

# MODELO DE DOCUMENTAÇÃO DE REST APIs

Este documento explicita com exemplos, como utilizar os recursos disponíveis no REST API de Reserva e Comparação de Hotéis. Assim como, as formas de se realizar uma requisição e suas possíveis respostas.

## 1. Consultar Hotéis

### Requisição

Requisição para listar todos os hotéis do sistema, podendo opcionalmente receber filtros personalizados via path, de forma que se o cliente não definir nenhum parâmetro de consulta (nenhum filtro), os parâmetros receberão os valores padrão.

- Possíveis parâmetros de consulta
  - cidade ⇒ Filtrar hotéis pela cidade escolhida. Padrão: Nulo
  - estrelas\_min ⇒ Avaliações mínimas de hotéis de 0 a 5. Padrão: 0
  - estrelas\_max ⇒ Avaliações máximas de hotéis de 0 a 5. Padrão: 5
  - diaria\_min ⇒ Valor mínimo da diária do hotel de R\$ 0 a R\$ 10.000,00. Padrão: 0
  - diaria\_max ⇒ Valor máximo da diária do hotel de R\$ 0 a R\$ 10.000,00. Padrão: 10000
  - limit ⇒ Quantidade máxima de elementos exibidos por página. Padrão: 50
  - offset ⇒ Quantidade de elementos pular (geralmente múltiplo de limit). Padrão: 0

Method	URL
GET	/hoteis?estrelas_min=4.5&limit=10&offset=0&diaria_max=600

## Resposta

Como resposta, obtém-se uma lista de hotéis que se enquadram nos filtros da requisição acima:

Status	Response Body
200 OK	<pre>{   "hoteis": [     {       "hotel_id": "golf",       "nome": "Golf Hotel",       "estrelas": 4.5,       "diaria": 500.9,       "cidade": "Santa Catarina"     },     {       "hotel_id": "lima",       "nome": "Lima Hotel",       "estrelas": 4.6,       "diaria": 423.75,       "cidade": "Santos"     },     {       "hotel_id": "tango",       "nome": "Tango Hotel",       "estrelas": 4.5,       "diaria": 530.9,       "cidade": "Rio de Janeiro"     },     {       "hotel_id": "xray",       "nome": "X-Ray Hotel",       "estrelas": 4.7,       "diaria": 398.95,       "cidade": "Rio de Janeiro"     }   ] }</pre>

## Requisição

Requisição para visualizar os dados de um hotel específico. Faz-se um GET de `/hoteis/{hotel_id}`

Method	URL
GET	/hoteis/bravo

## Resposta

Como resposta, obtém-se um JSON com os dados do hotel requisitado.

Status	Response Body
200 OK	<pre>{   "hotel_id": "bravo",   "nome": "Bravo Hotel",   "estrelas": 4.9,   "diaria": 720.2,   "cidade": "Rio de Janeiro" }</pre>

## Requisição

Requisição exemplo de quando o usuário pesquisar por um hotel que não existe.

Method	URL
GET	/hoteis/hotel_id_que_nao_existe

## Resposta

Como resposta, obtém-se uma mensagem de erro, dizendo que o hotel não foi encontrado.

Status	Response Body
404 Not Found	<pre>{   "message": "Hotel not found." }</pre>

## 2. Cadastro de Usuário

### Requisição

Exemplo de Requisição cadastrar um novo usuário.

Method	URL
POST	/cadastro

Header	
Content-Type	application/json

Request Body
<pre>{   "login": "ana",   "senha": "asdf" }</pre>

### Resposta

Como resposta, obtém-se uma mensagem de sucesso informado que usuário foi criado, e status code 201 Created (Criado).

Status	Response Body
201 Created	<pre>{   "message": "User created successfully!" }</pre>

## Requisição

Exemplo de Requisição para tentar cadastrar outro usuário com login “ana”.

Method	URL
POST	/cadastro

Header	
Content-Type	application/json

Request Body
<pre>{  "login": "ana",  "senha": "abcd"}</pre>

## Resposta

Como resposta, obtém-se uma mensagem de erro, informando que um usuário chamado “ana” já existe.

Status	Response Body
400 Bad Request	<pre>{  "message": "The login 'ana' already exists."}</pre>

### 3. Login de Usuário

#### Requisição

Exemplo de Requisição logar com um usuário.

Method	URL
POST	/login

Header	
Content-Type	application/json

Request Body
<pre>{   "login": "ana",   "senha": "asdf" }</pre>

#### Resposta

Como resposta, obtém-se uma mensagem o token de acesso que será necessário para fazer as requisições que só podem ser feitas com login.

Status	Response Body
200 OK	<pre>{   "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiE1NTY3MjE5OTUsIm5iZiI6MTU1NjcyMTI5NSwianRpIjo1ZDdiNmZiODktMTQ4ZS00ZWQ0LTg0YzAtODdmZWFLMTIwIiwiaXNjaXNTU2NzIyMTk1LCJpZGVudG0eSI6MiwiaWZnJlciOmZhbHNLCj0eXBIjo1YWNjZXNzIn0.D9SYe-OYi681bp0eeTBu0_yEmIsObe-EBLWBidxLaRg" }</pre>

## Requisição

Exemplo de Requisição para tentar fazer login com um usuário que não existe.

Method	URL
POST	/login

Header	
Content-Type	application/json

Request Body
<pre>{  "login": "usuario_que_nao_existe",  "senha": "asdf"}</pre>

## Resposta

Como resposta, obtém-se uma mensagem de erro 401 não autorizado, informando que usuário ou senha estão incorretos.

Status	Response Body
401 unauthorized	<pre>{  "message": "The username or password is incorrect."}</pre>

## 4. Criar Hotel

### Requisição

Exemplo de Requisição para criar um novo hotel: `/hoteis/{hotel_id}`

Method	URL
POST	/hoteis/teste

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Request Body
<pre>{   "nome": "Teste Hotel",   "estrelas": 4.4,   "diaria": 520.2,   "cidade": "Rio de Janeiro" }</pre>

### Resposta

Exemplos de Respostas: primeiro, a mensagem de sucesso ao criar pela primeira vez. Ao tentar criar novamente um hotel com id “teste”, o sistema emite uma mensagem de erro, informando que o id “teste” já existe.

Status	Response Body
201 Created	<pre>{   "nome": "Teste Hotel",   "estrelas": 4.4,   "diaria": 520.2,   "cidade": "Rio de Janeiro" }</pre>



400 Bad Request	<pre>{   "message": "Hotel id 'teste' already exists." }</pre>
-----------------	--

## 5. Atualizar Hotel

### Requisição

Exemplo de Requisição para atualizar um novo hotel: `/hoteis/{hotel_id}`

Method	URL
PUT	/hoteis/teste

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Request Body
<pre>{   "nome": "Teste Hotel Alterado",   "estrelas": 4.9,   "diaria": 720.2,   "cidade": "Rio de Janeiro" }</pre>

## Resposta

Exemplos de Respostas: primeiro, a mensagem de sucesso ao atualizar dados de hotel, e status code 200 OK. O PUT também consegue criar um novo hotel, como demonstrado no exemplo do hotel com id “novo”, emitindo assim uma mensagem de 201 Created a primeira vez, e 200 OK a partir da segunda requisição.

Status	Response Body
200 OK	<pre>{  "nome": "Teste Hotel Alterado",  "estrelas": 4.9,  "diaria": 720.2,  "cidade": "Rio de Janeiro"}</pre>
201 Created	<pre>{  "hotel_id": "novo",  "nome": "Novo Hotel",  "estrelas": 4.9,  "diaria": 720.2,  "cidade": "Rio de Janeiro"}</pre>
401 Unauthorized	<pre>{  "msg": "Token has expired"}</pre>

## 6. Deletar Hotel

### Requisição

Exemplo de Requisição para deletar um hotel: `/hoteis/{hotel_id}`

Method	URL
DELETE	/hoteis/teste

Header	
Authorization	Bearer {token_de_acesso}

## Resposta

Exemplos de Respostas, primeiro, a mensagem de sucesso ao deletar um hotel existente. Depois, ao tentar deletar o mesmo hotel, obtém-se o erro 404 not found, informando que o hotel não existe. No terceiro exemplo, o cliente esqueceu de enviar o token de autorização.

Status	Response Body
200 OK	<pre>{   "message": "Hotel deleted." }</pre>
404 Not Found	<pre>{   "message": "Hotel not found." }</pre>
401 Unauthorized	<pre>{   "msg": "Missing Authorization Header" }</pre>

## 7. Logout de Usuário

### Requisição

Exemplo de Requisição para fazer logout de usuário. Envia-se o token de acesso, e esse token é invalidado.

Method	URL
POST	/logout

Header	
Authorization	Bearer {token_de_acesso}

## Resposta

Exemplo de Resposta: mensagem de sucesso informando que o usuário foi deslogado. Ao tentar usar esse token de acesso em qualquer requisição, ele não funcionará mais, a não ser que o usuário faça o login.

Status	Response Body
200 OK	<pre>{"message": "Logged out successfully!"}</pre>

## 8. Consultar dados de Usuário

### Requisição

Exemplo de Requisição para ler dados de usuário: `/usuarios/{user_id}`

Method	URL
GET	<code>/usuarios/2</code>

Header	
Content-Type	application/json

## Resposta

Como resposta, obtém-se os dados do usuário com id “2” exceto a senha.

Status	Response Body
200 OK	<pre>{   "user_id": 2,   "login": "ana" }</pre>

## 9. Deletar Usuário

### Requisição

Exemplo de Requisição para deletar um usuário: `/usuarios/{user_id}`

Method	URL
DELETE	/usuarios/2

Header	
Authorization	Bearer {token_de_acesso}

### Resposta

Exemplos de Respostas, primeiro, a mensagem de sucesso ao deletar um usuário existente. Depois, ao tentar deletar o mesmo usuário, obtém-se o erro 404 not found, informando que o usuário não existe ou não foi encontrado. No terceiro exemplo, o cliente enviou um token de autorização expirado.

Status	Response Body
200 OK	<pre>{   "message": "User deleted." }</pre>
404 Not Found	<pre>{   "message": "User not found." }</pre>
401 Unauthorized	<pre>{   "msg": "Token has expired" }</pre>