



SZKOLENIE RUBY ON RAILS

Michał Makaruk

Bez Railsów i Modelu

- Tworzymy klasę , która jest odpowiedzialna za łączenie się z bazą danych.
- Musimy w naszym kodzie ręcznie pisać zapytania SQL.

Model

Model odpowiedzialny jest za biznesowe dane i logikę.

Dane przechowywane są w bazie danych.

Przykładowym modelem może być model odpowiedzialny za przechowywanie danych o użytkownikach czy firmach.

Model

Modele w Rails reprezentowane są przez klasy dziedziczące po klasie bazowej ActiveRecord::Base.

Obiekty kodu modelu reprezentują rzeczy istniejące w domenie systemu – tak jak bilety w systemie biletów.

Przykład tworzenia modelu

```
MacBook-Pro-Micha:rails4app michalos$ rails g model User name:string email:string description:text
invoke active_record
create db/migrate/20130919174310_create_users.rb
create app/models/user.rb
invoke # test_unit
create test/models/user_test.rb
create test/fixtures/users.yml
```

Przykład tworzenia modelu 2

```
MacBook-Pro-Micha:rails4app michalos$ rails g model Book name:string author:string isbn:string
  invoke  active_record
  create  db/migrate/20130920062401_create_books.rb
  create  app/models/book.rb
  invoke  test_unit
  create  test/models/book_test.rb
  create  test/fixtures/books.yml
```

Rails g model nazwaModelu

- tworzy migracje bazy danych;
- tworzy klasę dziedziczącą po `ActiveRecord::Base`;
- tworzy testy;
- tworzy fixtures.

Przykładowe typy

- string (łańcuch znaków);
- text (dłuższy łańcuch znaków);
- decimal (liczba dziesiętna);
- date (data);
- boolean (wartość logiczna prawda bądź nieprawda).

Migracja

- migracja to skrypt Ruby;
- migracja odpowiedzialna jest za aktualizacje schematu bazy danych;
- Więcej informacji na temat migracji:
- <http://guides.rubyonrails.org/migrations.html>

Migracja

- Przykładowa migracja

```
class CreateUsers < ActiveRecord::Migration
  def change
    create_table :users do |t|
      t.string :name
      t.string :email
      t.text :description

      t.timestamps
    end
  end
end
```

Migracja

Wykonanie migracji

```
MacBook-Pro-Micha:rails4app michalos$ rake db:migrate
== CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0290s
== CreateUsers: migrated (0.0291s) =====
```

Podstawowe polecenia migracje

- Duzo rubiszona- polecenie `rake db:migrate` wywołuje wszystkie migracje, które nie zostały wywołane wcześniej
- polecenie `rake db:rollback` odwołuje ostatnią migracje
- polecenie `rake db:roolback STEP=3` odwołuje 3 ostatnie migracje

Migracja dodająca atrybut do modelu



```
MacBook-Pro-Micha:cwiczenia_rails michalos$ rails g migration addOccupationToClient occupation:string
invoke active_record
create db/migrate/20131007173028_add_occupation_to_client.rb
```

Create – tworzenie obiektu

```
>> p = Person.new(:name => "John Doe")  
=> #<Person id: nil, name: "John Doe", created_at:  
nil, :updated_at: nil>  
  
>> p.new_record?  
=> true  
  
>> p.save  
=> true  
  
>> p.new_record?  
=> false
```

Create – tworzenie obiektu

```
>> p = Person.new(:name => "John Doe")  
=> #<Person id: nil, name: "John Doe", created_at:  
nil, :updated_at: nil>  
  
>> p.new_record?  
=> true  
  
>> p.save  
=> true  
  
>> p.new_record?  
=> false
```

Create – tworzenie obiektu 2

```
>> p = Person.new  
>> p.name = "Adam"  
>> p.save  
=> true
```


Save

- metoda save zachowuje dane w bazie danych, kiedy dane są poprawne.

Read

- `User.last` – zwraca ostatni element
- `User.first` – zwraca pierwszy element
- `User.count` – zwraca ilość elementów
- `User.order(:name)` – zwraca uporządkowanego User po nazwie
- `User.limit(5)` – zwraca tylko 5 Userów.
- `User.where(name: "Michał")` – zwraca Wszystkich Userów o imieniu Michał

Read

Inne zapytania:

http://guides.rubyonrails.org/active_record_querying.html

Polecenie create

Zamiast pisać:

```
user = User.new  
user.name = "Michał"  
user.occupation = "programmer"  
user.save
```

Możemy po prostu napisać:

```
user = User.create(name: "Michał", occupation:  
"programmer")
```

Update

```
irb(main):005:0> o = Organization.last
  Organization Load (0.4ms)  SELECT "organizations".* FROM "organizations" ORDER
  BY "organizations"."id" DESC LIMIT 1
=> #<Organization id: 3, name: "Coderslab", number_of_employees: nil, address: n
il, created_at: "2013-09-19 16:33:55", updated_at: "2013-09-20 04:35:55">
irb(main):006:0> o.name = "Webventure"
=> "Webventure"
irb(main):007:0> o.save
  (0.2ms)  begin transaction
  SQL (1.3ms)  UPDATE "organizations" SET "name" = ?, "updated_at" = ? WHERE "or
ganizations"."id" = 3  [["name", "Webventure"], ["updated_at", Fri, 20 Sep 2013
04:36:27 UTC +00:00]]
  (1.6ms)  commit transaction
=> true
```

Delete

```
irb(main):001:0> o = Organization.last
Organization Load (0.3ms) SELECT "organizations".* FROM "organizations" ORDER
BY "organizations"."id" DESC LIMIT 1
=> #<Organization id: 3, name: "Webventure", number_of_employees: nil, address:
nil, created_at: "2013-09-19 16:33:55", updated_at: "2013-09-20 04:36:27">
irb(main):002:0> o.destroy
```

Scaffold

- Przy tworzeniu scaffoldu automatycznie generowany jest za nas model.
- Dodatkowo także widoki i kontrolery.

Seeds - ziarna

- Jest to plik , w którym tworzymy przykładowe dane bądź dane , które potrzebne są nam w środowisku produkcyjnym (na samym początku).
- plik znajduje się w db/seeds.rb
- plik wywołujemy za pomocą polecenia `rake db:seed`
- kilkukrotne wywołanie powoduje od nowa uruchomienie pliku.

Seeds przykładowe

Przykład pliku:

```
10.times do
```

```
  User.create(name: "Adam", occupation: "programmer")
```

```
End
```

```
puts "Users created"
```

Polecenie `destroy all`

Polecenie `Model.destroy_all` niszczy wszystkie obiekty w bazie danych.

Konsola

- Za pomocą konsoli możemy korzystać z języka Ruby plus mamy połączenie do bazy danych.
- Aby wywołać konsole wpisujemy rails c w katalogu projektu.

Tryb sandbox Konsola

- tryb sandbox używany jest wtedy kiedy nie chcemy robić zmian w bazie danych

```
MacBook-Pro-Micha:cwiczenia_rails michalos$ rails c --sandbox
Loading development environment in sandbox (Rails 4.0.0)
Any modifications you make will be rolled back on exit
irb(main):001:0>
```



SZKOLENIE RUBY ON RAILS

Michał Makaruk

Dziękuję za uwagę