Getting to know FactoryGirl

Sameera (Sam)

sameera207

reinteractive

Why we need sample data ?

Simple rails example

```ruby
# app/models/users.rb
class User < ActiveRecord::Base
    # name -> string
    # email -> string
    # confirmed_at -> date time
    has_many :books, foreign_key: 'author_id'
    validates :name, presence: true
end
```

```
# test/fixtures/users.yml
one:
  name: MyString
  email: MyString

two:
  name: MyString
  email: MyString
```

FactoryGirl

- factorygirl gem
- factorygirl rails, for rails projects

Best practices when creating a factory

- have a factory for each class (E.g spec/factories/user.rb)
- provide the bare minimal attributes for the factory

basic factory

```
FactoryGirl.define do
  factory :user do
    name 'Sameera'
    email 'sameera207@gmail.com'
    confirmed_at '2017-03-11 11:25'
  end
end

FactoryGirl.create :user
```

Overriding factory attributes

```
FactoryGirl.create :user, email: 'test@gmail.com'
```

Dynamic attributes with factories

```
FactoryGirl.define do
  factory :user do
    name 'Sameera'
    email 'sameera207@gmail.com'
    confirmed_at { 2.days.ago }
  end
end
```

create vs build vs attributes_for

- `create` (saves the instance)
- `build` (only build the instance and will not save)
- `attributes_for` returns hash of attributes

Aliases and Associations

```ruby
# app/models.user.rb
class User < ActiveRecord::Base
  has_many :books, foreign_key: 'author_id'
    validates :name, presence: true
end

# app/models/book.rb
class Book < ActiveRecord::Base
  belongs_to :author, class_name: 'User'
end
```

..updating factories

```ruby
# spec/factories/user.rb
FactoryGirl.define do
  factory :user, aliases: [:author] do
    name 'Sameera'
    email 'sameera207@gmail.com'
    confirmed_at { 2.days.ago }
  end
end

# spec/factories/book.rb
FactoryGirl.define do
  factory :book do
    author # association :author, factory: :user
        name "Book1"
  end
end
```

Transient Attributes

```
# spec/factories/user.rb
FactoryGirl.define do
  factory :user, aliases: [:author] do
        transient do
            display_name false
        end
    name { "sameera #{email if display_name}" }
    email 'sameera207@gmail.com'
    confirmed_at { 2.days.ago }
  end
end

FactoryGirl.create :user, display_name: true
```

Traits

```
# spec/factories/book.rb
FactoryGirl.define do
  factory :book do
    author
    name "Book1"

    trait :published do
      published true
    end

    trait :unpublished do
      published false
    end

    factory :published_book,    traits: [:published]
    factory :un_published_book,    traits: [:unpublished]
  end
end

FactoryGirl.create :published_book
```

Building or Creating Multiple Records

```
FactoryGirl.create_list :user, 5
```

Lets do...

```
class User < ActiveRecord::Base
  has_many :books, foreign_key: 'author_id'

  validates :name, presence: true, uniqueness: true
end
```

```
FactoryGirl.create_list :user, 5 # -> ActiveRecord::RecordInvalid: Validation failed: Name has already been taken
```

Generating values dynamically

- sequence
- using a separate gem (E.g faker)

faker

faker

```
FactoryGirl.define do
  factory :user, aliases: [:author] do
    name { Faker::Name.name  }
    email 'sameera207@gmail.com'
    confirmed_at { 2.days.ago }
  end
end
```

Thank you !!!