



## (4.1.1) Requisições HTTP

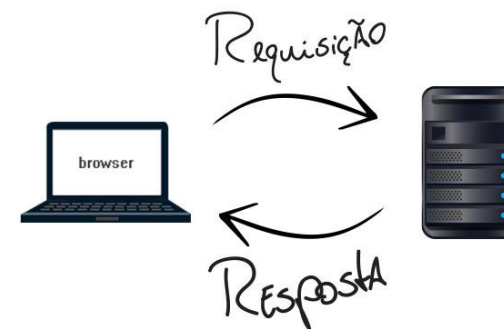
# O que são requisições HTTP

Requisições HTTP são trocas de dados entre *Cliente* e *Servidor*, utilizando, para isso, o protocolo HTTP.

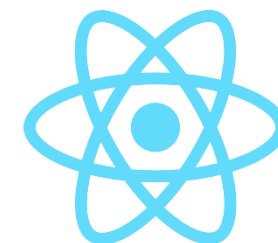
### Protocolo HTTP

HTTP é um protocolo de transferência que possibilita que as pessoas que inserem a URL do seu site na Web possam ver os conteúdos e dados que nele existem. A sigla vem do inglês Hypertext Transfer Protocol.

Esse sistema é a base da comunicação que existe em toda a Internet em que os sites e conteúdos que tragam hiperlinks possam ser encontrados mais facilmente pelo público por meio de um clique do mouse ou um toque na tela.



(C) Blog High5Devs, 2015.

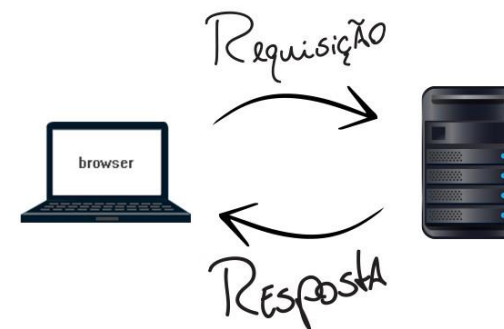


### O que são requisições HTTP

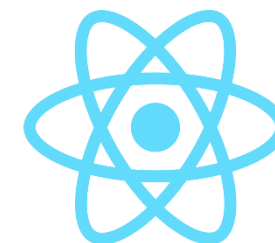
A parte que conhecemos melhor do protocolo HTTP é o endereço HTTP de um site.

Exemplo: <http://compracertaesegura.com.br>

O endereço acima é o que chamamos de URL (*Uniform Resource Locator – Localizador de Recurso Uniforme*). Ele representa um recurso específico na Web. Recursos são coisas que podemos interagir, como por exemplo: Imagens, páginas, arquivos e vídeos. No nosso exemplo, o recurso retornado é a página inicial do Site, normalmente um HTML.



(C) Blog High5Devs, 2015.



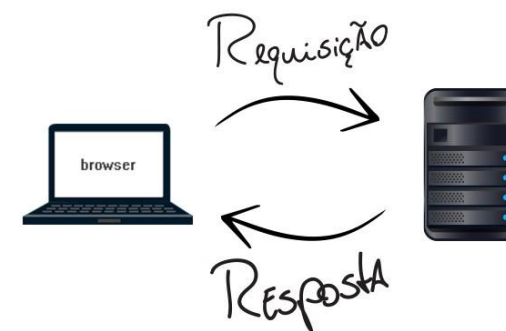
### O que são requisições HTTP

Imagine a seguinte URL, que representa um produto do nosso site.

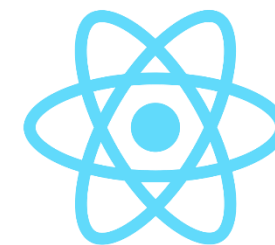
Exemplo: <http://compracertaesegura.com.br/produto/computador>

Podemos quebrá-la em 3 partes:

- **http**, a parte antes de "://" é o que chamamos de URL Scheme\_(esquema da URL). O esquema descreve como acessar um recurso em particular.
- **compracertaesegura.com.br** é o nome do **host**(servidor), que seria o nome do computador que armazena nosso recurso.
- **/produtos/computador** é o que chamamos de **URL Path** (caminho da URL). O servidor irá identificar qual é o recurso específico que deve devolver para este caminho quando a requisição chegar.



(C) Blog High5Devs, 2015.



# O que são requisições HTTP

### Status

Toda requisição recebe um código de resposta conhecido como status. Com o status é possível saber se uma operação foi realizada com ou não.

#### 200 OK

A requisição foi bem sucedida.

#### 201 Created

Indica que a requisição foi bem sucedida e que um novo recurso foi criado.

#### 401 Unauthorized

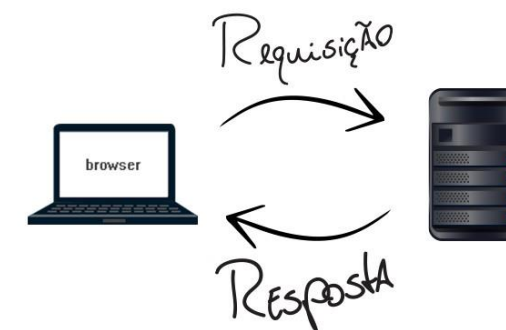
A URI especificada exige autenticação do cliente. O cliente pode tentar fazer novas requisições.

#### 404 Not Found

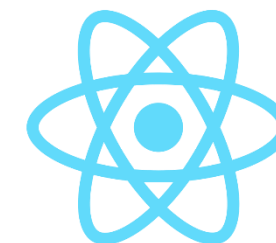
O servidor não encontrou nenhuma URI correspondente.

#### 500 Internal Server Error

O servidor não foi capaz de concluir a requisição devido a um erro inesperado.



(C) Blog High5Devs, 2015.



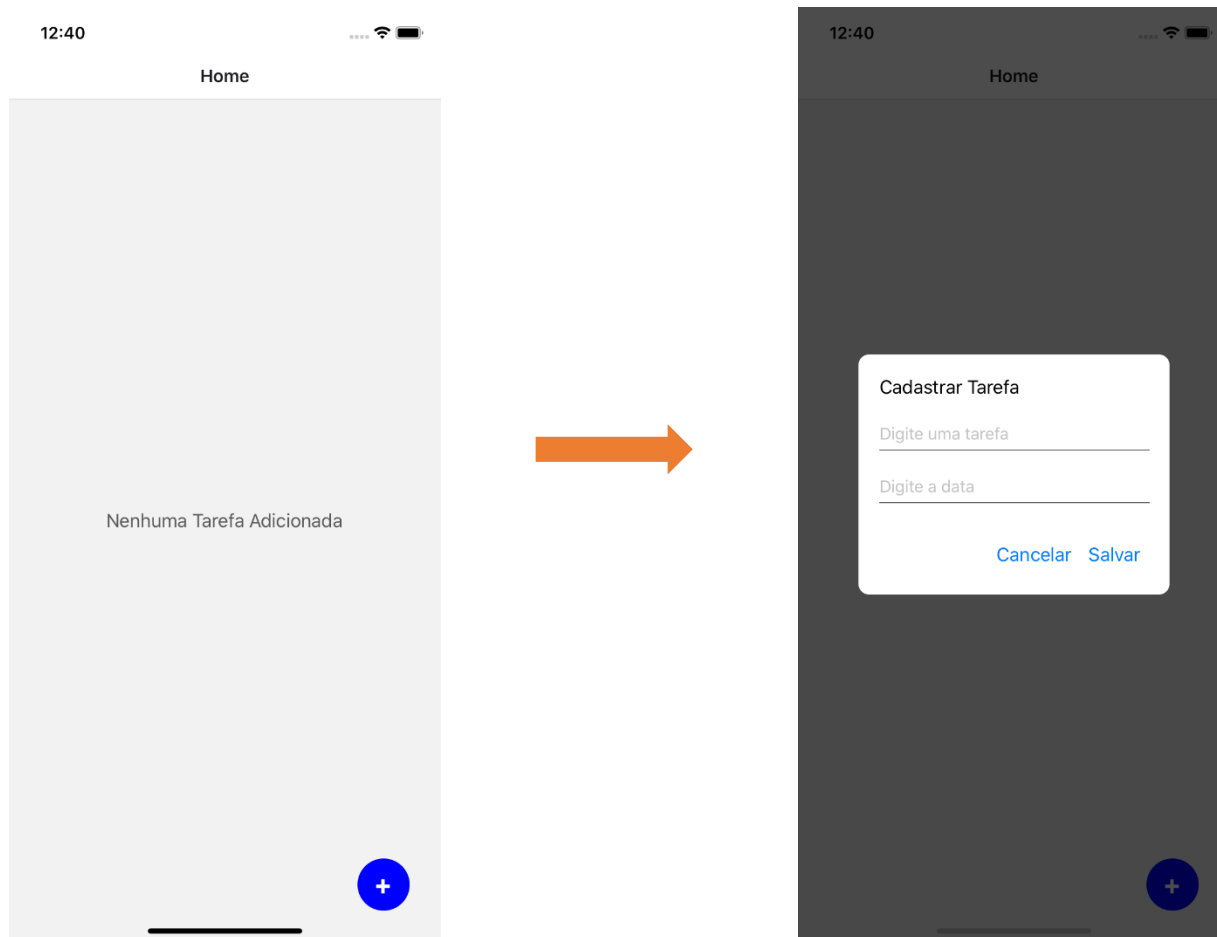


Obrigado!



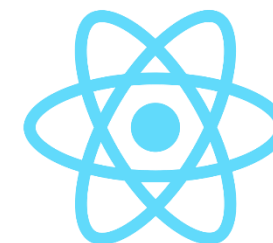
## (4.1.2) Preparando aplicação para Requisições HTTP

# Preparando aplicação para Requisições HTTP



### Componentes Utilizados:

- View,
- Text,
- Button,
- TouchableOpacity,
- Modal.





## Criando API Rest Fake com json-server

1 – Instalar Biblioteca: `npm install -g json-server`

2 – Criando arquivo .json: Crie um arquivo com o nome db.json

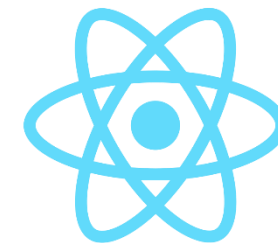
```
{
  "users": [{
    "id": 1,
    "name": "Andrew",
    "location": "Brazil"
  }, {
    "id": 2,
    "name": "Julio",
    "location": "Canadá"
  }]
}
```

3 – Rodando a API: `json-server db.json`

### 4. Testando os endpoints

utilizar o Postman para testar os seguintes endpoints

GET /users  
GET /users/1  
POST /users  
PATCH /users/1  
DELETE /users/1





Obrigado!

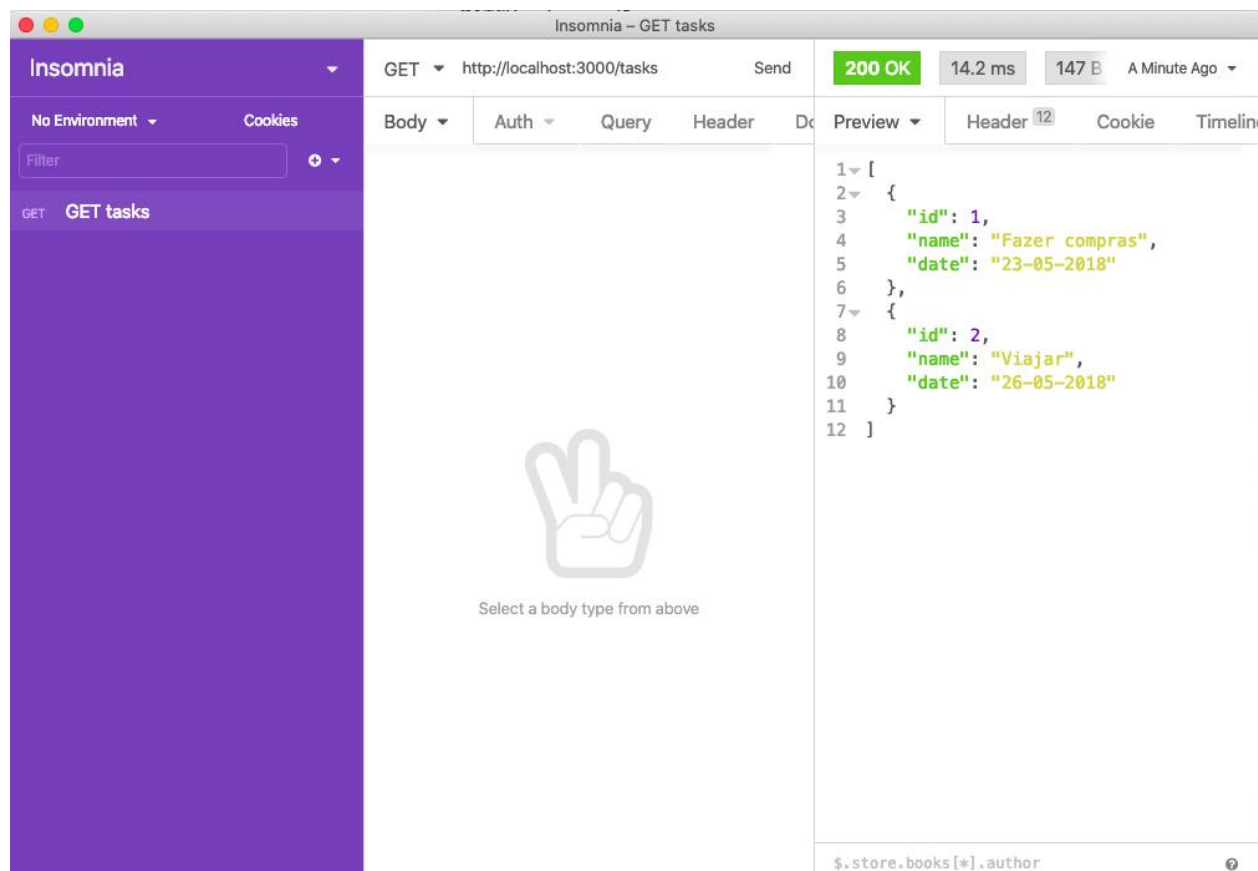


## (4.2.1) Método GET

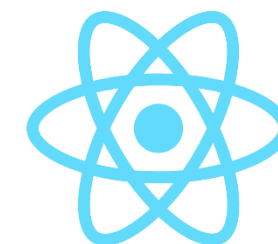
## Método GET

Essa é a requisição mais comum de todas. Através dessa requisição nós pedimos a representação de um recurso: que pode ser um arquivo html, xml, json, etc.

### *Todas as Tarefas*



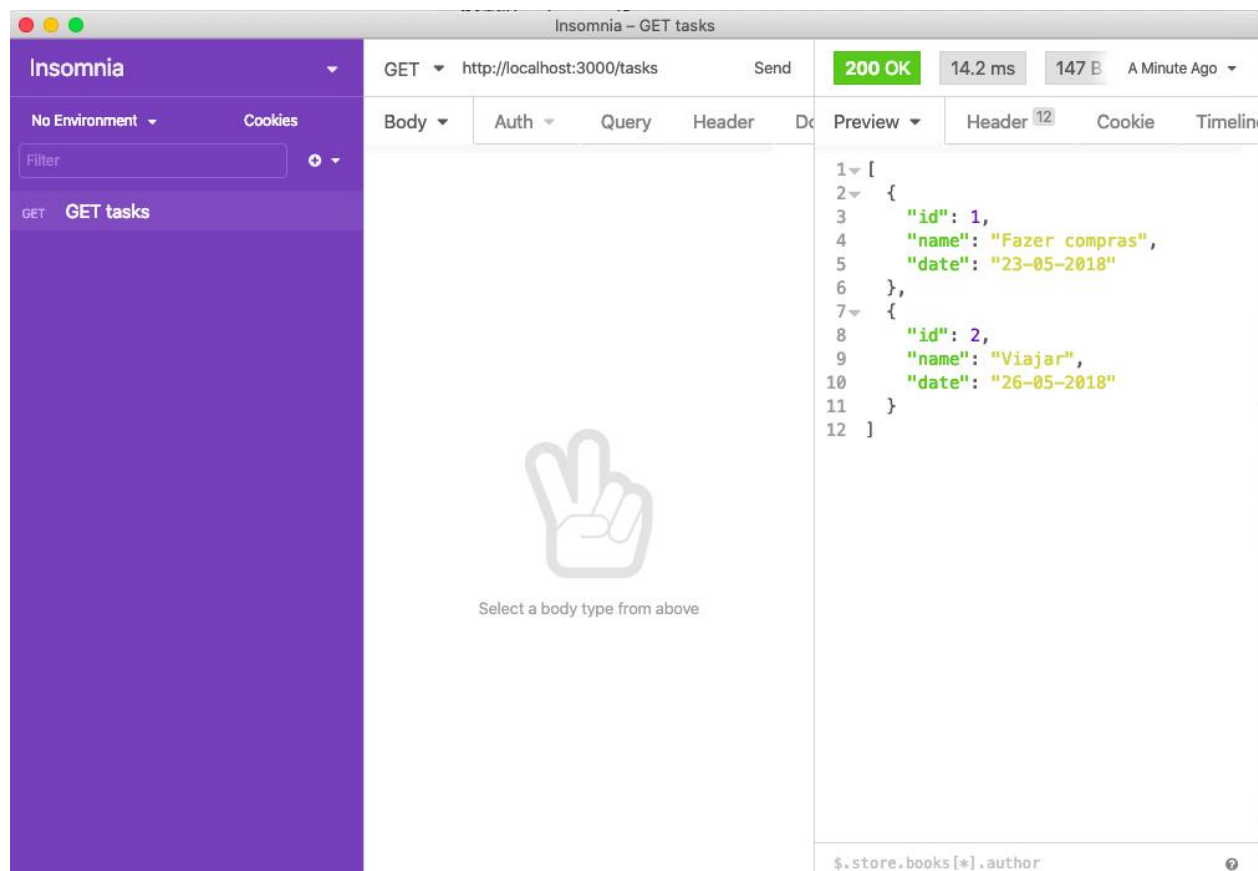
<http://localhost:3000/tasks>



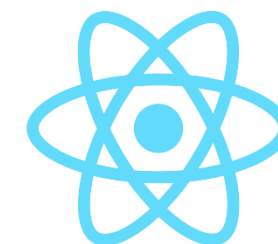
## Método GET

Essa é a requisição mais comum de todas. Através dessa requisição nós pedimos a representação de um recurso: que pode ser um arquivo html, xml, json, etc.

### *Tarefa específica*



<http://localhost:3000/tasks/1>



## Requisições no React Native

Existem várias maneiras de fazer requisições HTTPS em projetos React Native. Uma maneira simples e eficaz de implementar essa função é utilizando a biblioteca 'axios'

### axios

0.20.0 • Public • Published 9 days ago

Readme

Explore BETA

1 Dependency

40.749 Dependents

44 Versions

### axios

npm v0.20.0 cdnjs v0.20.0 build passing coverage 94% install size 383 kB downloads 47M/month

chat on github code helpers 142

Promise based HTTP client for the browser and node.js

### Table of Contents

- Features
- Browser Support
- Installing
- Example
- Axios API
- Request method aliases

### Install

```
> npm i axios
```

### Weekly Downloads

11.328.042

Version License

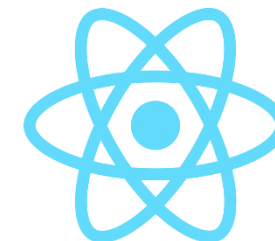
0.20.0 MIT

Unpacked Size Total Files

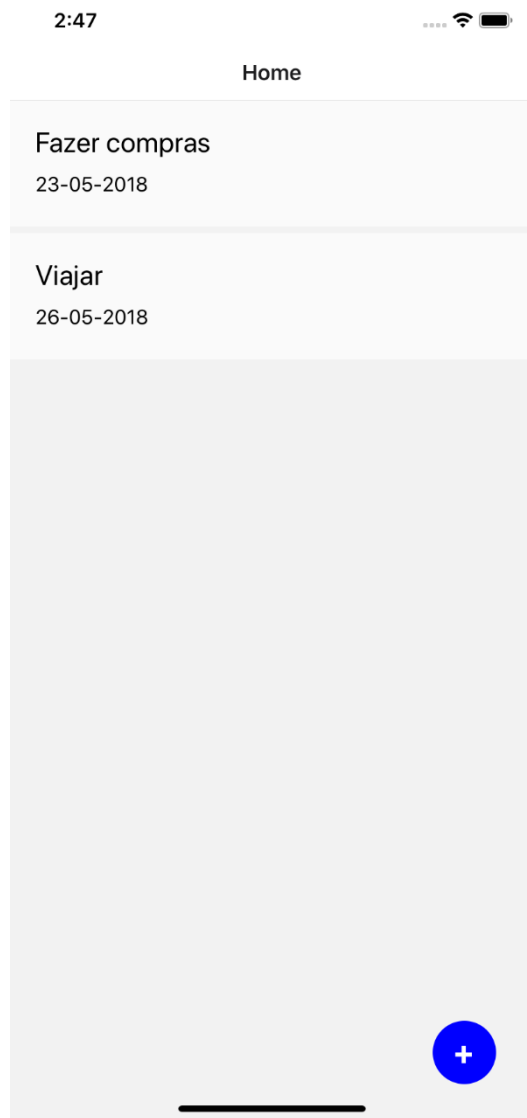
366 kB 41

Issues Pull Requests

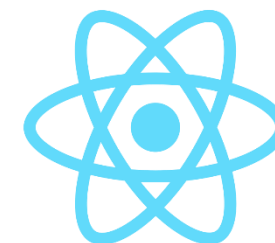
213 57



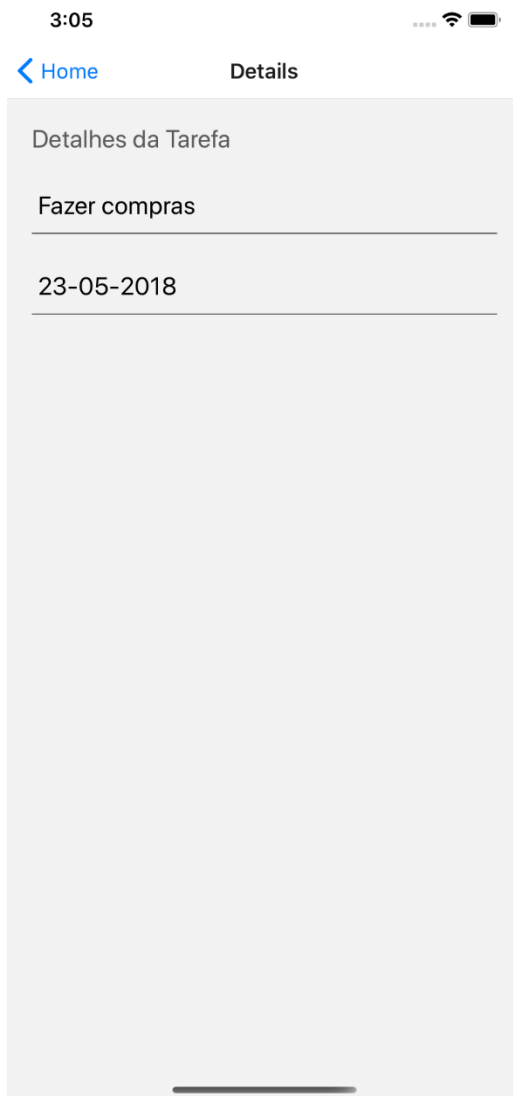
## Requisições no React Native – GET



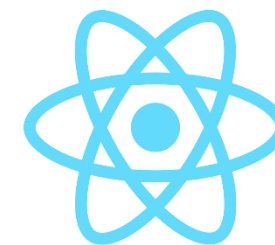
```
import axios from 'axios';
useEffect(() => {
  axios.get('http://localhost:3000/tasks').then((result) => {
    setTasks(result.data);
    console.log(JSON.stringify(result))
  })
}, [])
```



# Requisições no React Native – GET



```
import axios from 'axios';
const id = router.params.id;
useEffect(() => {
  axios.get('http://localhost:3000/tasks/' + id).then((result) =>
    > setTask(result.data));
}, [])
```







Obrigado!

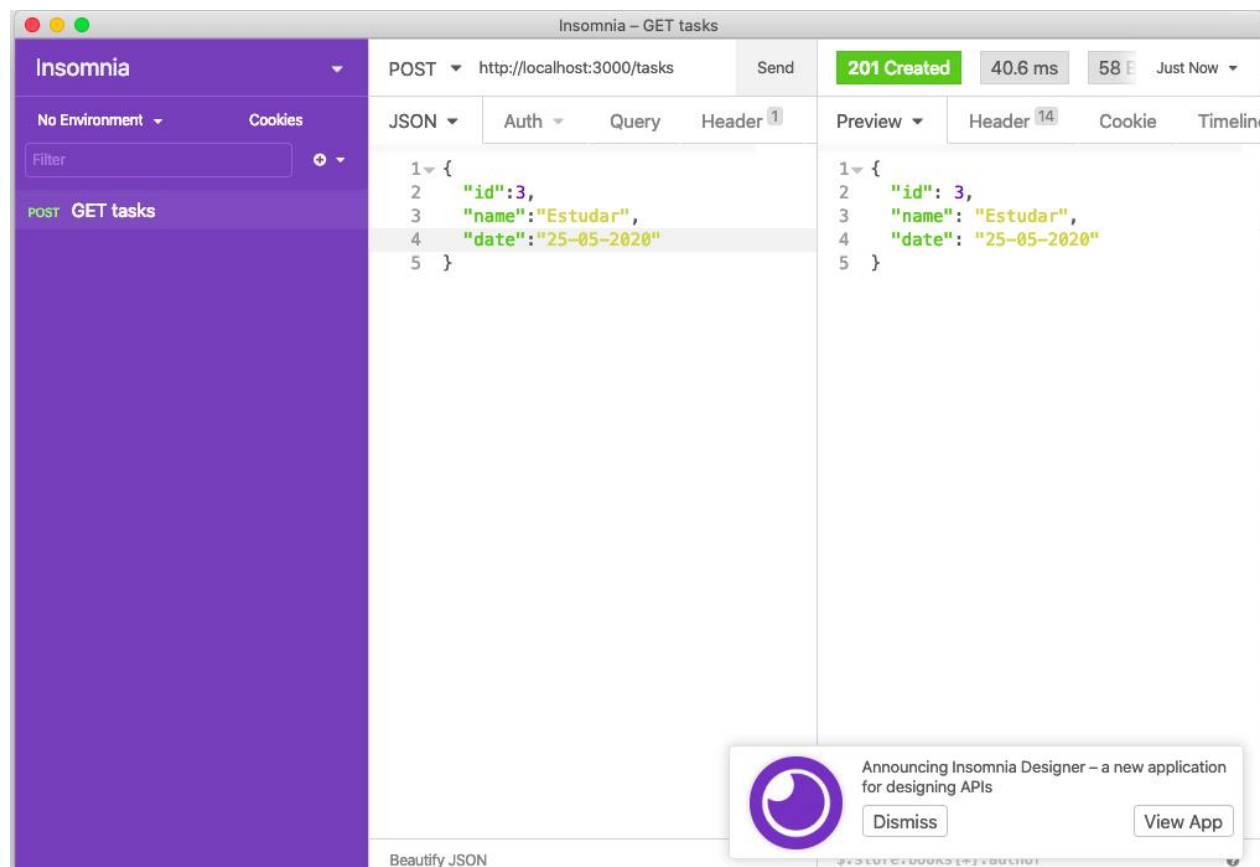


## (4.2.2) Método POST

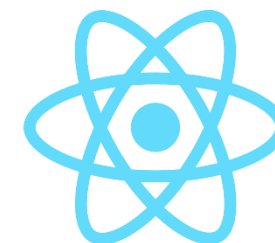
### Método POST

O método POST é utilizado quando queremos **criar** um recurso. Quando usamos POST, os dados vão no corpo da requisição e não na URI.

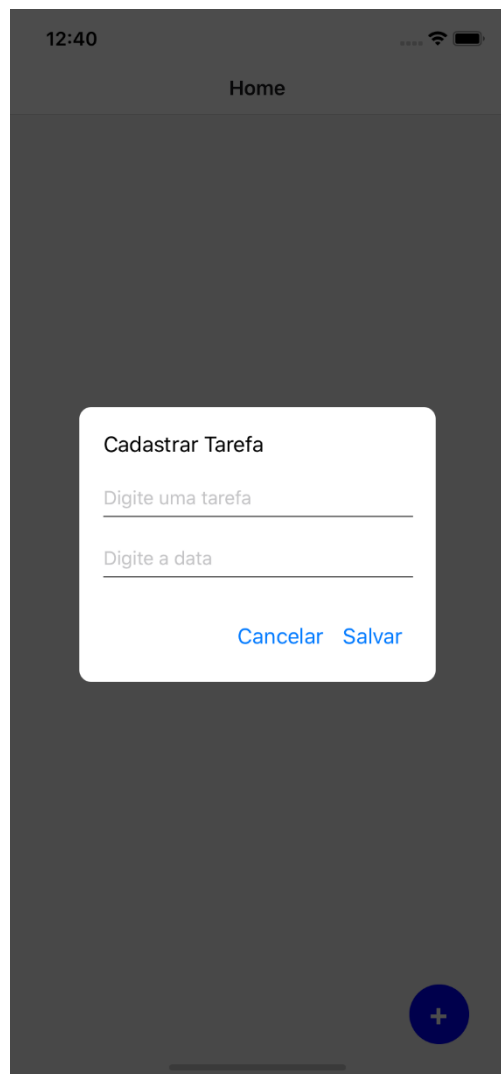
#### *Nova Tarefa*



`http://localhost:3000/tasks`



## Requisições no React Native – POST



12:40

Home

Cadastrar Tarefa

Digite uma tarefa

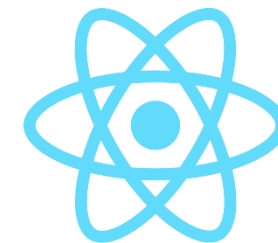
Digite a data

Cancelar Salvar

+

```
import axios from 'axios'

axios.post('http://localhost:3000/tasks', {
  id: 4,
  name: 'Ir ao shopping',
  date: '31-09-2020'
}).then(() => alert('Salvo com sucesso!'))
}
```





Obrigado!

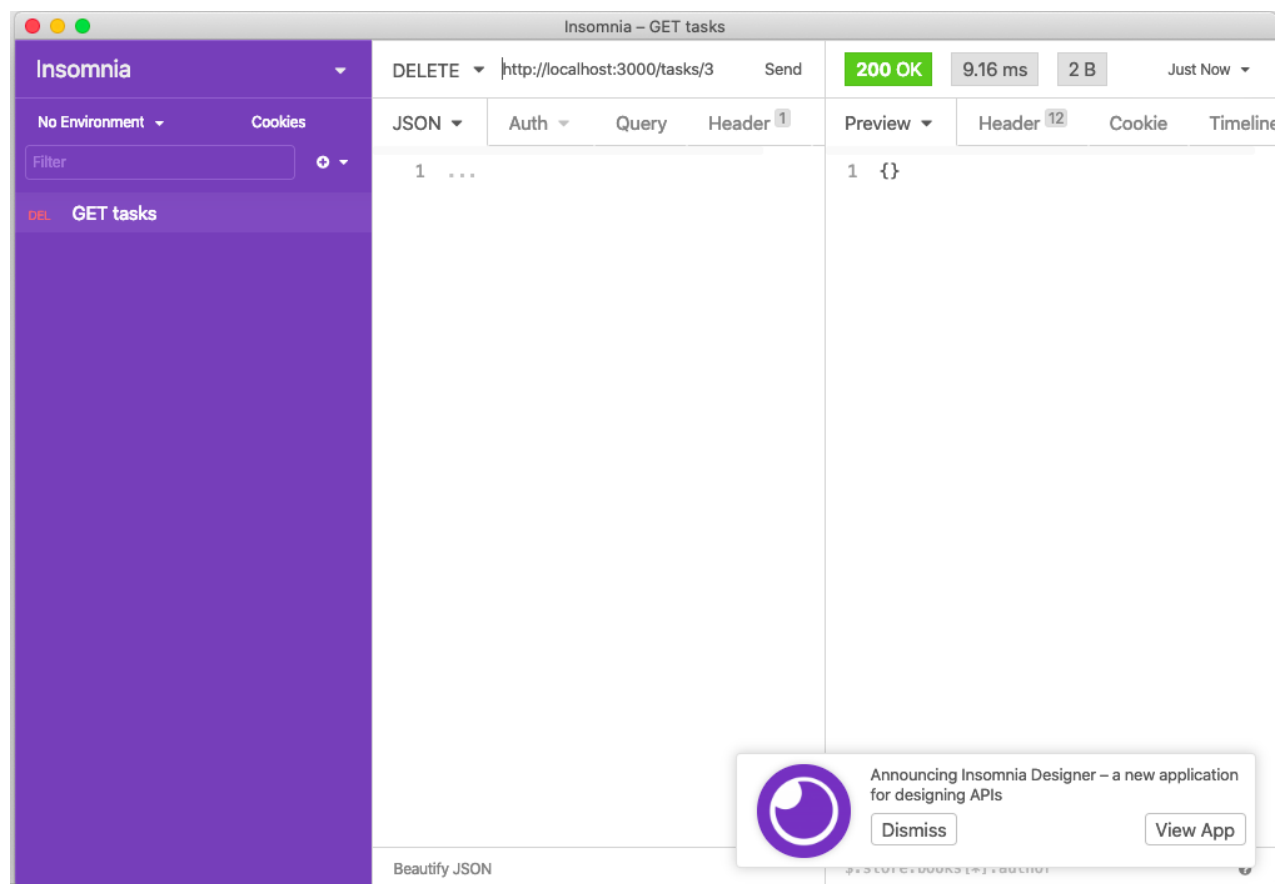


### (4.2.3) Método DELETE

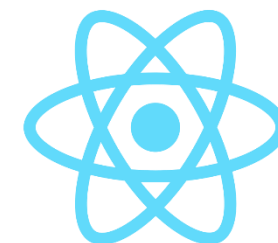
## Método DELETE

Exclui um recurso especificado.

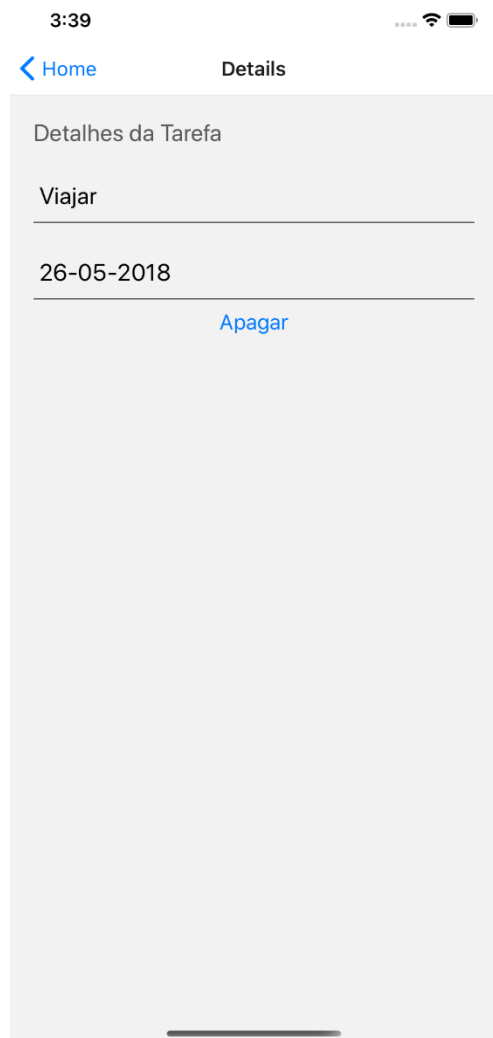
### *Excluir Tarefa*



<http://localhost:3000/tasks/3>

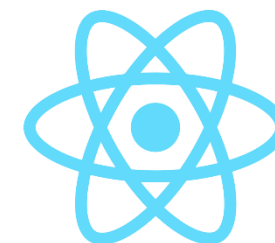


## Requisições no React Native – DELETE



```
import axios from 'axios';

const deleteTask = () => {
  axios.delete('http://localhost:3000/tasks/' + id).then(() =>
    alert('Deletada com sucesso!'))
}
```







Obrigado!

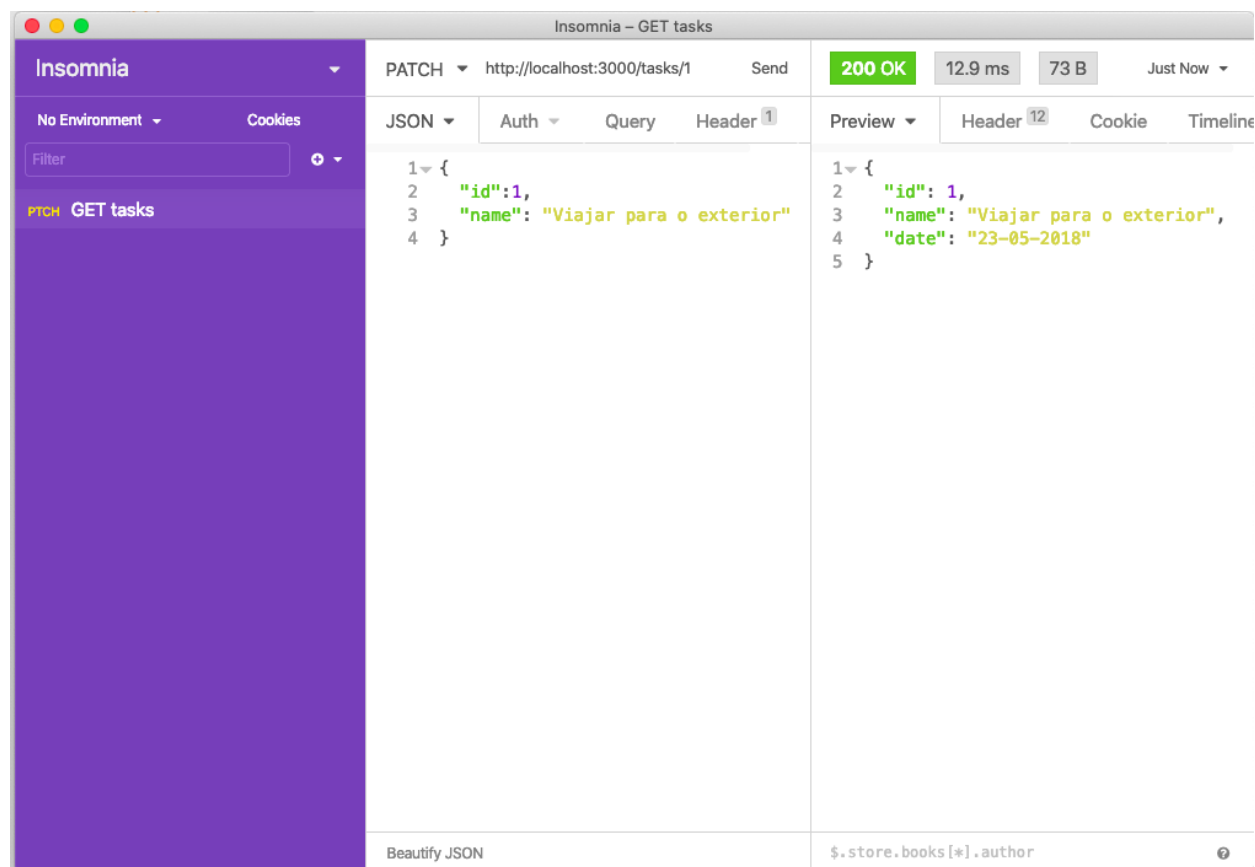


## (4.2.4) Método PATCH

## Método PATCH

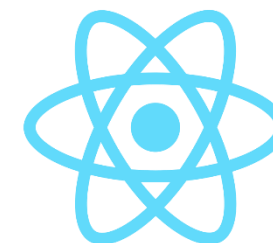
Serve para atualizar um recurso.

### Editar Tarefa

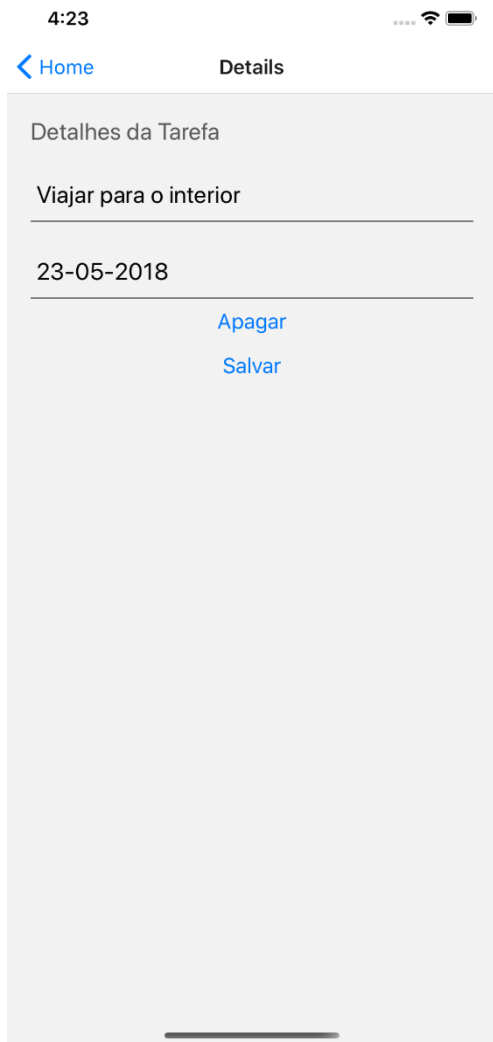


<http://localhost:3000/tasks/1>

*Podemos atualizar um registro utilizando o método **PUT**, porém, caso tentamos atualizar um registro que ainda não existe, um novo registro com os dados enviados no corpo da solicitação é criado.*

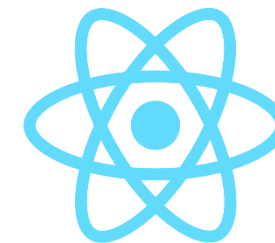


# Requisições no React Native – PATCH



```
import axios from 'axios';

const editTask = () => {
  axios.patch('http://localhost:3000/tasks/' + id,
    { name: 'Viajar para o interior' }).then(() =>
    alert('Atualizado com sucesso!'))
}
```



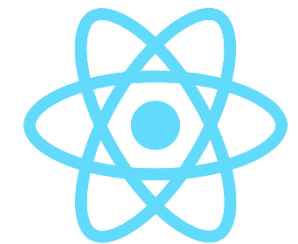
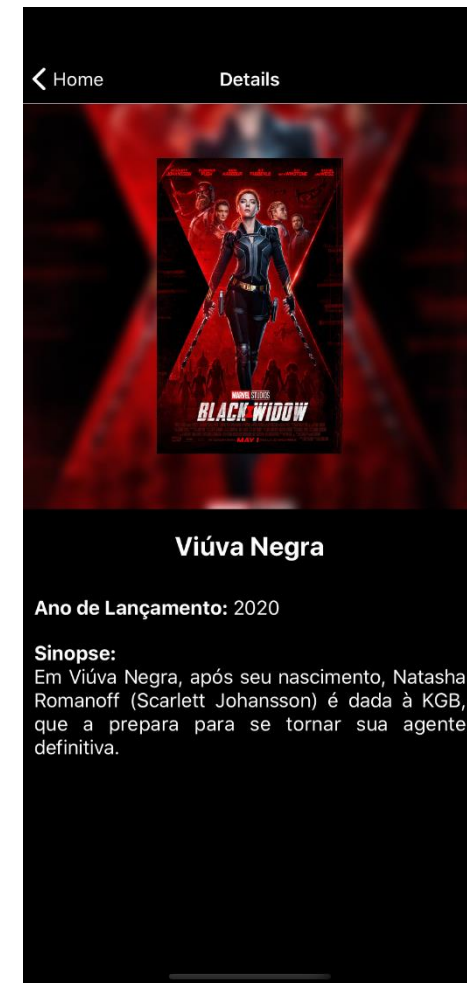
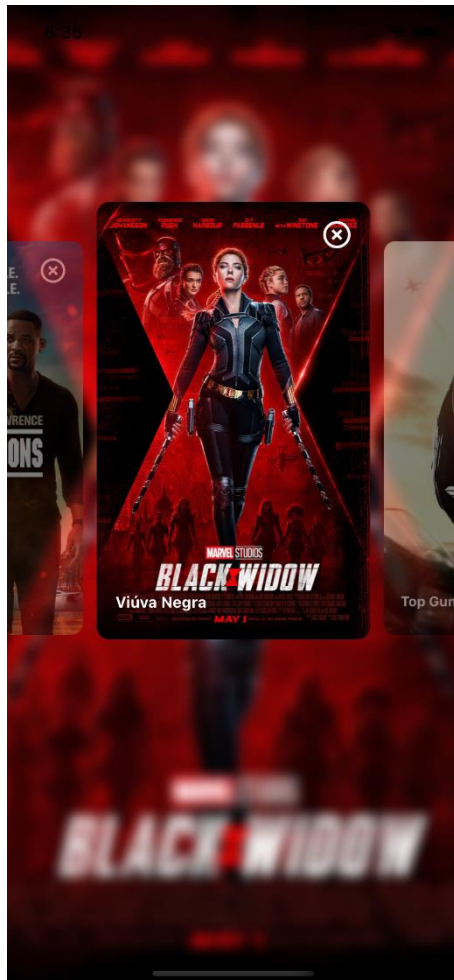


Obrigado!



## (4.3.1) Projeto Prático

## Projeto Prático – CRUD HTTP - Filmes Favoritos





Obrigado!