Trilha de Testes & Qualidade de Software

Workshop:
Testando APIs em Ruby

womakerscode.org

umbler

Daiane Azevedo de Fraga

Desenvolvedora de software na **Umbler**

Automação de testes + DevOps \o/

WOMAKERS CODE

umbler

O que é uma API

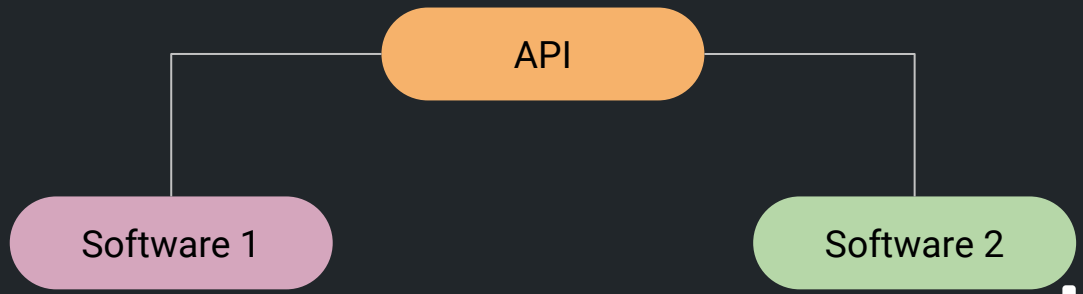
WOMAKERS
CODE

umbler

Uma API é...

Application Programming Interface

Conjunto de padrões e rotinas que compreendem uma forma de acessar um software com base na Web.



WOMAKERS CODE

umbler

Uma API...

Permite que outras pessoas/empresas desenvolvam softwares relacionados a outro software.

Potencializa automações no contexto de DevOps.

É uma forma de modularizar um sistema e facilitar a gerência de dependência.

umbler

O que é REST

REST é...

***Re**presentational **S**tate **T**ransfer*

Abstração da arquitetura da Web. Em outras palavras, é um modelo ou estilo de arquitetura, consistindo em um conjunto de padrões que, se utilizados, facilitam a comunicação entre sistemas na web.

WOMAKERS CODE

umbler

REST compreende...

Cliente e servidor se comunicam, mas suas implementações são independentes.

Requests são feitas a partir do cliente para o servidor.

Conexão através do protocolo HTTP.

HTTP Requests ⟶ Verbos HTTP (GET, POST, PUT, DELETE), URLs, métodos, paths...

WOMAKERS CODE

umbler

Verbos...

GET ⟶ Consultar/obter/ler dados

POST ⟶ Criar/registrar

PUT ⟶ Alterar/atualizar

DELETE ⟶ Excluir

Testando uma API

umbler

Testar uma API REST exige...

Conhecer os requisitos que precisam ser atendidos pela API (conhecer o sistema).

Permissão para acessar a API através da web.

Saber fazer requests HTTP e analisar as suas respostas.

umbler

Mãos à obra?!

umbler

Acesse a API

http://node-api-example-com.umbler.net/

umbler

O que essa API faz?

Uma API simples que armazena uma lista de palavras. Você pode consultar, criar, alterar ou remover palavras.

umbler

Rotas da API

`GET - /` : retorna uma string indicando que a API está online.

`GET - /total` : retorna o número total de palavras armazenadas.

`GET - /word/:id` : retorna uma palavra específica.

`GET - /words` : retorna todas as palavras armazenadas.

`POST - /words` : armazena uma nova palavra.

`PUT - /words/:id` : altera uma palavra específica.

`DELETE - /words/:id` : exclui uma palavra.

Requisições

GET - se houver parâmetros, eles são passados na própria URL.

Exemplo:
http://teste.com?param1=valor&param2=valor

Requisições

POST - se houver parâmetros, eles são passados no corpo da mensagem.

Exemplo: http://teste.com/metodo
Payload:

```
{
   “param1”: valor
}
```

Requisições

PUT/DELETE - identificação do elemento pode ser passada pela URL.

Exemplo: http://teste.com/testes/4

Respostas

GET - **/total** deve retornar um número:

5

Respostas

GET - **/word/:id** deve retornar um hashmap com ID e palavra:

```
{
   "1": "teste"
}
```

OU

```
{
   "error": "Element does not exist."
}
```

WOMAKERS CODE

umbler

Respostas

GET - **/words** deve retornar um hashmap com todas as palavras e seus IDs:

```
{
   "1": "teste",
   "2": "teste_2"
}
```

Respostas

POST - **/words** deve retornar um "OK" se conseguiu salvar:

```
"OK"
```

Respostas

PUT - **/words/:id** deve retornar um "OK" se conseguiu alterar:

```
"OK"
```

OU

```
{
   "error": "Could not update. Element does
not exist."
}
```

WOMAKERS CODE

umbler

Respostas

DELETE - **/words/:id** deve retornar um "OK" se conseguiu excluir:

```
"OK"
```

OU

```
{
   "error": "Could not delete. Element does
not exist."
}
```

WOMAKERS CODE

umbler

Gerencie as dependências

Gemfile

```
source 'https://rubygems.org'

gem 'airborne'
```

bundler.io

umbler

Crie um arquivo

```
api_test.rb
```

umbler

Use um framework

```
require 'airborne'
```

WOMAKERS CODE

umbler

Configuração inicial

```
Airborne.configure do |config|
  config.base_url =
'http://node-api-example-com.umbler.net'
end
```

GET e verificação de status

```
describe "quando acessamos a URL base via
GET" do
  it "deveria retornar status 200" do
    get "/"
    expect_status(200)
  end
```

umbler

GET e verificação de conteúdo

```
  it "deveria responder 'Funcionando'" do
    get "/"
    expect(body).to match(/Funcionando!/)
  end
end
```

POST e verificação de criação

```
describe "quando criamos uma nova palavra
via POST em /words"  do
  it "deveria retornar status 200" do
    post "/words", { :word => 'John Doe' }
    expect_status(200)
  end
```

POST e verificação de criação

```
  it "deveria retornar a palavra armazenada
via GET" do
    get "/words"
    expect_json({"1": 'John Doe'})
  end
end
```

node_test_airborne.rb

```ruby
require 'airborne'

Airborne.configure do |config|
  config.base_url = 'http://node-api-example-com.umbler.net'
end

describe "quando acessamos a URL base via GET"  do
  it "deveria retornar status 200" do
    get "/"
    expect_status(200)
  end

  it "deveria responder 'Funcionando'" do
    get "/"
    expect(body).to match(/Funcionando!/)
  end
end

describe "quando criamos uma nova palavra via POST em /words"  do
  it "deveria retornar status 200" do
    post "/words", { :word => 'John Doe' }
    expect_status(200)
  end

  it "deveria retornar a palavra armazenada via GET" do
    get "/words"
    expect_json({"1": 'John Doe'})
  end
end
```

WOMAKERS CODE

umbler

Executando

```
rspec api_test.rb
```

umbler

Se passar...

```
daiane@DaianeNote:~/api_test_example$ rspec node_test_airborne.rb
....

Finished in 0.56852 seconds (files took 0.38779 seconds to load)
4 examples, 0 failures
```

umbler

Se falhar...

```
daiane@DaianeNote:~/api_test_example$ rspec node_test_airborne.rb
...F

Failures:

  1) quando criamos uma nova palavra via POST em /words deveria retornar a palavra armazenada via GET
     Failure/Error: expect_json({"1": 'John Doe', "2": 'Teste'})

       expected: "Teste"
            got: "John Doe"

       (compared using ==)
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:95:in `block in expect_json_impl'
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:87:in `each'
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:87:in `expect_json_impl'
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:19:in `block in expect_json'
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:139:in `call_with_path'
     # /home/daiane/.rvm/gems/ruby-2.3.3/gems/airborne-0.2.13/lib/airborne/request_expectations.rb:18:in `expect_json'
     # ./node_test_airborne.rb:27:in `block (2 levels) in <top (required)>'

Finished in 0.54297 seconds (files took 0.39411 seconds to load)
4 examples, 1 failure

Failed examples:

rspec ./node_test_airborne.rb:25 # quando criamos uma nova palavra via POST em /words deveria retornar a palavra armazenada via GET
```

WOMAKERS CODE

umbler

Quer saber mais?

Sempre procure a documentação de seu framework:

`https://github.com/brooklynDev/airborne`

umbler

Dúvidas?!

WOMAKERS
CODE

umbler

Obrigada!

daiane@umbler.com
@daiane_a_fraga

WOMAKERS CODE

umbler