
ANDROID

RETROFIT
JWT
OKHTTP

INTRODUÇÃO

RETROFIT

É utilizado para facilitar o processo de realização de requisições;
Não precisamos fazer manualmente a serialização dos objetos;
Necessidade apenas de definir a nossa conexão.



[HTTP://SQUARE.GITHUB.IO/RETROFIT/](http://square.github.io/retrofit/)

COMPILE 'COM.SQUAREUP.RETROFIT2:RETROFIT:2.3.0'

- Open Source

INTRODUÇÃO

JWT

É um padrão aberto que possibilita a troca segura de informações dentre duas partes.

ESTRUTURA

JWT

A estrutura é dividida em 3 partes separadas por ponto:

Cabeçalho

Payload

Assinatura

CABEÇALHO

JWT

Consiste em 2 partes - Tipo de Algoritmo (256 ou RSA) e Tipo de Token:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD

JWT

Corpo da mensagem.

Composta por chave/valor.

```
{  
  "id": 1,  
  "nomeFantasia": "Nome Fantasia",  
  "razaoSocial": "Razão Social"  
}
```

ASSINATURA

JWT

Cabeçalho + Mensagem (Payload) codificados.

```
HMACSHA256(  
  base64UrlEncode(Cabeçalho) + "." +  
  base64UrlEncode(Payload),  
  secret)
```

Verificar se a mensagem não foi alterada no meio do caminho.

RESULTADO

JWT

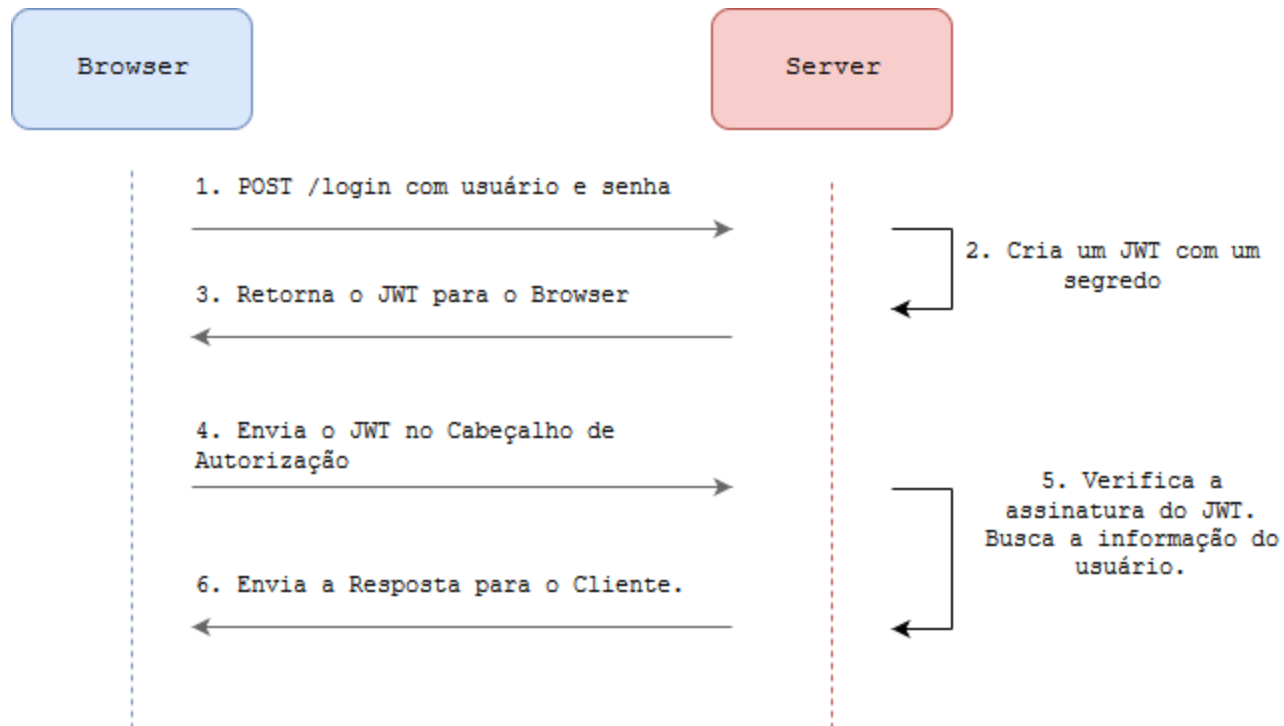
(Cabeçalho em base64).(Payload em base64).(Assinatura em base64)

GET /clientes HTTP/1.1

Host: <https://codexpauth.herokuapp.com/>

Authorization: Bearer (Cabeçalho em base64).(Payload em base64).(Assinatura em base64)

JWT

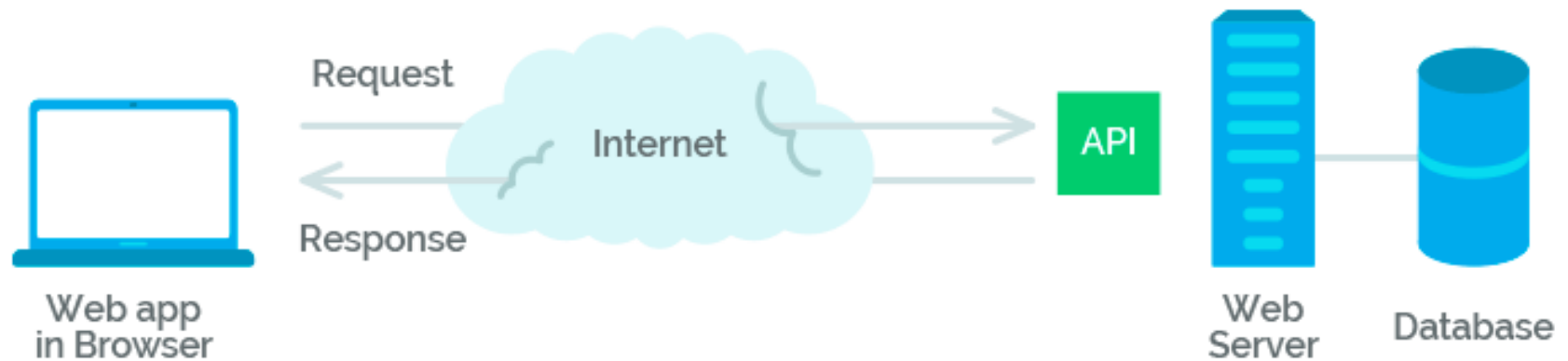


INTRODUÇÃO

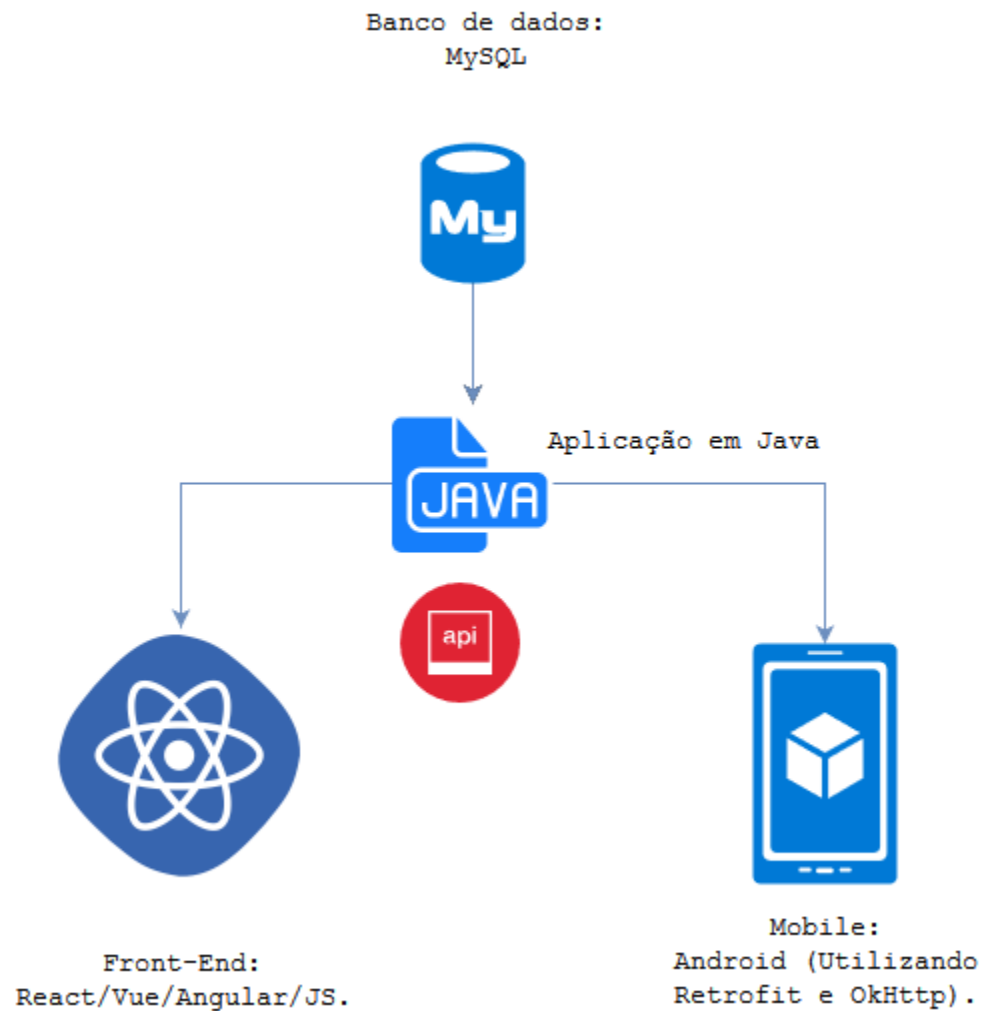
OKHTTP

Requisições para o servidor.
Interceptadores.

PADRÃO



ESTRUTURA D NOSSO PROJE



RETROFIT, JWT, OKHTTP

OBRIGADO!