

Better Rails Active Record Validation Messages



Deepak Singh



Jul 17 '18 Updated on Aug 01, 2018 • 2 min read

[#rails](#) [#activerecord](#) [#validations](#) [#errormessages](#)

Validations are used to ensure that only valid data is saved to database and no business rules are violated. For example, it may be important for business to ensure that users provides a valid email address and phone number. Model-level validations are the most common way (db-level constraints is another option) to achieve this. Model-level validations are database agnostic, simple to test and maintain. Rails makes this more fun with built-in helpers and support for custom validations.

All that being done, the next crucial step is to provide users with immediate feedback as they use your site. Rails allows to do just that with the `:message` option. It specifies the

customised error message that will be added to the errors collection when the validation fails.

The `:message` option accepts a `string` or a `proc`.

- When using strings, rails provides placeholder/interpolation variables to help add context and reduce cognitive load.
 - `%{value}`
 - `%{attribute}`
 - `%{model}` Use them in the message string as,

```
validates :age, numericality: { message: "%{attribute} must be a number" }
```

Age must be a number feels much better than must be a number.

- Using procs adds more flexibility and logic to the message string. Proc has access to the object being validated and a data hash with keys `model`, `attribute` and `value`, same as the interpolation variable above.

```
validates :username,
  uniqueness: {
    message: ->(object, data) do
      "Hey #{object.name}!, #{data[:value]} is taken already!"
    end
  }
```

```

validates :score,
  presence: {
    message: ->(object, data) do
      if object.member?
        "something"
      else
        "Something else"
      end
    end
  end
}

```

Magic does not end here, rails also allows overriding default validation messages via i18n-localization files.

```

en:
  activerecord:
    errors:
      models:
        user:
          attributes:
            firstname:
              blank: "custom message, can\'t be blank"

```

Profile

To access the courses you need to complete your profile

All fields are mandatory

First Name



custom message, can't be blank

Active Record provides couple of namespaces where message

translations can be placed in order to provide different messages and translation for certain models, attributes, and/or validations. It also transparently takes single table inheritance into account.

Also placeholder/interpolation variables are still available.

```
en:
  activerecord:
    errors:
      models:
        user:
          attributes:
            firstname:
              blank: "Please fill in your %{attribute}"
```

<http://guides.rubyonrails.org>

Further, `ActiveModel::Errors#full_messages` prepends the attribute name to the error message using a separator that is looked up from `errors.format` (and defaults to `%{attribute} %{message}`). This too can be customised as

```
en:
  errors:
    # format to use in full error messages.
    format: "%{message} %{attribute}"
```

The software industry moves fast. But if you keep up, you can have an incredible career.

Join DEV. 100% Free Forever.



Deepak Singh + FOLLOW

Director, Eloquent Studio Pvt. Ltd. | Ruby on Rails, Go, React Developer

@spidergears spider_gears spidergears eloquent.studio

Add to the discussion



PREVIEW

SUBMIT

[code of conduct](#) - [report abuse](#)

Classic DEV Post from Dec 13 '19

What is the most inspiring tech talk of 2019 ?



Etienne Depaulis



72



11

Another Post You Might Like

How To Give Engineers Filtered Database Access



Molly Struve (she/her)

A common issue that many companies often face, including DEV, is how much data access to give their e...



120



10

Another Post You Might Like

No more 'new-to-opensource' excuses!



Arit Amana

As a newly-minted coding bootcamp grad, I was terrified of opensource. During bootcamp, I had tried t...



50



4



Common errors when learning rails!!!!

Edward Mendoza - Apr 10



Ruby and Ruby on Rails: uma conversa com o especialista

Fellipe Couto - Apr 8



É o fim do Ruby on Rails?

Fellipe Couto - Apr 8



Rails Scaffold

Nesha Zoric - Apr 9

[Home](#) [About](#) [Privacy Policy](#) [Terms of Use](#) [Contact](#) [Code of Conduct](#)

DEV Community copyright 2016 - 2020 🔥