

Autenticação

Autenticação

- Autenticação é o processo de verificação da identidade de um usuário;
- Em uma aplicação Angular, é possível implementar autenticação usando vários métodos:
 - Nome de usuário e senha;
 - OAuth;
 - JWT (JSON Web Token).

OAuth

- OAuth é um padrão aberto para autorização;
- Permite que os usuários concedam aos aplicativos acesso aos seus recursos sem compartilhar suas credenciais de login. No Angular, podemos usar o módulo HttpClient para fazer uma solicitação GET ao servidor com o código de autorização do usuário. O servidor retornará um token de autenticação se for bem-sucedido.

Nome de usuário e senha

- Usuários inserem nome de usuário e senha para se autenticarem;
- É Possível usar o módulo HttpClient para fazer uma solicitação POST ao servidor com as credenciais do usuário;
- O servidor verificará as credenciais e retornará um token de autenticação se for bem-sucedido.

Nome de usuário e senha

```
loginUser(email: string, password: string) {  
  return this.http.post<AuthResponseData>('https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=[digitesuachave]',  
  {  
    email: email,  
    password: password,  
    returnSecureToken: true  
  }).pipe(  
    tap(resData => {  
      const expiracaoData = new Date(new Date().getTime() + +resData.expiresIn * 1000);  
      const usuario = new Usuario(  
        resData.email,  
        resData.localId,  
        resData.idToken,  
        expiracaoData  
      );  
      this.usuario.next(usuario);  
      localStorage.setItem('userData', JSON.stringify(usuario));  
    }  
  ),  
  );  
}
```

OAuth

- Padrão aberto para autorização;
- Permite que os usuários concedam aos aplicativos acesso aos seus recursos sem compartilhar suas credenciais de login;
- É possível usar o módulo HttpClient para fazer uma solicitação GET ao servidor com o código de autorização do usuário;
- O servidor retornará um token de autenticação se for bem-sucedido.

```
loginComOAuth(codigo: string) {  
    return this.http.get('https://enderecoservidor.com.br/oauth', { codigo });  
}
```

JWT (JSON Web Token)

- é um meio compacto e seguro para URL de representar reivindicações a serem transferidas entre duas partes;
- Autenticar usuários sem armazenar suas credenciais no lado do cliente;
- É possível usar o módulo HttpClient para fazer uma solicitação GET ao servidor com o token JWT do usuário;
- O servidor então verificará o token e retornará um token de autenticação se for bem-sucedido.

```
loginWithJWT(token) {  
  | return this.http.get('https://enderecoservidor.com.br/jwt', { token });  
}
```

Autorização

- Depois que o usuário for autenticado, é preciso autorizá-lo para ações ou recursos específicos;
- É possível usar o serviço `HttpInterceptor` para interceptar solicitações HTTP e adicionar cabeçalhos de autorização;

Http Interceptor

```
@Injectable()
export class AutenticaInterceptor implements HttpInterceptor {

  constructor(private authService: AutenticaService) {}

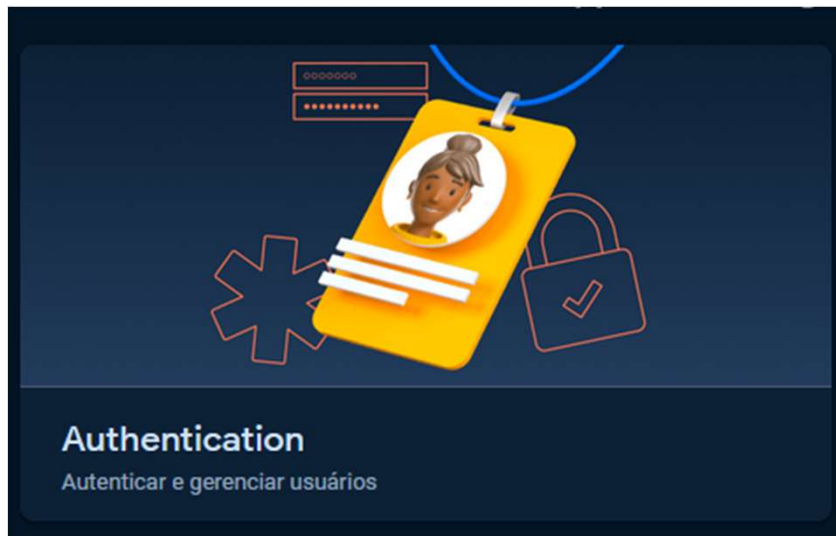
  intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEvent<unknown>> {
    return this.authService.usuario.pipe(
      take(1),
      exhaustMap(usuario => {
        if(!usuario){
          return next.handle(request);
        }
        const requestModificado = request.clone({
          params: new HttpParams().set('auth', usuario.token!)
        });
        return next.handle(requestModificado);
      })
    );
  }
}
```

Configurar app.module.ts

- Adicionar no providers em app.module.ts

```
providers: [{provide: HTTP_INTERCEPTORS, useClass: AutenticaInterceptor, multi: true}],
```

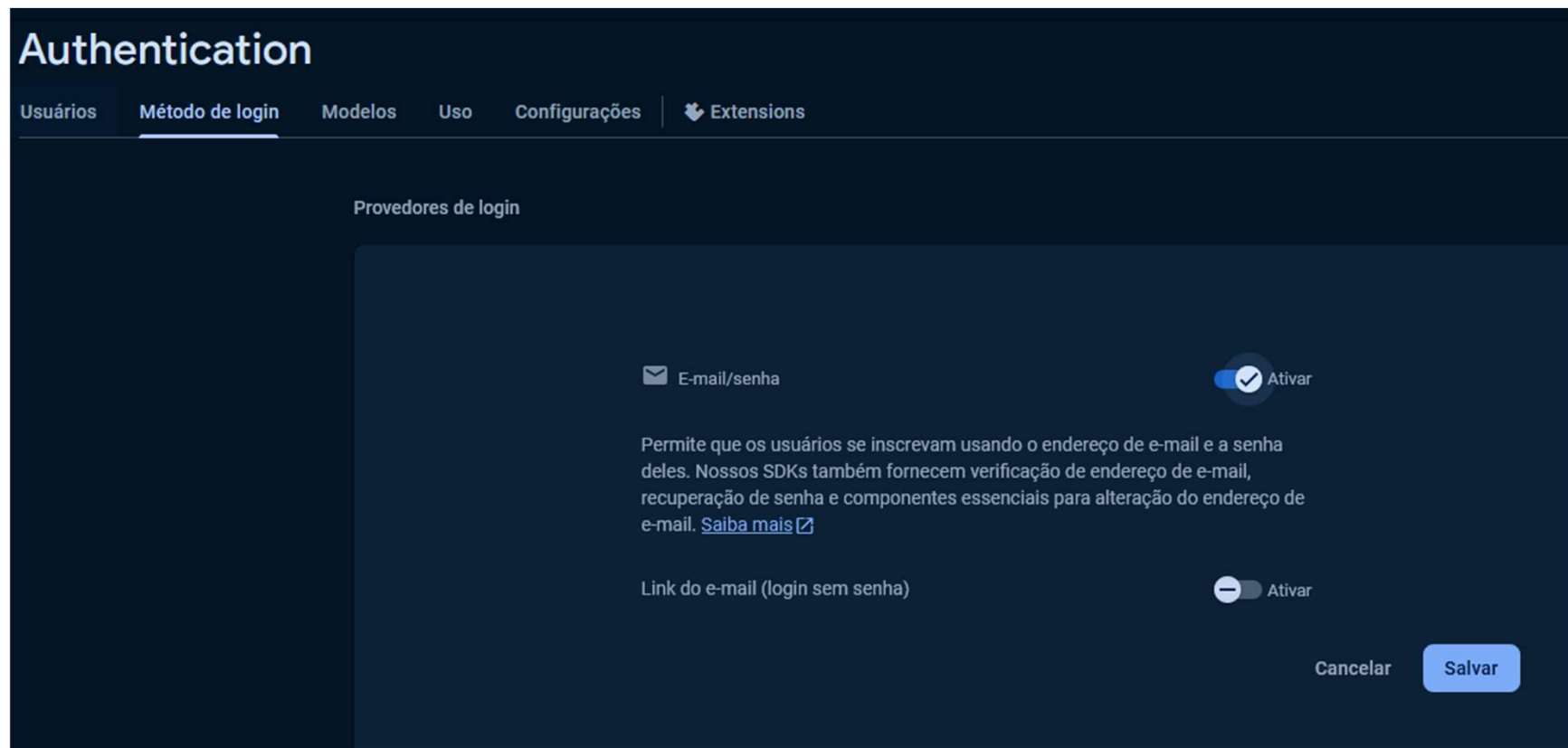
Criando serviço de autenticação no firebase - Nome de usuário e senha





Criando serviço de autenticação no firebase - Nome de usuário e senha



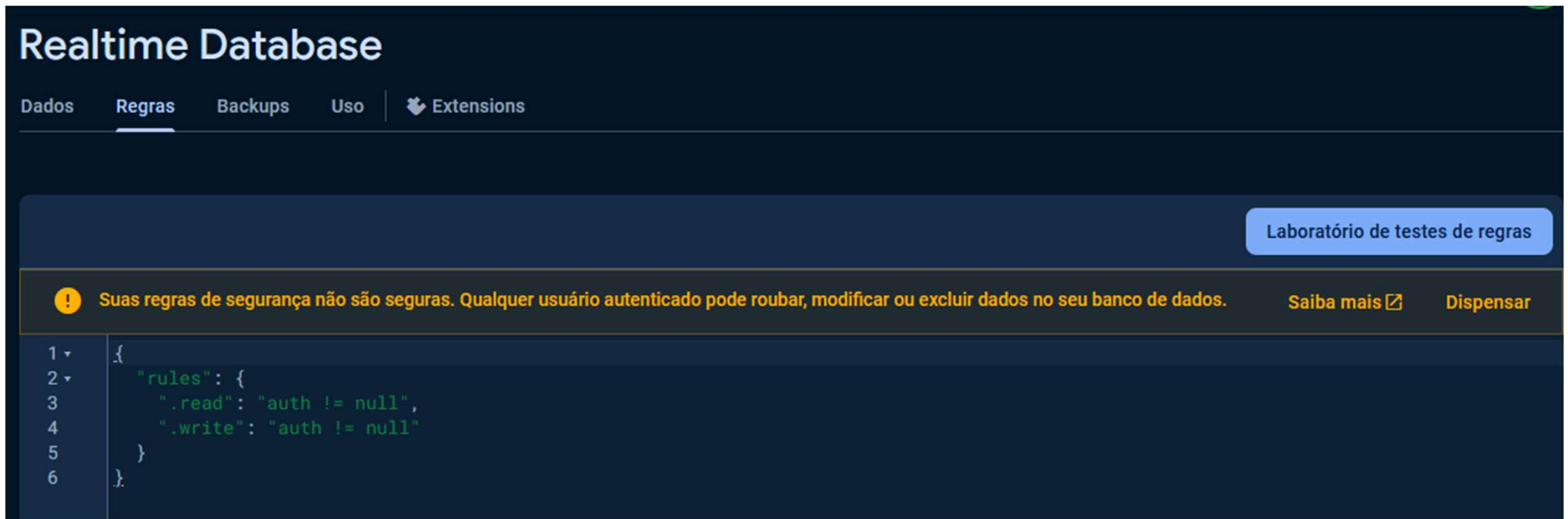
Criando serviço de autenticação no firebase - Nome de usuário e senha



Método de login

| Provedores de login | |
|--|--|
| <div>Adicionar novo fornecedor</div> | |
| Provedor | Status |
|  E-mail/senha |  ativado |

Criando serviço de autenticação no firebase - Nome de usuário e senha



The screenshot shows the 'Realtime Database' interface in the Firebase console. The 'Regras' (Rules) tab is selected. A warning message states: 'Suas regras de segurança não são seguras. Qualquer usuário autenticado pode roubar, modificar ou excluir dados no seu banco de dados.' (Your security rules are not secure. Any authenticated user can steal, modify, or delete data in your database). Below the warning, the current rules are displayed in a code editor:

```
1 {  
2   "rules": {  
3     ".read": "auth != null",  
4     ".write": "auth != null"  
5   }  
6 }
```

Navigation tabs at the top include 'Dados', 'Regras', 'Backups', 'Uso', and 'Extensions'. A button for 'Laboratório de testes de regras' (Rules Test Lab) is visible in the top right.

Firestore Rest API – Se inscrever

Inscreva-se com e-mail/senha

Você pode criar um novo usuário de e-mail e senha emitindo uma solicitação HTTP `POST` para o ponto de extremidade Auth `signupNewUser`.

Método: POST

Tipo de conteúdo: application/json

Endpoint

`https://identitytoolkit.googleapis.com/v1/accounts:signup?key=[API_KEY]`



Request Body Payload

| Nome da propriedade | Modelo | Descrição |
|---------------------|----------|---|
| o email | corda | O e-mail para o usuário criar. |
| senha | corda | A senha para o usuário criar. |
| returnSecureToken | booleano | Se deve ou não retornar um ID e um token de atualização. Deve ser sempre verdade. |

<https://firebase.google.com/docs/reference/rest/auth?hl=pt-br#section-create-email-password>

Firestore Rest API – Se inscrever

Carga de resposta

| Nome da propriedade | Modelo | Descrição |
|---------------------|--------|---|
| idToken | corda | Um token de ID de autenticação do Firebase para o usuário recém-criado. |
| o email | corda | O e-mail do usuário recém-criado. |
| refreshToken | corda | Um token de atualização do Firebase Auth para o usuário recém-criado. |
| expira em | corda | O número de segundos em que o token de ID expira. |
| localId | corda | O uid do usuário recém-criado. |

Firebase Rest API – Login

Sign in with email / password

You can sign in a user with an email and password by issuing an HTTP `POST` request to the Auth `verifyPassword` endpoint.

Method: POST

Content-Type: application/json

Endpoint

```
https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=[API_KEY]
```



Request Body Payload

| Property Name | Type | Description |
|-------------------|---------|--|
| email | string | The email the user is signing in with. |
| password | string | The password for the account. |
| returnSecureToken | boolean | Whether or not to return an ID and refresh token. Should always be true. |

Firestore Rest API – Login

Response Payload

| Property Name | Type | Description |
|---------------|---------|---|
| idToken | string | A Firebase Auth ID token for the authenticated user. |
| email | string | The email for the authenticated user. |
| refreshToken | string | A Firebase Auth refresh token for the authenticated user. |
| expiresIn | string | The number of seconds in which the ID token expires. |
| localId | string | The uid of the authenticated user. |
| registered | boolean | Whether the email is for an existing account. |

Sample request

<https://firebase.google.com/docs/reference/rest/auth#section-sign-in-email-password>

BehaviorSubject vs Subject

- Ao assinar um Subject, não é obtido o valor atual ou o valor inicial;
- Ao se inscrever em um BehaviorSubject, um valor atual ou inicial é obtido;
- Valor Inicial
 - Para um Subject, não é preciso inicializar um valor;
 - Mas sempre que é declarado um BehaviorSubject é preciso inicializar um valor padrão.
- Subscribers
 - Para um Subject , os assinantes receberão apenas o próximo valor;
 - No BehaviorSubject, os assinantes receberão o valor anterior e também o valor futuro.

Bootstrap

- `ng add @ng-bootstrap/ng-bootstrap`

```
i Using package manager: npm
✓ Found compatible package version: @ng-bootstrap/ng-bootstrap@15.1.2.
✓ Package information loaded.

The package @ng-bootstrap/ng-bootstrap@15.1.2 will be installed and executed.
Would you like to proceed? (Y/n) █
```

```
The package @ng-bootstrap/ng-bootstrap@15.1.2 will be installed and executed.
Would you like to proceed? Yes
✓ Packages successfully installed.
UPDATE package.json (1183 bytes)
✓ Packages installed successfully.
UPDATE src/app/app.module.ts (459 bytes)
UPDATE angular.json (2797 bytes)
UPDATE src/main.ts (259 bytes)
UPDATE tsconfig.app.json (294 bytes)
UPDATE tsconfig.spec.json (300 bytes)
```

Referências

- <https://ng-bootstrap.github.io/#/home>
- <https://firebase.google.com/docs/reference/rest/database?hl=pt-br>
- <https://firebase.google.com/docs/reference/rest/auth?hl=pt-br>
- <https://firebase.google.com/docs/reference/rest/auth?hl=pt-br#section-create-email-password>
- <https://firebase.google.com/docs/reference/rest/auth#section-sign-in-email-password>
- <https://loading.io/css/>