BASIC STEPS TO START WORKING WITH A RAILS APPLICATION AND DB2

1. Create a Rails application called 'Hello': `rails hello`
2. Enter the 'hello' folder that was just generated: `cd hello`
3. From the DB2 Control Center (`db2cc.exe`) create the database `blog_dev`. If you prefer you can do this through the DB2 command line (`db2cmd; db2 create database blog_dev`)
4. Edit `config\database.yml` with the proper credentials and a schema of your choice (don't use tabs, and leave a space between the parameters and their values):

```
development:
  adapter: ibm_db2
  database: blog_dev
  username: db2admin
  password: db2password
  schema: blog
```

5. Generate a model for Post: `ruby script\generate model Post`
6. Generate a model for Comment: `ruby script\generate model Comment`
7. Edit `db\migrate\001_create_posts.rb`, in order to indicate the structure for the Posts table:

```
class CreatePosts < ActiveRecord::Migration
  def self.up
    create_table :posts do |t|
    t.column "title", :string, :null => false
    t.column "body", :text, :null => false
    t.column "author", :string, :limit => 50, :null => false
    t.column "email", :string
    t.column "created_at", :datetime, :null => false
    t.column "updated_at", :datetime, :null => false
    end
 end

  def self.down
    drop_table :posts
  end
end
```

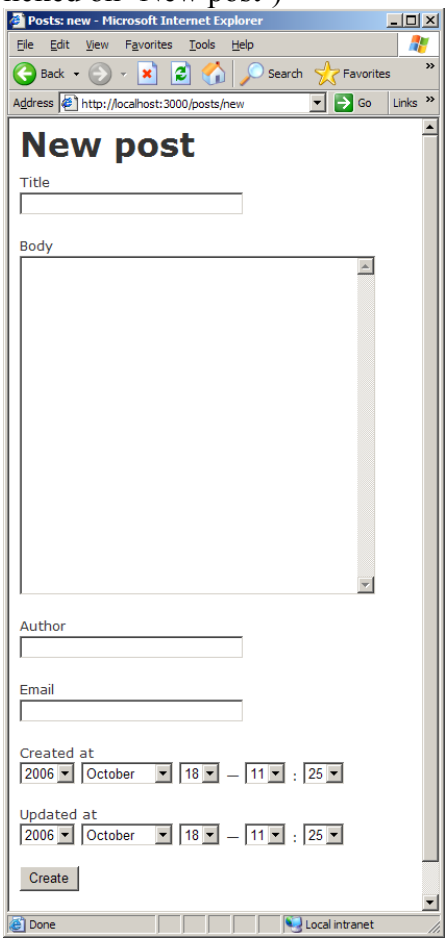
8. Edit db`\migrate\002_create_comments.rb` in order to specify the structure for the Comments table:

```
class CreateComments < ActiveRecord::Migration
  def self.up
    create_table :comments do |t|
    t.column "title", :string, :null => false
```

```
      t.column "body", :text, :null => false
      t.column "author", :string, :limit => 50, :null => false
      t.column "created_at", :datetime, :null => false
      t.column "updated_at", :datetime, :null => false
      t.column "post_id", :integer, :null => false
      end
    end

    def self.down
      drop_table :comments
    end
  end
```

9. Migrate the database with: `rake db:migrate`. This will create the above tables and an additional `schema_info` table within `blog_dev`. schema_info is used to store the current migration version.
10. Generate scaffold for the Posts table with:
    `ruby script\generate scaffold Post`
11. Run the WEBrick server with: `ruby script\server` and point your browser to http://localhost:3000/posts/
12. At this point, you should be able to see a basic front-end for your post table (in the picture we already clicked on 'New post')



13. You may want the created_at and updated_at fields to be handled automatically by Rails, without having to select the proper date and time. To do this, edit the partial `app\views\posts\_form.rhtml` and remove:

```
<p><label for="post_created_at">Created at</label><br/>
<%= datetime_select 'post', 'created_at'  %></p>

<p><label for="post_updated_at">Updated at</label><br/>
<%= datetime_select 'post', 'updated_at'  %></p>
```

14. In order to specify the relationship between the Post and Comment models, you can edit `app\models\post.rb` and insert `has_many :comments` within the class. For Comment, insert `belongs_to :post` within `app\models\comment.rb`.
15. At this point you can customize the application as you wish. Feel free to experiment (e.g. use `ruby script\console`) and remember that the relationships we specified in step 14 will now enable you to directly access the post for a given comment or the comments for a given post (e.g. @comment.post and @post.comments, where @post and @comment are two instances of Post and Comment respectively)

Enjoy!

Need help? http://www.alphaworks.ibm.com/tech/db2onrails/forum