

API

- 1) `mysql -u root -p`
- 2) `show databases;`
- 3) `create database myart_development;`
- 4) `create database myart_test;`
- 5) `grant all on myart_development. * to myart@localhost identified by 'myart123';`
- 6) `grant all on myart_test. * to myart@localhost identified by 'myart123';`
- 7) `show databases;`
- 8) `select host, user from mysql.user;`
- 9) `FLUSH privileges;`
- 10) `sudo nano /etc/apache2/apache2.conf`

 `ServerName ubuntu`
 `Include /etc/phpmyadmin/apache.conf` (add this in the bottom)
- 11) `sudo service apache2 restart` (for phpmyadmin)
- 12) `rails new myart --api`
- 13) `cd myart`
- 14) `add mysql2 gem`
- 15) `bundle`
- 16) `change this`

 `development:`
 `<<: *default`
 `adapter: mysql2`
 `database: myart_development`
 `username: myart`
 `password: myart123`

 `test:`
 `<<: *default`
 `adapter: mysql2`
 `database: myart_test`
 `username: myart`
 `password: myart123`
- 17) `rails g model Article title:string body:text`

- 18) rails [db:migrate](#)
- 19)

```
class Article < ApplicationRecord
  validates :title, presence: true
  validates :body, presence: true
end
```
- 20) search for 'rails faker'

add this in gemfile 'faker' in development
- 21) now go to seeds.rb file


```
5.times do
  Article.create({
    title: Faker::Book.title,
    body: Faker::Lorem.sentence
  })
end
```
- 22) rails [db:seed](#)
- 23) create new folder in controllers/api
create new folder in controllers/api/v1
- 24) create new file in controllers/api/v1/articles_controller.rb


```
module Api
  module v1
    class ArticlesController < ApplicationController
      def index
        articles = Article.order('created_at DESC')
        render json: {status: 'SUCCESS', message:'Loaded articles',
data:articles},status:ok
      end
    end
  end
end
```
- 25) routes.rb


```
Rails.application.routes.draw do
  namespace 'api' do
    namespace 'v1' do
      resources :articles
    end
  end
end
```
- 26) then goto postman and check it for index option

http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ + ... No Environment

GET http://localhost:3000/api/v1/articles Params Send Save

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK Time: 58 r

Pretty Raw Preview JSON

```

1 {
2   "status": "SUCCESS",
3   "message": "Loaded articles",
4   "data": [
5     {
6       "id": 4,
7       "title": "Moab Is My Washpot",
8       "body": "Cum aut est et et nihil adipisci alias.",
9       "created_at": "2017-11-15T02:46:05.000Z",
10      "updated_at": "2017-11-15T02:46:05.000Z"
11    },
12    {
13      "id": 5,
14      "title": "Updated title",
15      "body": "Body also updated",
16      "created_at": "2017-11-15T02:46:05.000Z",
17      "updated_at": "2017-11-15T03:54:26.000Z"
18    },
19    {
20      "id": 1,
21      "title": "That Good Night",
22      "body": "Et qui molestiae in.",
23      "created_at": "2017-11-15T02:46:04.000Z",
24      "updated_at": "2017-11-15T02:46:04.000Z"
25    }
26  ]
27 }

```

27) create new file in controllers/api/v1/articles_controller.rb

def show

```

    article = Article.find(params[:id])
    render json: {status: 'SUCCESS', message:'Loaded article',

```

```

    data:article},status: :ok

```

```

    end

```

http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ http://localhost:3000/ + ... No Environment

GET http://localhost:3000/api/v1/articles/3 Params Send

Authorization Headers Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (6) Test Results Status: 200 OK

Pretty Raw Preview JSON

```

1 {
2   "status": "SUCCESS",
3   "message": "Loaded article",
4   "data": {
5     "id": 3,
6     "title": "No Highway",
7     "body": "Corporis occaecati optio officiis saepe non voluptas sequi voluptate.",
8     "created_at": "2017-11-15T02:46:04.000Z",
9     "updated_at": "2017-11-15T02:46:04.000Z"
10  }
11 }

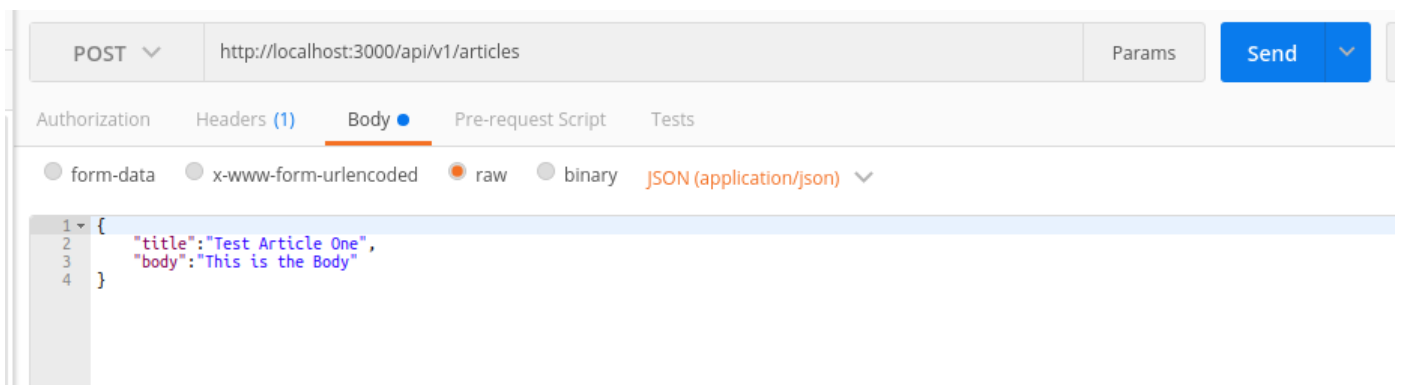
```

28) then create

```
def create
  article = Article.new(article_params)

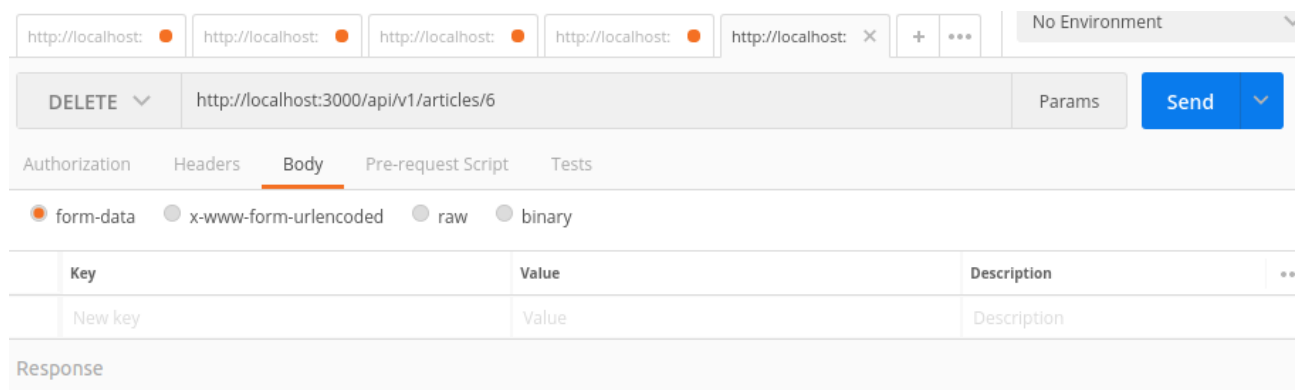
  if article.save
    render json: {status: 'SUCCESS', message:'Saved article',
data:article},status: :ok
  else
    render json: {status: 'ERROR', message:'Article not saved',
data:article.errors},status: :unprocessable_entity
  end
end

def article_params
  params.permit(:title, :body)
end
```



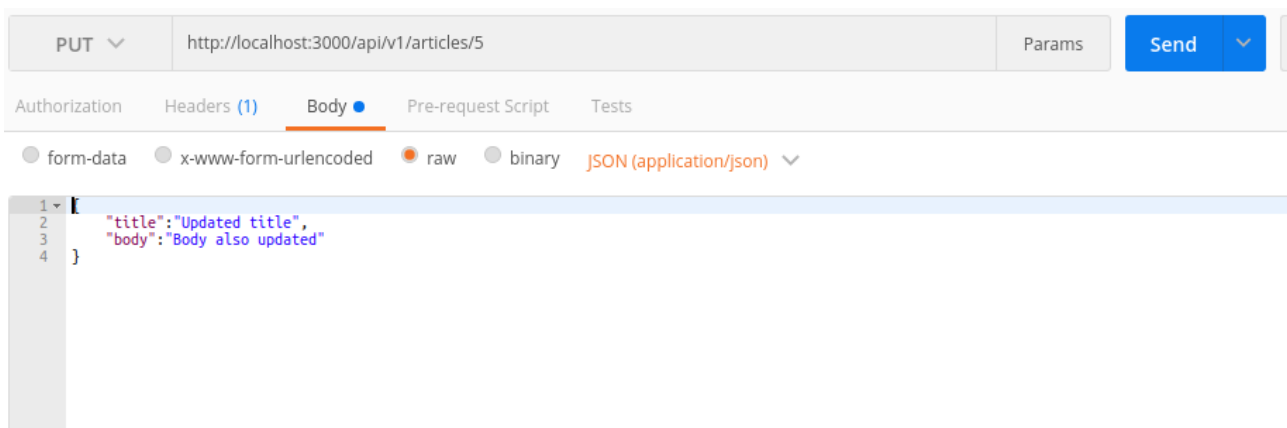
29) destroy

```
def destroy
  article = Article.find(params[:id])
  article.destroy
  render json: {status: 'SUCCESS', message:'Deleted article',
data:article},status: :ok
end
```



30) update

```
def update
  article = Article.find(params[:id])
  if article.update_attributes(article_params)
    render json: {status: 'SUCCESS', message:'Updated article',
data:article},status: :ok
  else
    render json: {status: 'ERROR', message:'Article not Updated',
data:article.errors},status: :unprocessable_entity
  end
end
```



```
module Api
  module V1
    class ArticlesController < ApplicationController
      def index
        articles = Article.order('created_at DESC')
        render json: {status: 'SUCCESS', message:'Loaded articles',
data:articles},status: :ok
      end

      def show
        article = Article.find(params[:id])
        render json: {status: 'SUCCESS', message:'Loaded article',
data:article},status: :ok
      end

      def create
        article = Article.new(article_params)

        if article.save
          render json: {status: 'SUCCESS', message:'Saved article',
data:article},status: :ok
        end
      end
    end
  end
end
```

```

        else
            render json: {status: 'ERROR', message:'Article not saved',
data:article.errors},status: :unprocessable_entity
        end
    end

    def destroy
        article = Article.find(params[:id])
        article.destroy
        render json: {status: 'SUCCESS', message:'Deleted article',
data:article},status: :ok
    end

    def update
        article = Article.find(params[:id])
        if article.update_attributes(article_params)
            render json: {status: 'SUCCESS', message:'Updated article',
data:article},status: :ok
        else
            render json: {status: 'ERROR', message:'Article not Updated',
data:article.errors},status: :unprocessable_entity
        end
    end

    def article_params
        params.permit(:title, :body)
    end
end
end
end
end

```