hci+d lab.

Ruby on Rails Workshop

Ruby on Rails 02

Joonhwan Lee
human-computer interaction + design lab.

오늘 다룰 내용

- Rails 4.0 Update
- Rails application 생성
- Model 생성 및 데이터의 입력
- Controller 생성
- View 생성

Rails 4.0 Update

4.0에서 바뀐 부분

- Rails 3.x
  - 데이터를 입력 혹은 수정 시
    - Model 의 각 field (attribute) 들은 모두 protect 되어 있기 때문에 Controller 의 new 혹은 edit 액션에서 해당 field 를 억세스 할 수 있게 해주어야 함.
      - Model (e.g., `person.rb`) 에 다음과 같은 한 줄 추가
      - **`attr_accessible :name, :cell_no, :email, :address`**
- Rails 4.x
  - `attr_accessible` → deprecated
  - old method 사용하려면
    - Gemfile 에 다음의 라인 추가한 후 터미널에서 **bundle install 실행**
    - `gem 'protected_attributes'`

Recommendation for 4.x

- 다음의 코드를 controller 에 추가
  - ```
    private
      def person_params
        params.require(:person).permit(:name,
        :cell_no, :email, :address)
      end
    ```
  - access 하고 싶은 추가 field 가 있으면 위의 코드에 추가
- create 와 update 액션의 내용을 다음과 같이 수정
  - create
    - `@person = Person.new(person_params)`
  - update
    - `@person.update(person_params)`
    ```

Ruby on Rails 로 온라인 주소록 만들기

온라인 주소록 만들기

- 목표
  - contact 를 생성하고, 이름, 전화번호, 주소를 저장한다
  - contact 의 정보를 갱신/삭제한다
  - contact 정보를 저장하기 위해 database 를 생성한다

Rails 시작

- 어플리케이션의 생성

  ```
  > rails new AddressBook
  ```

- 어플리케이션 시작

  ```
  > cd AddressBook
  > rails server (WEBrick 서버구동)
  ```

- 브라우저에서 어플리케이션 주소를 입력

  http://localhost:3000/

데이터 베이스 설계

- Person Table
  - name: string
  - cell_no: string
  - email: string
  - address: string

name	cell_no	email	address
박OO	010-123-1234	park@...	...
강OO	010-222-2222	kang@...	...
전OO	010-321-4321	jun@...	...

Model 생성

- Person table 과 연결되는 모델 생성

  `> rails generate model Person`

- `db/migrate/2015xxxxxxxx_create_people.rb` 내용 수정

```
def change
  create_table :people do |t|
    t.string      :name
    t.string      :cell_no
    t.string      :email
    t.string      :address
    t.timestamps
  end
end
```

Model 생성

- Migration: “이주” 작업
  - rails 에서 정의해 놓은 db 스트럭춰 대로 실제 데이터베이스에 테이블을 생성하는 과정
- db 구조 확인

```
> cd db
> sqlite3 development.sqlite3
sqlite> .table   → list of tables
sqlite> .schema people
...
sqlite> .exit
```

Model 생성

- Migration

```
> rake db:migrate
```

- migration 후 mysql db 구조 확인

```
sqlite> .table
people  schema_migrations
sqlite> .schema people
CREATE TABLE "people" ("id" INTEGER
PRIMARY KEY AUTOINCREMENT NOT NULL, "name"
varchar(255), "cell_no" varchar(255),
"email" varchar(255), "address"
varchar(255), "created_at" datetime NOT
NULL, "updated_at" datetime NOT NULL);
```

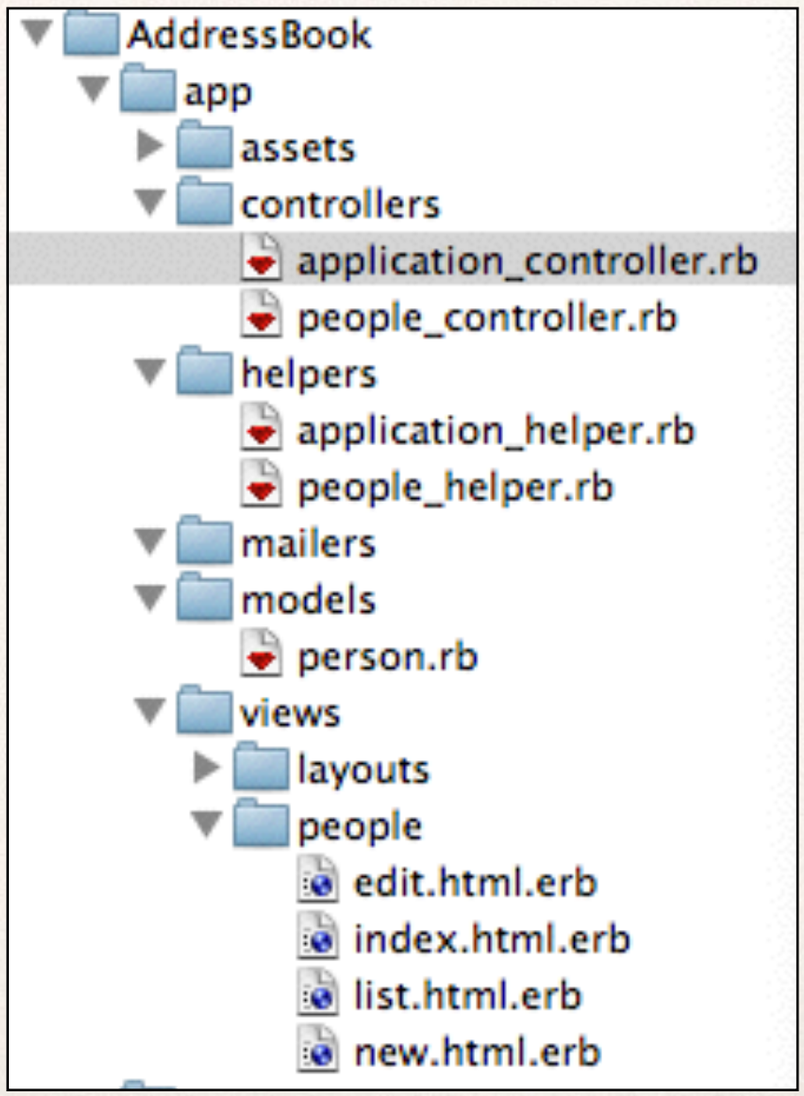
Controller 생성

- 컨트롤러: db (model) 과 view (html 페이지) 를 연결하는 역할
- 생성할 컨트롤러의 구조
  - people (주소록 컨트롤러)
    - list : 주소 리스트를 보여줌
    - new : 새로운 주소의 입력
    - edit : 주소의 수정
- 보다 복잡한 구조의 웹사이트
  - login, admin, user 등 다양한 컨트롤러 필요

Controller 생성

- 컨트롤러 생성 (index, list, new, edit 액션도 같이 생성)

```
> rails generate controller people index list
new edit
```



Controller 수정

- action 생성
  - list, new, edit action 생성

```
def index
end
```

→ http://localhost:3000/people/index

```
def list
end
```

→ http://localhost:3000/people/list

```
def new
end
```

→ http://localhost:3000/people/new

```
def edit
end
```

→ http://localhost:3000/people/edit

Controller 수정

- action 테스트
  - http://localhost:3000/people/new

    People#new

    Find me in app/views/people/new.html.erb

- `new.html.erb` 파일 내용 수정
  - 다른 action 도 테스트!

Controller 수정

- open people_controller.rb

```
def list
  @people = Person.all
end
```

- table_name.find(조건) 명령어 → 해당 테이블에서 데이터를 가져 옴. (find.all → 전부 가져올 때)

  ==> select * from table Person;

  ==> .all → 전부 (*) 가져오는 경우에 사용

- list view 의 수정
  - list.html.erb

Rails Console

- 모델, 콘트롤러의 내용을 터미널에서 테스트

```
> rails console
> @people = Person.all
Person Load (0.2ms)  SELECT
"people".* FROM "people"
 => []
```

View 수정

- list view 의 수정

```
<% for person in @people %>
<tr>
  <td><%= person.name %></td>
  <td><%= person.cell_no %></td>
  <td><%= person.email %></td>
  <td><%= person.address %></td>
  <td><%= link_to '수정', people_edit_path(:id =>
person) %></td>
</tr>
<% end %>

<%= link_to '새 주소록 등록', people_new_path %>
```

- View Test: http://localhost:3000/people/list

Data 입력

- Sample Data 를 rails console을 이용해서 입력!
  - 이름: Laura
  - 전화번호: 234-5678
  - 이메일: laura@mail.com
  - 주소: address 12, seoul, korea
- How?
  - Person Class 를 이용해서 Person 객체를 하나 만든다.
  - 객체에 값을 입력
  - 객체를 db에 저장

Data 입력

- How?
  - Person Class 를 이용해서 Person 객체를 하나 만든다.

    ```
    person = Person.new
    ```

  - 객체에 값을 입력

    ```
    person.name = 'Laura'
    person.cell_no = '234-5678'
    person.email = 'laura@mail.com'
    person.address = 'address 12, seoul,
    korea'
    ```

  - 객체를 db에 저장

    ```
    person.save
    ```

Data 입력

- Ruby Console 을 통해 입력해 보자.

```
AddressBook $ rails console
> person = Person.new
=> #<Person id: nil, name: nil, cell_no:
nil, email: nil, address: nil,
created_at: nil, updated_at: nil>

> person.name = 'Laura'
> person.cell_no = '234-5678'
> person.email = 'laura@mail.com'
> person.address = 'address 12, seoul,
korea'
```

Data 입력

- Ruby Console 을 통해 입력해 보자.

```
> person
=> #<Person id: 1, name: "Laura",
cell_no: "234-5678", email:
"laura@email.com", address: "seoul,
korea", created_at: "2012-11-28
21:04:45", updated_at: "2012-11-28
21:04:45">
```

Data 입력

- Ruby Console 을 통해 입력해 보자.

```
> person.save
   (0.1ms)  begin transaction
   (0.8ms)  UPDATE "people" SET
"address" = 'address 12, seoul,
korea', "updated_at" = '2012-11-28
21:11:43.900584' WHERE "people"."id"
= 1
   (2.5ms)  commit transaction
 => true
```

new action

- def new ... end 를 사용해서 데이터 입력 가능
- def new ... end 수정

```
def new
  @person = Person.new
end
```

- config/route.rb 에 다음의 행 추가

```
resources :people
```

→ people 데이터 베이스에 있는 모든 테이블과 데이터에 억세스 하기 위한 명령어.

View 수정

- open new.erb

```
<%= form_for(@person) do |f| %>
    <label>Name: </label>
    <%= f.text_field :name %><br>
    <label>Cellphone Number: </label>
    <%= f.text_field :cell_no %><br>
    <label>Email: </label>
    <%= f.text_field :email %><br>
    <label>Address</label>
    <%= f.text_field :address %><br>
    <%= f.submit %>
<% end %>
<%= link_to '돌아가기', people_path %>
```

데이터 입력

- 다음과 같이 데이터 입력

- 에러 메시지 발생!

```
Unknown action
The action 'create' could not be
found for PeopleController
```

Controller 수정

- new view 에 있는 form 은 submit 을 할 때 create 액션을 찾는다 → 사용자로 부터 입력받은 데이터를 데이터베이스에 저장하기 위한 목적
- people controller 에 create 액션 추가
- 입력을 마치면 (저장하면) list 액션으로 이동
- open people_controller.rb

```
def create
  @person = Person.new(person_params)
  @person.save
  redirect_to people_list_path
end
```

Controller 수정

- Error!

NameError in PeopleController#create

undefined local variable or method `person_params' for #<PeopleController:0x007fd1fb597818>

Extracted source (around line **#14**):

```
12
13   def create
14     Person.create(person_params)
15     redirect_to people_list_path
16   end
17
```

```
Rails.root: /Users/joonhwan/Desktop/workshop_apps/AddressBook
```

Application Trace | Framework Trace | Full Trace

app/controllers/people_controller.rb:14:in `create'

- Can't mass-assign protected attributes: name, cell_no, email, address

Model 수정

- 모델이 가지고 있는 4개의 필드 → protected되어 있음.
- 해당 필드를 프로그램의 여러 function 들이 접근할 수 있도록 바꾸어 주어야 함.
- People Controller에 다음을 추가

```
private
def person_params
 params.require(:person).permit
  (:name, :cell_no, :email, :address)
end
```

Controller 수정

- open people_controller.rb

```
def edit
  @person = Person.find(params[:id])
end
def update
  @person = Person.find(params[:id])
  @person.update(person_params)
  redirect_to people_list_path
end
```

- edit view 의 추가
  - new.html.erb 를 복사하여 edit.html.erb 생성

    → 코드의 중복??

new view 의 수정

- 중복되는 코드 → form 부분을 떼어내서 new.html.erb 와 edit.html.erb 에서 동시에 사용하도록 하자!
  - `<%= form_for(@person) do |f| %>`
    `...`
    `<% end %>`
    부분을 카피하여 _form.html.erb 로 저장
- new.html.erb 과 edit.html.erb 의 해당 부분을 다음의 내용으로 대체
  - `<%= render "form" %>`

index 만들기

- http://localhost:3000/people 로 접속하면 default로 index 액션으로 접근.
- 현재 index 액션은 비어있는 상태 → list 로 자동으로 redirect 될 수 있도록 수정.
- ```
  def index
    redirect_to :action => "list"
  end
  ```
  ```

Show 액션 만들기

- 클릭한 contact의 보다 자세한 정보를 보고 싶을 경우
  - eg. show blog article
- People Controller 에 다음의 내용 추가
  - ```
    def show
      @person = Person.find(params[:id])
    end
    ```
- view/people 아래에 show.html.erb 추가
- config/routes.rb 에 다음의 내용 추가
  - ```
    get "people/show"
    ```

Show 액션 만들기

- list.html.erb 수정
  - 주소 목록에서 바로 개별 주소록 보기로 넘어갈 수 있도록 링크를 만든다.
  - 다음의 내용 추가
  - ```
    <td>
      <%= link_to '수정', people_edit_path(:id => person) %> |
      <%= link_to '보기', people_show_path(:id => person) %>
    </td>
    ```
    ```

Show 액션 만들기

- show.html.erb 수정
  - 다음의 내용 추가
  -

```
<h1><%= @person.name %></h1>
<hr>
<p>
  전화번호: <%= @person.cell_no %><br>
  이메일: <%= @person.email %><br>
  주소: <%= @person.address %>
</p>
<%= link_to '돌아가기', people_path %>
```

Summary

- rails application 생성
- database 생성 및 migration
- controller 와 view 의 생성
- database (model) 과 controller 의 연결
- 사용자가 view 에 data 입력 → controller가 data 수집
  → database에 입력, 수정
- Controller 가 data 를 불러와서 view 를 통해 출력

Questions?