

Ruby On Rails course

API



What is API?

Application program interface

Why do we need API?

Application Programming Interface

Application Programming Interface (API) is a set of protocols, routines, and tools for building an application that uses the services of another application (or operating system, database, or network).

APIs allow different software applications to communicate with each other, even if they are written in different programming languages or run on different platforms.

APIs are used in a wide variety of contexts, from web services and mobile applications to operating systems and databases.

APIs provide a standardized way for applications to interact, making it easier to develop and maintain software.

APIs are essential for building modern software systems, as they enable different components to work together seamlessly.

Internal usage

- your mobile application
- integrations with your other products

External usage

- allow others to integrate your product



Basic example

Get all subreddits

Request

tidder.com/api/subreddits

Response

```
{
  "response": [
    {
      "id": 1,
      "name": "...",
      "description": "...",
    },
    {
      "id": 2,
      "name": "...",
      "description": "...",
    }
  ],
  "count": 2
}
```



Formats

Response data formats

Formats

- JSON
- XML
- YAML
- RSS
- ...

Versioning

Prevent breaking changes

Versioning

- `tidder.com/api/v1/subreddits`
- `tidder.com/api/v2/subreddits`



Authentication

Authentication

- private API
- open API

Token authentication

- "893970fb-3190-49cf-bb59-d3edc65bf898"
- add token to user
- generate token on create
- regenerate token periodically

Token authentication

- find user by token

```
@current_user = User.find_by(token: params[:token])
```



Building API

Ways to do it

- build everything from scratch
- rails-api gem
- grape gem
- rocket_pants gem



RocketPants

API building tools on top of ActionController. Also, an awesome name.

Why RocketPants?

- conventions & configuration
- versioning
- dealing with errors
- caching
- testing

Example app

```
class Recipe < ActiveRecord::Base
  validates :name, presence: true
  validates :description, presence: true

  has_many :ingredients
end

class Ingredient < ActiveRecord::Base
  validates :name, presence: true
  validates :description, presence: true

  belongs_to :recipe
end
```



Example app

- GET /api/v1/recipes
- GET /api/v1/recipes/:id



app/controllers/api/v1/recipes_controller.rb

```
module Api
  module V1
    class RecipesController < RocketPants::Base
      version 1

      def index
        expose Recipe.paginate(page: params[:page])
      end

      def show
        expose Recipe.find(params[:id])
      end
    end
  end
end
```



Routes

```
namespace :api do
  api version: 1, module: 'v1', allow_prefix: 'v' do
    resources :recipes, only: [:index, :show]
  end
end
```

Prefix	Verb	URI Pattern	Controller#Action
api_recipes	GET	/api/:version/recipes	api/v1/recipes#index
api_recipe	GET	/api/:version/recipes/:id	api/v1/recipes#show



Error handling

- built-in errors

```
error! :bad_request # 400 Bad Request
```

```
error! :unauthenticated # 401 Unauthorized
```

```
error! :throttled # 503 Unavailable
```



Error handling

- mapping ActiveRecord

```
def update
  expose Recipe.find(params[:id]).update(recipe_params)
end
```

```
{
  "error": "invalid_resource",
  "error_description": "The current resource was deemed invalid.",
  "messages": {
    "name": ["can't be blank"],
    "description": ["can't be blank"]
  }
}
```

Serializing

Ways to do it

- `to_json`
- jbuilder gem
- active model serializers



Active Model Serializers

Why Active Model Serializers?

- separate file for definition
- ruby like syntax
- overriding
- caching



app/serializers/api/v1/recipe_serializer.rb

```
module Api
  module V1
    class RecipeSerializer < ActiveRecord::Serializer
      attributes :name, :description

      has_many :ingredients
    end
  end
end
```



app/controllers/api/v1/recipes_controller.rb

```
def show  
  expose @recipe, serializer: RecipeSerializer  
end
```



Response

```
{
  "response": [
    {
      "id": 1,
      "name": "...",
      "description": "...",
      "ingredients": [
        {
          "id": 1,
          "name": "...",
          "description": "...",
        },
        {
          "id": 1,
          "name": "...",
          "description": "...",
        }
      ]
    }
  ],
  "count": 1
}
```



app/serializers/api/v1/recipe_serializer.rb

```
module Api
  module V1
    class RecipeSerializer < ActiveRecord::Serializer
      attributes :name, :description

      has_many :ingredients, serializer: IngredientSerializer
    end
  end
end
```



Overriding

```
class RecipeSerializer < ActiveRecord::Serializer
  attributes :name, :description, :short_description

  has_many :ingredients, serializer: IngredientSerializer

  def name
    object.name.downcase
  end

  def short_description
    object.description.truncate(25)
  end

  def ingredients
    object.ingredients.approved
  end
end
```



Testing

Testing API

- manual
- semi-automatic
- automatic

POSTTMMAN

- chrome extension
- <https://www.getpostman.com/>



Documenting

What my API provides?

Documentation Example

- <http://www.reddit.com/dev/api>
- <https://productive.io/docs/index.html>

Ways to do it

- apipie-rails
- grape-swagger
- apiblueprint
- Rocket_Docs



Rocket Docs Example

```
51 |   |  
52 |   # PUT  
53 |   #   Doc  
54 |   #     Praesent maximus, leo a maximus fringilla, urna felis sollicitudin  
55 |   #     nunc, eu pulvinar est urna eu justo. Phasellus quis hendrerit nibh.  
56 |   #     Praesent id nunc ac augue ultricies rutrum at vel quam.  
57 |   #   Params  
58 |   #     id: integer  
59 |   # PATCH  
60 |   #   Doc  
61 |   #     Phasellus ac diam sit amet elit cursus tincidunt. Donec vel  
62 |   #     tincidunt orci. Maecenas in feugiat tortor. Lorem ipsum dolor sit  
63 |   #     amet, consectetur adipiscing elit.  
64 |   #   Params  
65 |   #     id: integer  
66 |   #     age: integer  
67 |   #     name: string  
68 |   |  
69 |   def update  
70 |     puts 'Why would you want to change things?'  
71 |   end
```



Rocket Docs Example

Documentation All versions v1 v2 v3

Version v1

People

index

create

new

edit

show

update

PATCH PUT

URL

Description

Phasellus ac diam sit amet elit cursus tincidunt. Donec vel tincidunt orci. Maecenas in feugiat tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Params

Name	Type
id	integer
age	integer
name	string

destroy

localhost:3000/api-doc#version_v1_controller_people_action_update



SDK

API wrapper

SDK Example

```
@subreddits = Tidder::Subreddit.all  
# GET /api/v1/subreddits
```

```
@subreddit = Tidder::Subreddit.find(1)  
# GET /api/v1/subreddits/1
```

```
@subreddit = Tidder::Subreddit.create(subreddit_params)  
# POST /api/v1/subreddits
```



SDK Example

- <https://github.com/reddit/reddit/wiki/API-Wrappers>
- <https://github.com/sferik/twitter>
- <https://github.com/arsduo/koala>

HW note

Additional task for replacing author/creator_name with user_id