

CS169 Week 4 Section Worksheet

Let's make a Rails app from the bottom up!

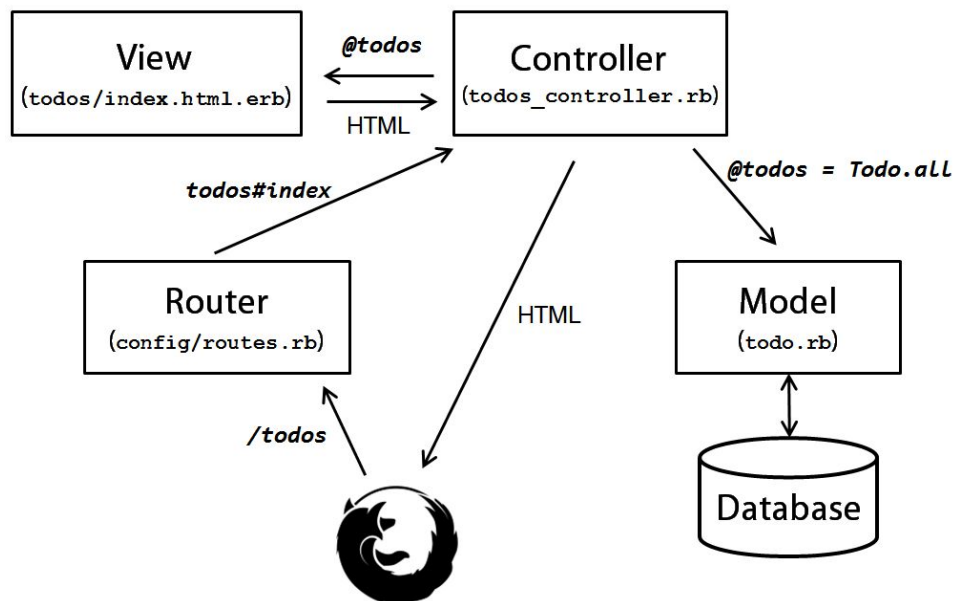
```
rails new app_name
```

```
rails new todo_app
```

```
rails generate scaffold model attribute:type
```

```
rails generate scaffold todo description:string
```

- What does this make for us?
- **MVC in rails**



Create the database for our app

```
bundle exec rake db:migrate
```

- Without this, no changes will be made to our database schema

Insert some records in the database

Open `db/seeds.rb`

Insert the following two todo items into the database using `Todo.create`:

```
Todo.create(description: "start the cs169 discussion section")
```

```
Todo.create(description: "release the third homework")
```

Use the following command to insert the data into database

```
rake db:seed
```

Check if the records are inserted into the database (use `rails console`)

Start the server:

```
rails server -p $PORT -b $IP
```

- Use rake routes to see what was created by the scaffold command
- Navigate to some of the routes to see what we have without writing a single line of code!

Extend the app

```
rails generate migration AddAttributeToModel attribute:type
```

```
rails generate migration AddDueDateToTodo due_date:datetime
```

- Note the format of command - this will generate the entire migration file automatically if we follow convention
- Make sure you run `rake db:migrate` again!

Create new route in the app

- Add a route in config/routes.rb and map it to a controller action

```
get '/hello', to: 'todos#hello'
```

- Add a view - what should it be called according to convention?

```
hello.html.erb
```

```
<h1>Hello!</h1>
```

- Add the corresponding method to a controller - do we need anything else in our method?

```
def hello
```

```
  respond_to do |format|
```

```
    format.html { render :hello }
```

```
    format.json { render json: "hello world!" }
```

```
  end
```

```
end
```

Possible group tasks

- Add a new attribute to the Todo model and update the views to display/edit the new field

Q: we want to add a new boolean field with name “done” for each Todo record

(default value: false), user should be able to view and edit the new attribute.

A: Make the following changes:

1. `rails generate migration add_done_to_todos done:boolean`

2. change the migrate file into:

```
class AddDoneToTodos < ActiveRecord::Migration
  def change
    add_column :todos, :done, :boolean, default: false
  end
end
```

3. `bundle exec rake db:migrate`

4. use `rails console` command to check if the change has been integrated

5. add the following content into `app/views/todos/_form.html.erb`

```
<%= f.label :done %>
<%= f.check_box :done %><br>
```

6. change `index.html.erb` and `show.html.erb` respectively
7. and change `todos_controller.rb`:

```
# Never trust parameters from the scary internet,
# only allow the white list through.
def todo_params
  params.require(:todo).permit(:description, :due_date,
:done])
end
```

- Add a page that displays all Todos we have sorted by due date
 - Change the routing schema
- Q:** route `/new_todo` to the page that creates a new Todo item.
- A:** add the following code into the `config/routes.rb` file
- ```
get '/new_todo', to: 'todos#new'
```
- route `todos#index` as the homepage for this app.
  - `root 'todos#index'`

## Optional

Commands for reversing back to a previous version of the DB:

```
rake db:migrate VERSION=0
```

Use the following command to show all migrations:

```
rake db:migrate:status
```

Console output:

| Status | Migration ID          | Migration Name       |
|--------|-----------------------|----------------------|
| up     | 20160211013318        | Create todos         |
| up     | <b>20160211025643</b> | Add done to todos    |
| up     | 20160211043246        | Add due date to todo |

Now rollback to a previous version:

```
rake db:migrate VERSION=20160211025643
```

There's a checklist in the appendix of the SaaS book whenever you make a new Rails app. Check it out!