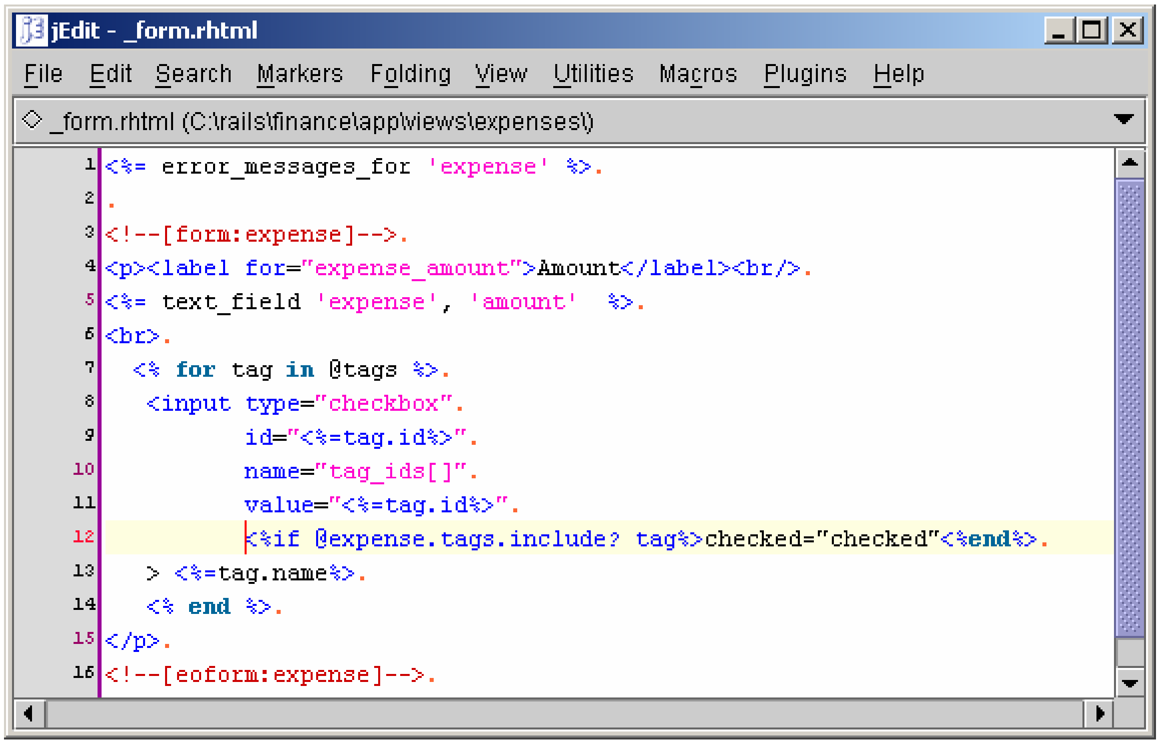
Now we will actually customize our edit and new views. We can do this in one location; edit your *expenses_form.rhtml* to include lines 7 through 13.


jEdit - _form.rhtml
File Edit Search Markers Folding View Utilities Macros Plugins Help
_form.rhtml (C:\rails\finance\app\views\expenses\)

```
 1 <%= error_messages_for 'expense' %>
 2 
 3 <!--[form:expense]-->
 4 <p><label for="expense_amount">Amount</label><br/>
 5 <%= text_field 'expense', 'amount'  %>
 6 <br>
 7    <% for tag in @tags %>
 8      <input type="checkbox"
 9             id="<%=tag.id%>"
10             name="tag_ids[]"
11             value="<%=tag.id %>"
12      > <%=tag.name%>
13    <% end %>
14 </p>
15 
16 <!--[form:expense]-->
```

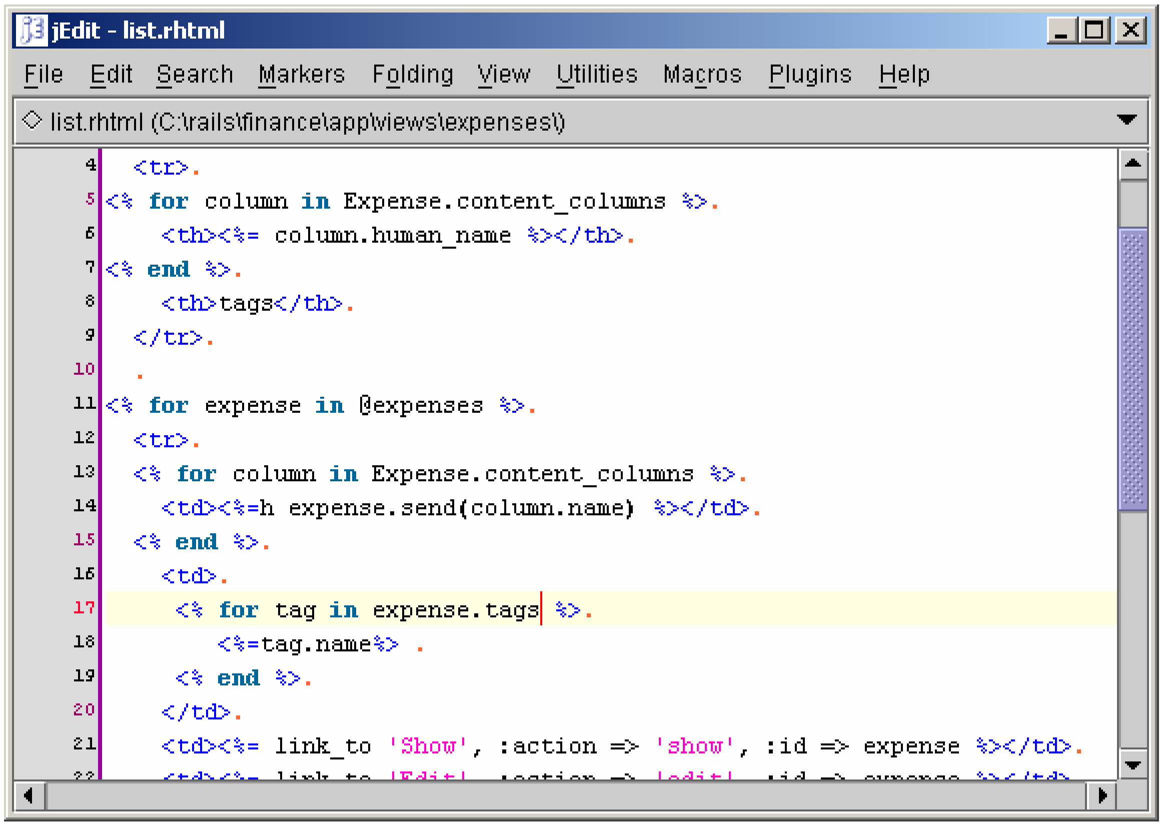
The tags of existing expenses should be checked when we edit. To enable this we add the following if statement to line 12.

<%if @expense.tags.include? tag%>checked="checked"<%end%>


jEdit - _form.rhtml
File Edit Search Markers Folding View Utilities Macros Plugins Help
_form.rhtml (C:\rails\finance\app\views\expenses\)

```
 1 <%= error_messages_for 'expense' %>
 2 
 3 <!--[form:expense]-->
 4 <p><label for="expense_amount">Amount</label><br/>
 5 <%= text_field 'expense', 'amount'  %>
 6 <br>
 7   <% for tag in @tags %>
 8    <input type="checkbox"
 9           id="<%=tag.id%>"
10           name="tag_ids[]"
11           value="<%=tag.id%>"
12           <%if @expense.tags.include? tag%>checked="checked"<%end%>
13    > <%=tag.name%>
14    <% end %>
15 </p>
16 <!--[eoform:expense]-->
```

We will also edit the list view, so that we will be able to view our tags. Add the code on lines 8 and 17-19 to your *list.rhtml*


jEdit - list.rhtml
File Edit Search Markers Folding View Utilities Macros Plugins Help
list.rhtml (C:\rails\finance\app\views\expenses\)

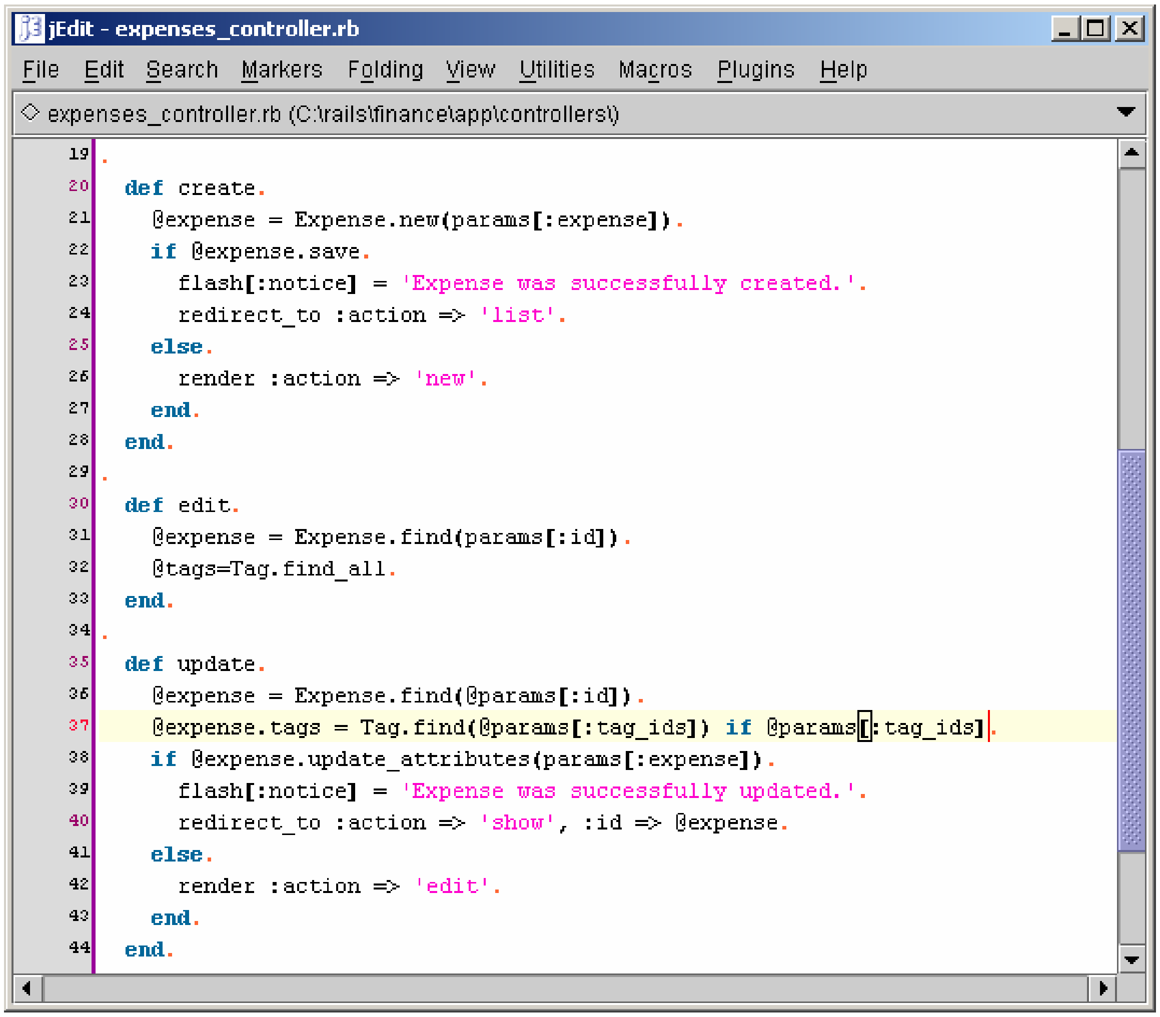
```
 4   <tr>
 5 <% for column in Expense.content_columns %>
 6     <th><%= column.human_name %></th>
 7 <% end %>
 8     <th>tags</th>
 9   </tr>
10
11 <% for expense in @expenses %>
12   <tr>
13   <% for column in Expense.content_columns %>
14     <td><%=h expense.send(column.name) %></td>
15   <% end %>
16     <td>
17      <% for tag in expense.tags %>
18         <%=tag.name%>
19      <% end %>
20     </td>
21     <td><%= link_to 'Show', :action => 'show', :id => expense %></td>
22     <td><%= link_to 'Edit', :action => 'edit', :id => expense %></td>
```

3. Controller

The *expense_controller's update* and *create* method receive the requests from the *edit* and *new* view. To store the relationship we need to convert the *tag_ids* to actual Tag objects with *Tag.find(@params[:tag_ids]) if @params[:tag_ids].*

The *if @params[:tag_ids]* prevents a *nil object* error when the user doesn't select any tag.

Add the lines 22 & 38 to *expenses_controller.rb*


jEdit - expenses_controller.rb
File Edit Search Markers Folding View Utilities Macros Plugins Help
expenses_controller.rb (C:\rails\finance\app\controllers\)

```
19
20    def create
21      @expense = Expense.new(params[:expense])
22      if @expense.save
23        flash[:notice] = 'Expense was successfully created.'
24        redirect_to :action => 'list'
25      else
26        render :action => 'new'
27      end
28    end
29
30    def edit
31      @expense = Expense.find(params[:id])
32      @tags=Tag.find_all
33    end
34
35    def update
36      @expense = Expense.find(@params[:id])
37      @expense.tags = Tag.find(@params[:tag_ids]) if @params[:tag_ids]
38      if @expense.update_attributes(params[:expense])
39        flash[:notice] = 'Expense was successfully updated.'
40        redirect_to :action => 'show', :id => @expense
41      else
42        render :action => 'edit'
43      end
44    end
```

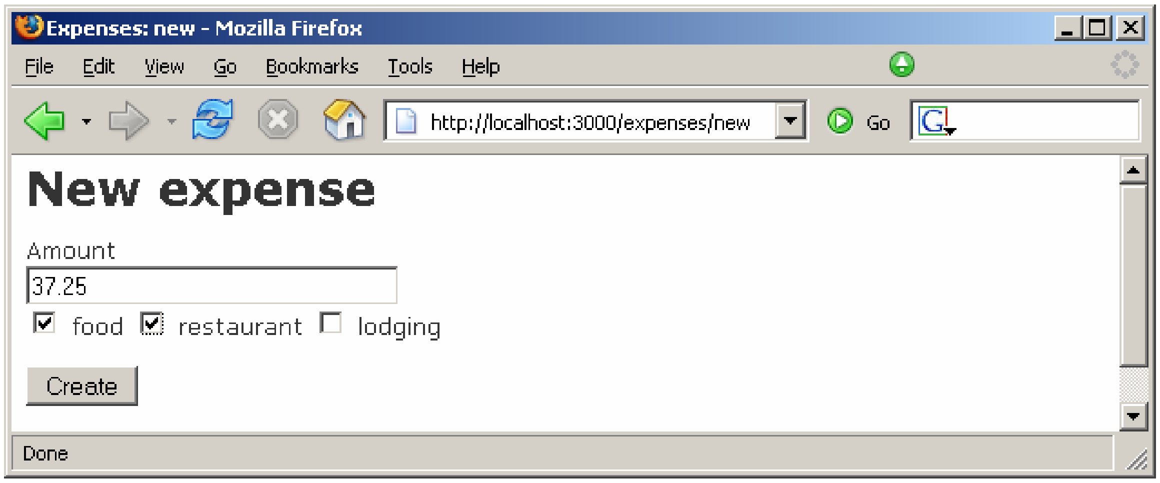
Optionally

If you intention was to force the user to select a tag, I’ve been told to add this to the expense model. (expense_controller.rb)

```
def validate
        if tags.blank?
                errors.add_to_base("You must specify a tag")
        end
end
```

Conclusion

Add an expense with multiple tags.


Expenses: new - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:3000/expenses/new
Go
Create
Done

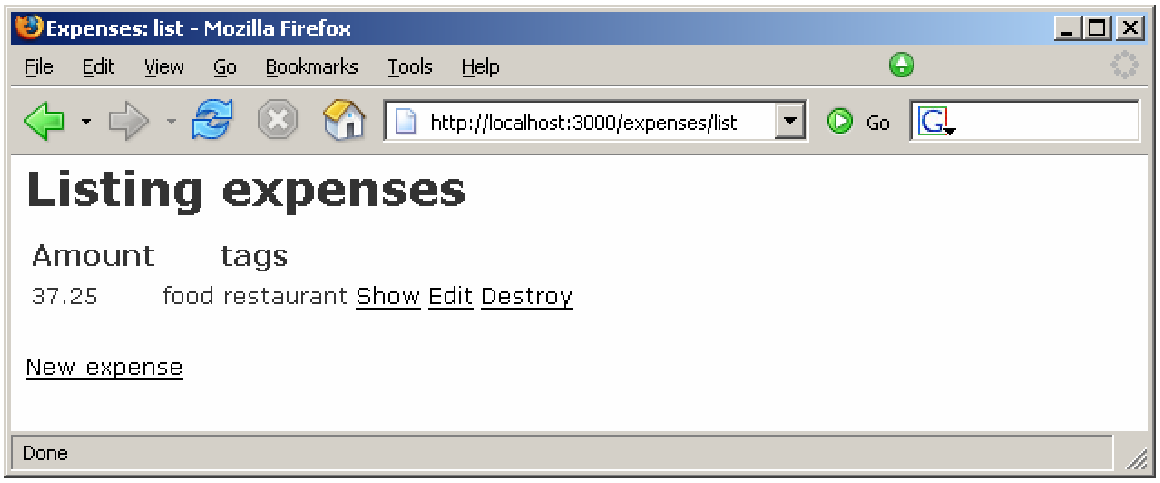
New expense

Amount

37.25

[✓ food] [✓ restaurant] lodging

View the stored tags after you hit the Create button.

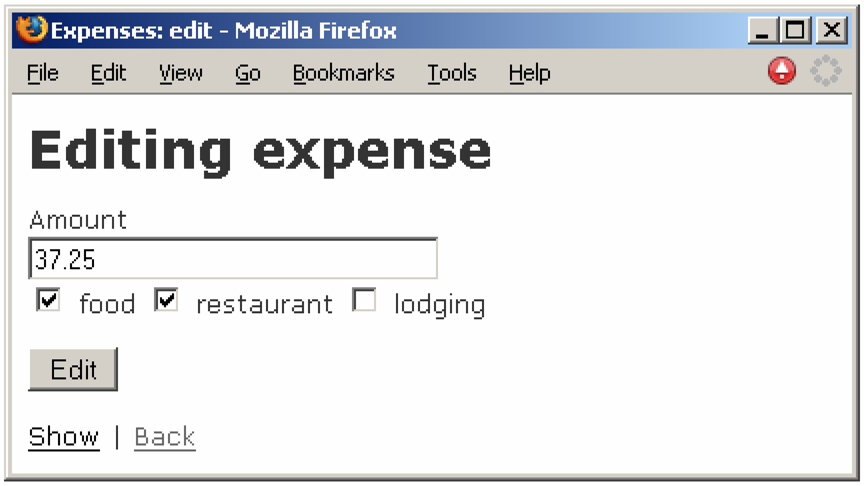

Expenses: list - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:3000/expenses/list
Go
Done

Listing expenses

Amount	tags			
37.25	food restaurant	Show	Edit	Destroy

New expense

Edit the new expense to see stored tags as checked.


Expenses: edit - Mozilla Firefox
File Edit View Go Bookmarks Tools Help

Editing expense

Amount

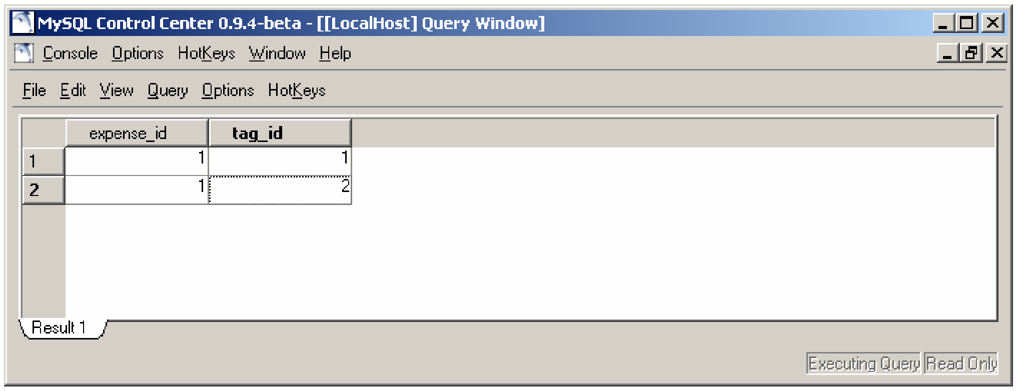
37.25

[✓] food [✓] restaurant [ ] lodging

Edit

Show | Back

Check out the database entries in the *expenses_tags* table.


MySQL Control Center 0.9.4-beta - [[LocalHost] Query Window]
Console Options HotKeys Window Help
File Edit View Query Options HotKeys
Result 1
Executing Query
Read Only

	expense_id	tag_id
1	1	1
2	1	2

Thanks

August 9, 2005 - Sheldon Hearn noticed a transactional condition with the database. Where @expense.tags.clear and @expense.tags<<Tag.find(params[:tag_ids]) should be replaced with a single @expense.tags=Tag.find(params[:tag_ids])

August 9, 2005 – Ecow pointed out multiple documentation errors where I failed to provide the tags table schema, missing code for assigning attributes on expense creation, and multiple incorrect references to view and controller methods.

August 11, 2005 – Spiralis pointed out that when editing existing tags … the existing tags should be checked.

August 17, 2005 – Brian NG suggested a fix for allowing existing tags to be checked.

August 25 – Brandt proposed a solution to the *nil object* error received when the user doesn't select a tag.