daftcode

RUBY ON RAILS Level UP

3. Zaawansowane API

Praca domowa

RUBY ON RAILS
Level UP

Obsługa błędów

```ruby
def update
  shogun = Shogun.find(params[:id])
  shogun.update(shogun_attributes)
  if shogun.errors.empty?
    render json: shogun
  else
    render json: { errors: shogun.errors.messages }, status: 422
  end
rescue ActiveRecord::RecordNotFound
  render json: { error: error.message }, status: 404
end
```

RUBY ON RAILS
Level UP

```ruby
class ApplicationController
  def render_valid(object)
    if object.errors.empty?
      render json: object
    else
      render json: { errors: object.errors.messages }, status: 422
    end
  end
end

class ShogunController < ApplicationController
  def update
    shogun = Shogun.find(params[:id])
    shogun.update(shogun_attributes)
    render_valid(shogun)
  rescue ActiveRecord::RecordNotFound
    render json: { error: error.message }, status: 404
  end
end
```

Obsługa błędów

StandardError vs Exception

Obsługa błędów

update! create! find_by!

Obsługa błędów

rescue_from

```ruby
class ApplicationController
  rescue_from ActiveRecord::RecordInvalid, with: :render_validation_error
  rescue_from ActiveRecord::RecordNotFound, with: :render_not_found_error

  def render_validation_error(exception)
    render json: exception.record.errors, status: 422
  end

  def render_not_found_error(exception)
    render json: { error: exception.message }, status: 404
  end
end

class ShogunController < ApplicationController
  def update
    shogun.update!(shogun_attributes)
    render json: shogun
  end

  private

  def shogun
    @shogun ||= Shogun.find(params[:id])
  end
end
```

```ruby
module Errorable
  extend ActiveSupport::Concern

  included do
    rescue_from ActiveRecord::RecordInvalid, with: :render_validation_error
    rescue_from ActiveRecord::RecordNotFound, with: :render_not_found_error
  end

  def render_validation_error(exception)
    render json: exception.record.errors, status: 422
  end

  def render_not_found_error(exception)
    render json: { errors: exception.message }, status: 404
  end
end

class ApplicationController
  include Errorable
end
```

Obsługa błędów

500

Monitorowanie aplikacji

Captain Planet / Heart
ISSUES
EVENTS
OVERVIEW
USER FEEDBACK
RELEASES
Star
Subscribe
Settings
Unresolved Issues
Sort by: Last Seen
GRAPH:
24H
14D
EVENTS
USERS
Filter people
Chris Jennings

is:unresolved

Error TypeError poll(../../sentry/scripts/views.js)
Object [object Object] has no method 'updateFrom'
HEART-1D 6 days ago — 4 months old 1
26 1

Error ★ javax.servlet.ServletException org.hsqldb.jdbc.Util in throwError
Something bad happened
HEART-1G 6 days ago — 4 months old
26 1

Error script-src example.com
Blocked 'script' from 'example.com'
HEART-1K 6 days ago — 4 months old
26 1

Error ZeroDivisionError bin/raven in <main>
divided by 0
HEART-1H 6 days ago — 4 months old
26 1

Standard {JSON:API}

REST API

GET /castles/:id

```
{
   "id": 2,
   "clan": "Toyotomi",
   "name": "Osaka Castle",
   "lord": {
     "id": 2,
     "name": "Toyotomi Hideyori"
   }
 }
```

```
{
  "castle": {
    "id": 1,
    "clan": "Oda",
    "lord_id": 12,
    "name": "Azuchi Castle"
  }
 }
```

{JSON:API}

{JSON:API}

GET /clans/:id
Content-type: application/vnd.api+json

```json
{
  "data": {
    "id": 1,
    "type": "clans",
    "attributes": {
      "name": "Hojo"
    }
  }
}
```

{JSON:API}

Resource object

```
{
  "id": 1,
  "type": "clans",
  "attributes": {
    "name": "Hojo"
  }
}
```

GET /clans
Content-type: application/vnd.api+json

```json
{
  "data": [
    {
      "id": 1,
      "type": "clans",
      "attributes": {
        "name": "Hojo"
      }
    },
    {
      "id": 2,
      "type": "clans",
      "attributes": {
        "name": "Ashikaga"
      }
    }
  ]
}
```

GET /clans/:id
Content-type: application/vnd.api+json

```
{
  "data":
  {
    "id": 1,
    "type": "clans",
    "attributes": {
      "name": "Hojo"
    },
    "relationships": {
      "leader": {
        "data": {
          "type": "samurai",
          "id": 5
        }
      }
    }
  }
}
```

```
{
  "data":
  {
    "id": 1,
    "type": "clans",
    "attributes": {
      "name": "Hojo"
    },
    "relationships": {
      "leader": { "data": { "type": "samurai", "id": 5 }
      }
    },
    "included": [
      {
        "type": "samurai",
        "id": 5,
        "attributes": {
          "name": "Hojo Takatoki"
        }
      }
    ]
  }
```

{JSON:API}

Serializacja danych

Active Model Serializer

```ruby
gem 'active_model_serializers'
```

Active Model Serializer

```
rails g serializer shogun
app/serializers/shogun_serializer.rb
```

```ruby
class ShogunSerializer < ActiveModel::Serializer
  attributes :id, :name
end
```

Active Model Serializer

```ruby
class ShogunController
  def index
    render json: Shogun.all
  end

  def show
    render json: shogun
  end

  private

  def shogun
    @shogun ||= Shogun.find(params[:id])
  end
end
```

Active Model Serializer

```
class ShogunSerializer < ActiveModel::Serializer
  attributes :id, :name
  belongs_to :clan
end
```

```
{
  "id": 1,
  "name": "Toyotomi Hideyori",
  "clan": {
    "id": 1,
    "name": "Toyotomi"
  }
}
```

Active Model Serializer

config/application.rb

```ruby
ActiveModelSerializers.config.adapter = :json_api
```

```json
{
  "data": {
    "id": "1",
    "type": "shoguns",
    "attributes": { "name": "Toyotomi Hideyori" },
    "relationships": {
      "clan": { "data": { "id": "1", "type": "clans" } }
    }
  }
}
```

Active Model Serializer

```
class ShogunController
  def show
   render json: shogun, include: 'clan'
  end

  private

  def shogun
    @shogun ||= Shogun.find(params[:id])
  end
end
```

Active Model Serializer

```
{
  "data": {
    "id": "1",
    "type": "shoguns",
    "attributes": { "name": "Toyotomi Hideyori" },
    "relationships": {
      "clan": { "data": { "id":"1", "type":"clans" } }
    }
  },
  "included": [
    {
      "id": "1",
      "type": "clans",
      "attributes": { "name": "Toyotomi" }
    }
  ]
}
```

Alternatywy

```ruby
gem 'fast_jsonapi'
gem 'jsonapi-resources'
gem 'panko'
```

Thanks!