

Grape入門

やまもとじゅん

Version 0.1, 2018/12/16

目次

1. 概要.....	2
1.1. オフィシャルな情報源.....	2
2. Grape インストール.....	3
2.1. Bundler.....	3
3. Grape API の実装.....	4
3.1. 最初の一步.....	4
3.2. Grap::Entity.....	8
4. 参考資料.....	9

GrapeはRailsでRESTful APIを書く時に便利なマイクロフレームワーク。APIを簡単に作ることができるライブラリの一つです。

1. 概要

Grape は Grape::API のサブクラス化で生成する [Rackアプリケーション](#) です

- RubyのREST-likeなAPIフレームワーク
- [Rack](#) 上で動作するように設計
- RESTful APIを簡単に開発できるシンプルなDSLを提供

1.1. オフィシャルな情報源

- [Grape \(Git\)](#)
- [Web Site](#)

CAUTION	rails5にはapiモードがある
----------------	-------------------

2. Grape インストール

2.1. Bundler

[Bundler](#) を利用しているなら [Gemfile](#) に記述

```
gem 'grape'
```

[gem](#) を Bundler でインストール

```
bundle install
```

```
gem install grape
```

3. Grape API の実装

実装を学ぶ

3.1. 最初の一步

とにかく動くものを作ってみる

3.1.1. API の実装

Grape::API を継承した Module を作成します

[/app/api/test/api.rb](#)

```
module Test
  class API < Grape::API

    format :json

    resource :api do

      desc "Return Hello"
      get :hello do
        {message: "Hello Grape"}
      end

      desc "Hello Name"
      params do
        requires :name, type: String, desc: "twice str"
      end
      get :twice do
        {message: 'Hello ' + params[:name]}
      end
    end
  end
end
```

3.1.2. Rails側の設定

[/app/api](#) 配下のファイルの読み込み

Railsでapiファイルを自動で読み込む

[/config/application.rb](#)

```
config.paths.add File.join('app', 'api'), glob: File.join('**', '*.rb')
config.autoload_paths += Dir[Rails.root.join('app', 'api', '**')]
```

APIのルーティングを追加

[/config/routes.rb](#)

```
mount Test::API => '/test/'
```

3.1.3. 動作確認

起動

```
ruby server
```

curl 等で確認

```
curl http://localhost:9292/test/api/hello
```

```
curl http://localhost:9292/test/api/name?name=June
```

3.1.4. オフィシャルなサンプル

オフィシャルドキュメントのサンプルを実装する。

Basic Usage [ruby-grape/grape@GitHub](#)

Rackアプリケーション

Grape::API を継承した Grape API。Twitter を模したサンプルのよう
[/app/api/twitter/api.rb](#)

```
module Twitter
  class API < Grape::API
    version 'v1', using: :header, vendor: 'twitter'
    format :json
    prefix :api

    helpers do
      def current_user
        @current_user ||= User.authorize!(env)
      end

      def authenticate!
        error!('401 Unauthorized', 401) unless current_user
      end
    end

    resource :statuses do
      desc 'Return a public timeline.'
      get :public_timeline do
        # とりあえずデータないので...
        #Status.limit(20)
        [{message:"hoge"}]
      end

      desc 'Return a personal timeline.'
      get :home_timeline do
        authenticate!
        current_user.statuses.limit(20)
      end

      desc 'Return a status.'
      params do
        requires :id, type: Integer, desc: 'Status id.'
      end
      route_param :id do
        get do
          Status.find(params[:id])
        end
      end
    end
  end
end
```



```

desc 'Create a status.'
params do
  requires :status, type: String, desc: 'Your status.'
end
post do
  authenticate!
  Status.create!({
    user: current_user,
    text: params[:status]
  })
end

desc 'Update a status.'
params do
  requires :id, type: String, desc: 'Status ID.'
  requires :status, type: String, desc: 'Your status.'
end
put ':id' do
  authenticate!
  current_user.statuses.find(params[:id]).update({
    user: current_user,
    text: params[:status]
  })
end

desc 'Delete a status.'
params do
  requires :id, type: String, desc: 'Status ID.'
end
delete ':id' do
  authenticate!
  current_user.statuses.find(params[:id]).destroy
end
end
end
end

```

Rails側の設定

/app/api 配下のファイルの読み込み

Railsでapiファイルを自動で読み込む

[/config/application.rb](#)

([最初の一步](#) と同じ設定のままでOK)

APIのルーティングを追加

[/config/routes.rb](#)

```
mount Twitter::API => '/twitter/'
```

動作確認

```
curl http://localhost:3000/twitter/api/statuses/public_timeline
```

3.2. Grape::Entity

この gem は、Grape などのAPIフレームワークに エンティティサポート を追加します。Grape's Entity は、オブジェクトモデルの最上位に位置するAPIフォーカスファードです。

3.2.1. インストール

Gemfile

```
gem 'grape-entity'
```

```
bundle install
```

3.2.2. エンティティの定義

エンティティは Grape::Entity から継承し、単純なDSLを定義します。

NOTE

Entity = 実態
類義語として「インスタンス」、「オブジェクト」など。

4. 参考資料

- [Rails&grapeで簡単WebAPI @Qiita](#)
- [Rails\(Grape\)でAPIを作成する備忘録1 @Qiita](#)
- [railsでGrapeを使ってAPI作成、プラス例外処理 @Qiita](#)
- [Grapeを使ってWeb APIを作成する @DevelopersIO](#)