



RUBY ON RAILS BEGINNERS

4. Model jako
reprezentacja zasobu

Praca domowa

setter vs zmienna

```
class Rock

  def initialize weight
    weight # zmienna lokalna (argument)
    @weight # zmienna instancyjna
    self.weight # getter

    weight = weight # nie ma sensu
    @weight = weight # przypisanie do zmiennej
    self.weight = weight # setter
  end

  # attr_reader :weight
  def weight
    puts 'using getter now!'
    @weight
  end

  # attr_writer :weight
  def weight= value
    puts 'using setter now!'
    @weight = value
  end
end
```

Generacja

```
$ rails g model rock
Running via Spring preloader in process 11069
  invoke  active_record
  create  db/migrate/20181121112710_create_rocks.rb
  create  app/models/rock.rb
  invoke  test_unit
  create  test/models/rock_test.rb
  create  test/fixtures/rocks.yml
```

Migracje

Baza danych

```
# Gemfile

# Use sqlite3 as the database for Active Record
gem 'sqlite3'
# gem 'pg'
# gem 'mysql2'
```

Tworzenie tabeli

```
# config/db/migrate/20181121112710_create_rocks.rb

class CreateRocks < ActiveRecord::Migration[5.2]
  def change
    create_table :rocks do |t|

      t.timestamps # created_at i updated_at
    end
  end
end
```

Tworzenie tabeli

```
# config/db/migrate/20181121112710_create_rocks.rb

class CreateRocks < ActiveRecord::Migration[5.2]
  def change
    create_table :rocks do |t|
      # t.type :column, **options
      t.float :weight, null: false
      t.timestamps # created_at i updated_at
    end
  end
end
```




0 vs NULL

Migrate

```
$ rake db:migrate
== 20181121112710 CreateRocks: migrating =====
-- create_table(:rocks)
   -> 0.0021s
== 20181121112710 CreateRocks: migrated (0.0022s) ==
```

Schema

```
# db/schema.rb

ActiveRecord::Schema.define(version: 2018_11_21_112710) do

  create_table "rocks", force: :cascade do |t|
    t.float "weight", null: false
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
  end
end
```

Generacja

```
$ rails g migration AddNicknameToRock
Running via Spring preloader in process 12429
  invoke  active_record
  create  db/migrate/20181121122127_add_nickname_to_rock.rb
```

Dodanie kolumny

```
# db/migrate/20181121122127_add_nickname_to_rock.rb

class AddNicknameToRock < ActiveRecord::Migration[5.2]
  def change
  end
end
```

Dodanie kolumny

```
# db/migrate/20181121122127_add_nickname_to_rock.rb

class AddNicknameToRock < ActiveRecord::Migration[5.2]
  def change
    # operation :table, **options
    add_column :rocks, :nickname, :string
  end
end
```

Migracja

```
$ rake db:migrate
== 20181121122127 AddNicknameToRock: migrating =====
-- add_column(:rocks, :nickname, :string)
   -> 0.0071s
== 20181121122127 AddNicknameToRock: migrated (0.0073s) ==
```

Schema

```
# db/schema.rb

ActiveRecord::Schema.define(version: 2018_11_21_122127) do

  create_table "rocks", force: :cascade do |t|
    t.float "weight", null: false
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.string "nickname"
  end
end
```


Przydatne komendy

```
$ rake db:create # stwórz  
$ rake db:migrate # migruj  
$ rake db:rollback # cofnij  
$ rake db:drop # usuń  
$ rake db:reset # stwórz od nowa
```

Model

Definicja

```
# app/models/rock.rb  
  
class Rock < ApplicationRecord  
end
```

Definicja

```
$ rails c
Running via Spring preloader in process 12898
Loading development environment (Rails 5.2.1)
2.3.1 :001 > rock = Rock.new
=> #<Rock id: nil, weight: nil, created_at: nil, updated_at: nil, nickname: nil>
2.3.1 :002 >
```

Zapisywanie

```
2.3.1 :001 > rock = Rock.new
=> #<Rock id: nil, weight: nil, ..., nickname: nil>
2.3.1 :002 > rock.save
(0.1ms) begin transaction
Rock Create (0.8ms) INSERT INTO "rocks" (...)
(0.1ms) rollback transaction
ActiveRecord::NotNullViolation: (...)
```

Zapisywanie

```
2.3.1 :003 > rock.weight = 1.5
=> 1.5
2.3.1 :004 > rock.save
(0.1ms) begin transaction
Rock Create (0.8ms) INSERT INTO "rocks" (...)
(9.1ms) commit transaction
=> true
```

Zapisywanie

```
2.3.1 :005 > Rock.create(weight: 3.2, nickname: 'Bob')  
  (0.1ms)  begin transaction  
  Rock Create (0.8ms)  INSERT INTO "rocks" (...)  
  (9.1ms)  commit transaction  
=> #<Rock id: 2, weight: 3.2, ..., nickname: "Bob">
```

Wybieranie

```
2.3.1 :005 > Rock.all
Rock Load (0.4ms)  SELECT  "rocks".* FROM "rocks" LIMIT ?  [["LIMIT", 11]]
=> #<ActiveRecord::Relation [#<Rock id: 1, ...,>, #<Rock id: 2, ...,>]]>
```


Wybieranie

```
Rock.find_by(nickname: 'Bob') # znajduje pierwszy  
Rock.where.not(nickname: nil) # filtruje  
Rock.where(created_at: (Time.now.yesterday..Time.now))  
Rock.order(updated_at: :desc) # sortuje
```

Zmiana

```
2.3.1 :006> Rock.first.update(weight: 9.2)
Rock Load (0.6ms)  SELECT "rocks".* FROM "rocks" (...)
(0.2ms)  begin transaction
Rock Update (0.6ms)  UPDATE "rocks" SET (...)
(2.9ms)  commit transaction
=> true
```

Usuwanie

```
2.3.1 :007> Rock.first.delete
Rock Load (0.2ms)  SELECT "rocks".* FROM "rocks" (...)
Rock Destroy (2.5ms)  DELETE FROM "rocks" WHERE "rocks"."id" = ?  [["id", 1]]
=> #<Rock id: 1, weight: 9.2, ..., nickname: nil>
```

reload!

```
2.3.1 :007> reload!  
Reloading...  
=> true
```

Walidacje

Istnienie

```
# app/models/rock.rb  
  
class Rock < ApplicationRecord  
  validates :weight, presence: true  
end
```

Istnienie

```
rock = Rock.new
```

```
rock.valid?  
#=> false
```

```
rock.errors.messages  
#=> {:weight=>["can't be blank"]}
```

```
rock.save  
#=> false
```

Numeryczność

```
# app/models/rock.rb
```

```
class Rock < ApplicationRecord
  validates :weight, presence: true, numericality: { greater_than: 0 }
end
```


Numeryczność

```
rock.weight = -3
```

```
rock.valid?  
#=> false
```

```
rock.errors.messages  
#=> {:weight=>["must be greater than 0"]}
```

```
rock.save  
#=> false
```

Custom

```
# app/models/rock.rb

class Rock < ApplicationRecord
  validates :weight, presence: true, numericality: { greater_than: 0 }
  validate :luck

  def luck
    errors.add(:base, 'bad luck') if rand(2).zero?
  end
end
```

Custom

```
rock = Rock.new(weight: 3)
```

```
rock.valid?  
#=> true
```

```
rock.valid?  
#=> false
```

```
rock.errors.messages  
#=> {:base=>["bad luck"]}
```



Thanks!