

Ruby On Rails par Rspec

Cyril Mougel

24 novembre 2008



- 1 Rspec, BDD ?
- 2 Rails est agile avant tout
- 3 La couche model de Rails



Behaviour Driven Development

- Méthode Agile
- Extreme programming
- TDD (Test Driven Development)
- l'empire du 'should'



- 1 Rspec, BDD ?
- 2 Rails est agile avant tout
- 3 La couche model de Rails



Test : :Unit de base dans Rails

- Dossier test créé à chaque création de projet Rails
- Stats de nombre de LOC et LOC de test (TODO : mettre image ici)
- Possibilite de tester chaque couche de Rails



Mais pourquoi Rspec alors ?

- Rspec == BDD Framework
- Documentation générée plus claire que les Test : :Unit
- Réutilisation plus simple du comportement
- Ajout des Stories (Cucumber)



- 1 Rspec, BDD ?
- 2 Rails est agile avant tout
- 3 La couche model de Rails



Des classes de Mapping a la Base de donnée

```
1  class User < ActiveRecord::Base
2    has_many :products
3
4    validates_presence_of :name
5    validates_uniqueness_of :name
6    validates_format_of :email ,
7      :with => /\A([^\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\Z/i ,
8      :on => :create
9  end
10
11 class Product < ActiveRecord::Base
12   has_one :user
13 end
```



Doit créer les accesseurs sur les colonnes

```
1 describe User do
2
3   before(:each) do
4     @user = User.first
5   end
6
7   it 'should access to name' do
8     @user.name.should_not be_nil
9   end
10
11 end
```



Doit valider le model

```
1 describe User do
2   describe 'create' do
3     def user_valid(options)
4       User.new({:name => 'Cyril Mougel',
5                 :email => 'cyril.mougel@gmail.com'}).merge(options))
6     end
7
8     it 'should user valid' do
9       user_valid.should be_valid
10    end
11
12    it 'should presence of name' do
13      user_valid(:name => nil).should_not be_valid
14    end
15
16    it 'should unique name' do
17      user_valid.save
18      user_valid.should_not be_valid
19    end
20
21    it 'should not valid with bad email' do
22      user_valid(:email => 'cool').should_not be_valid
23    end
24  end
25 end
```



Doit chercher des données

```
1  describe User do
2
3    it 'should find by name' do
4      User.find_by_name('Cyril Mougel').should == users(:shingara)
5    end
6
7    it 'should find by email' do
8      User.find_by_email('cyril.mougel@gmail.com').should == users(:shingara)
9    end
10
11    it 'should find by name and email' do
12      User.find_by_name_and_email('Cyril Mougel',
13                                'cyril.mougel@gmail.com').should == users(:shingara)
14    end
15
16    it 'should find all' do
17      User.all.should == [users(:shingara), users(:underflow_)]
18    end
19  end
```



Doit avoir des associations

```
1  describe User do
2    before :each do
3      @user = users(:shingara)
4    end
5
6    it 'should have 2 products' do
7      @user.should have(2).products
8      # @user.products.size == 2
9    end
10
11    it 'should have book products' do
12      @user.products[0].should == products[:book]
13    end
14  end
15
16  describe Product do
17    before :each do
18      @product = products(:book)
19    end
20
21    it 'should have user shingara' do
22      @product.user.should == users(:shingara)
23    end
24  end
```

