Ruby On Rails - Models, ActiveRecords, & ActiveRelations

Sachin

June 26, 2016

1 Theory

- *active record*: Design patterm for relational databases.
- *ActiveRecord*: Rails implementation of active record pattern.
- Retrive and manipulate data as objects, not as static rows.
- *Active Relations*: ORMs

2 Basic commands

- New app

  ```
  rails new <app> [-d [mysql]]
  ```

- New controller

  ```
  rails generate controller <controller_name> <view> <view> ..
  ```

- New model

  ```
  rails generate model <model-name-in-camelcase>
  ```

- Dump database schema

  ```
  rake db:schema:dump
  ```

- Rake – ruby make
  - List all tasks

    ```
    rake -T
    ```

  - List task related to db

    ```
    rake -T db
    ```

  - dump dev/prod database schema

    ```
    rake db:schema:dump RAILS_ENV=production
    rake db:schema:dump RAILS_ENV=development
    ```

  - Migrations

    ```
    rails generate migration <migrations-name>
    ```

  - create table

    ```
    rails generate model <name>
    ```

  - run migration(default will be on development)

    ```
    rake db:migrate
    ```

  - drop migration(to VERSION=0, before any migrations were ran)

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

  - Migration status

```
$ rake db:migrate:status

database: /home/psachin/course/ror-yt/simple_cms/db/development.sqlite3

 Status   Migration ID    Migration Name
--------------------------------------------------
  down    20160622121918  Do nothing yet
  down    20160622122201  Create users
```

– run migrations with versions

```
$ rake db:migrate VERSION=20160622121918
== 20160622121918 DoNothingYet: migrating =====================================
== 20160622121918 DoNothingYet: migrated (0.0000s) ============================

$ rake db:migrate VERSION=20160622122201
== 20160622122201 CreateUsers: migrating ======================================
-- create_table(:users)
-> 0.0009s
== 20160622122201 CreateUsers: migrated (0.0009s) =============================

$ rake db:migrate:down VERSION=20160622121918
== 20160622121918 DoNothingYet: reverting =====================================
== 20160622121918 DoNothingYet: reverted (0.0000s) ============================

$ rake db:migrate:status
database: /home/psachin/course/ror-yt/simple_cms/db/development.sqlite3

 Status   Migration ID    Migration Name
--------------------------------------------------
  down    20160622121918  Do nothing yet
   up     20160622122201  Create users
```

3 Active Records

- Rails console

```ruby
subject = Subject.new
# Check if the record exist in db or not(new record)
subject.new_record?             # true

# Create new record
eng = Subject.new(:name => "Eng", :position => 1, :visible => true)
eng.save
eng.new_record?             # false

# Create new inst, add values, and save
math = Subject.create(:name => "Math", :position => 2)

# find by id
find_math = Subject.find(1)
find_math.name = "English"
find_math.save

# Update multiple atts
find_math.updated_attribute(:name => "Maths", :visible => true)

# Delete a record
find_math.destroy

visible_subjects = s = Subject.where(:visible =>  true)
```

```ruby
visible_subjects.class          # type
visible_subjects.to_sql         # SQL query

Subject.where(:visible =>  true).where(:position => 1)
Subject.where(:visible =>  true, :position => 1).first.name

# ASCending order
Subject.where(:visible =>  true).order("position ASC")

# DESCending order, limited to one entry
Subject.where(:visible =>  true).order("position DESC").limit(1).offset(1)
```

- Named scopes

```ruby
class Subject < ActiveRecord::Base

  # Similar to model manager in Dango
  scope :visible, lambda { where(:visible => true)}
  scope :invisible, lambda { where(:visible => false)}

  scope :sorted, lambda { order("subjects.position ASC")}
  scope :newest_first, lambda { order("subjects.created_at DESC")}

  scope :search, lambda {|query|
    where(["name LIKE ?", "%#{query}%"])
  }
end
```

  - Queries

```ruby
Subject.visible
Subject.newest_first
Subject.search("Math")
```

4 Associations

- One-to-one
  - Classroom `has_one :teacher`
  - Teacher `belongs_to :classroom`
- One-to-many
  - Teacher `has_many :courses`
  - Course `belongs_to :teacher`
  - Photographer `has_many :photographs`
- Many-to-many
  - Course `has_and_belongs_to_many :students`
  - Student `has_and_belongs_to_many :courses`

5 Links

- Youtube