THE DEVELOPER'S CONFERENCE

API – Desmistificando o OAuth2

Maxmiliano Reipert Andriani
Arquiteto de Software

Sumário

THE DEVELOPER'S CONFERENCE

- Ambientação
- OAuth2
- Clients
- Authorization Flow
- Device Flow
- Auth Request
- Scope
- Redirect Url
- State
- PKCE
- Tokens Request
- Errors

Sumário

THE DEVELOPER'S CONFERENCE

- ✓ Ambientação
- ✓ OAuth2
- ✓ Clients
- ✓ Authorization Flow
- ✓ Device Flow
- ✓ Auth Request
- ✓ Scope
- ✓ Redirect Url
- ✓ State
- ✓ PKCE
- ✓ Tokens Request
- ✓ Errors


No princípio, tudo era mato!

Então vieram Desktops, Websites, Integrações

E o JavaScript tocou as trombetas do apocalípse

THE DEVELOPER'S CONFERENCE

Além dos computadores,
outros dispositivos também queriam
acessar seus dados.

OAuth2

THE DEVELOPER'S CONFERENCE

Quatro bravos guerreiros se erguem

Flickr Auth

AuthSub

BBAuth

OAuth Core 1.0

OAuth2

THE DEVELOPER'S CONFERENCE

> We want something like Flickr Auth / Google AuthSub / Yahoo! BBAuth, but published as an open standard, with common server and client libraries, etc.

Blaine Cook, April 5, 2007

OAuth2

THE DEVELOPER'S CONFERENCE

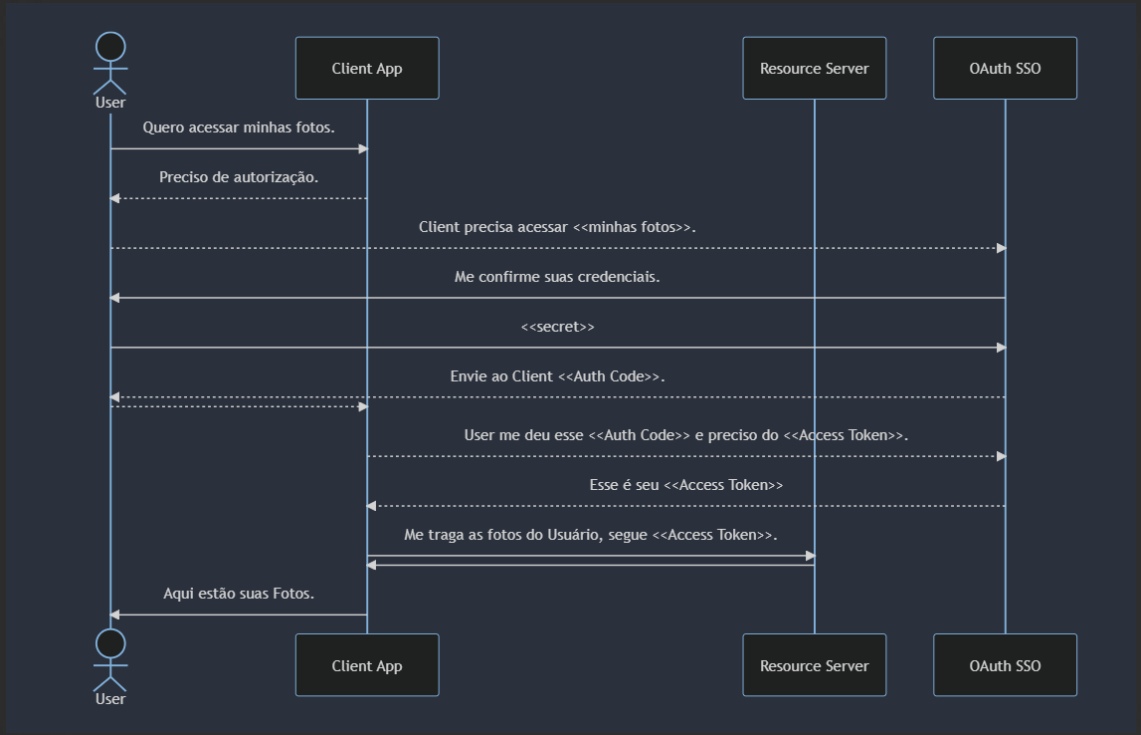
A Sociedade do OAuth

Clients

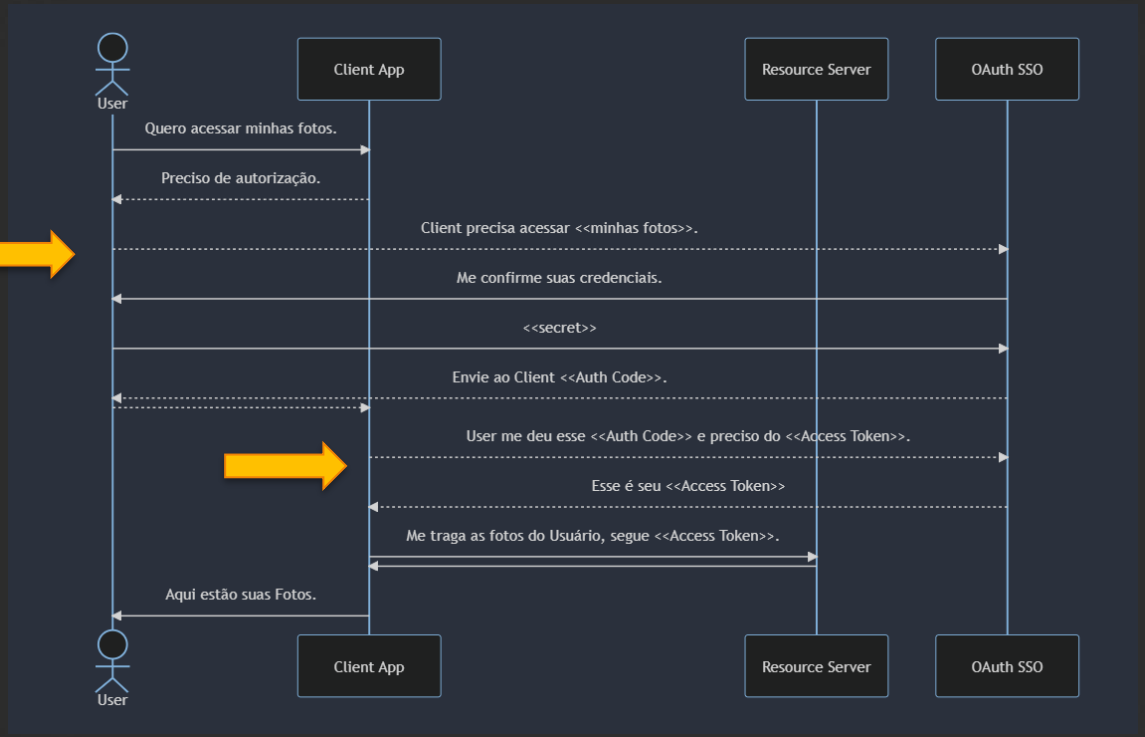
Público

Confidencial

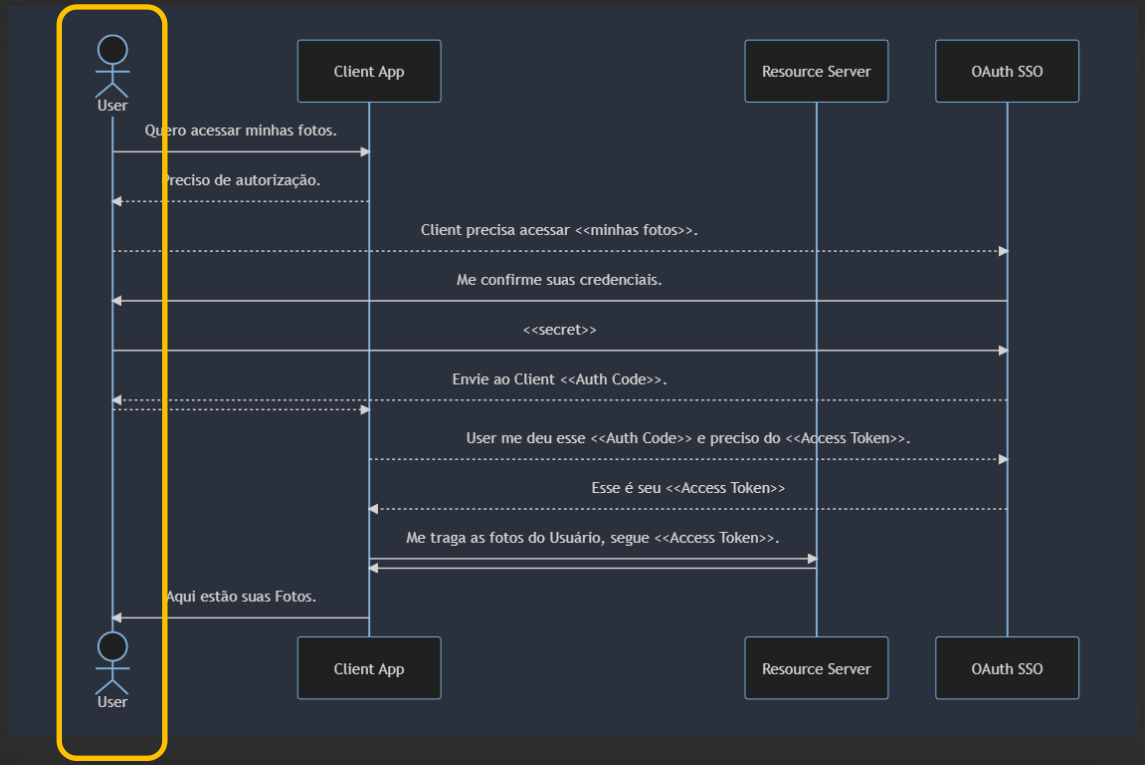
Authorization Flow



Authorization Flow



Authorization Flow



Authorization Flow

THE DEVELOPER'S CONFERENCE

```
5
6   export function startAuthorizationFlow({
7     authorizationUrl,
8     clientId,
9     redirectUri
10  }) {
11    const params = new URLSearchParams();
12    params.set('response_type', 'code');
13    params.set('client_id', clientId);
14    params.set('redirect_uri', redirectUri);
15    return `${authorizationUrl}?${params.toString()}`;
16  }
17
```

Response_type:

- code
- id_token
- token
- none

<https://openid.net/specs/oauth-v2-multiple-response-types-1_0.html#ResponseTypesAndModes>

redirect_uri

THE DEVELOPER'S CONFERENCE

```
https://example.com/dashboard
```

<https://sec.okta.com/articles/2021/02/stealing-oauth-tokens-open-redirects>

redirect_uri

THE DEVELOPER'S CONFERENCE

`https://example.com/dashboard`

`https://example.com/login?redirect=https://example.com/dashboard`

<https://sec.okta.com/articles/2021/02/stealing-oauth-tokens-open-redirects>

redirect_uri

THE DEVELOPER'S CONFERENCE

```
https://auth.com?...&redirect_uri=https://example.com/login?redirect=https://example.com/dashboard
```

<https://sec.okta.com/articles/2021/02/stealing-oauth-tokens-open-redirects>

redirect_uri

THE DEVELOPER'S CONFERENCE

```
https://auth.com?...&redirect_uri=https://example.com/login?redirect=https://attacker.com
```

OPEN REDIRECT ATTACK

<https://sec.okta.com/articles/2021/02/stealing-oauth-tokens-open-redirects>

state


```
5
6   export function startAuthorizationFlow({
7     authorizationUrl,
8     clientId,
9     redirectUri,
10    state
11  }) {
12    const params = new URLSearchParams();
13    params.set('response_type', 'code');
14    params.set('client_id', clientId);
15    params.set('redirect_uri', redirectUri);
16    params.set('state', state);
17    return `${authorizationUrl}?${params.toString()}`;
18  }
19
```

Um valor opaco para o cliente registrar sua sessão.

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#section-4.1.1>

scope


```
5
6   export function startAuthorizationFlow({
7     authorizationUrl,
8     clientId,
9     redirectUri,
10    state,
11    scope
12  }) {
13    const params = new URLSearchParams();
14    params.set('response_type', 'code');
15    params.set('client_id', clientId);
16    params.set('redirect_uri', redirectUri);
17    params.set('state', state);
18    params.set('scope', scope);
19    return `${authorizationUrl}?${params.toString()}`;
20  }
21
```

- profile
- openid
- email
- roles
- publish_photo
- publish_post
- accounts
- invoices
- full_access

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#access-token-scope>


OAKLEY

csrf

THE DEVELOPER'S CONFERENCE

CROSS-SITE REQUEST FORGERY

<https://en.wikipedia.org/wiki/Cross-site_request_forgery>

csrf

THE DEVELOPER'S CONFERENCE

CROSS-SITE REQUEST FORGERY

https://snap.com/publish

<https://en.wikipedia.org/wiki/Cross-site_request_forgery>

csrf

THE DEVELOPER'S CONFERENCE

CROSS-SITE REQUEST FORGERY

https://snap.com/publish

https://auth.com?cliente_id=snap&...

<https://en.wikipedia.org/wiki/Cross-site_request_forgery>

csrf

THE DEVELOPER'S CONFERENCE

CROSS-SITE REQUEST FORGERY

https://snap.com/publish

https://auth.com?cliente_id=snap&...

https://snap.com/oauth?code=attackercode&...

<https://en.wikipedia.org/wiki/Cross-site_request_forgery>

pkce


```
5
6   export function startAuthorizationFlow({
7     authorizationUrl,
8     clientId,
9     redirectUri,
10    state,
11    scope,
12    codeChallenge,
13    codeChallengeMethod = 'S256'
14  }) {
15    const params = new URLSearchParams();
16    params.set('response_type', 'code');
17    params.set('client_id', clientId);
18    params.set('redirect_uri', redirectUri);
19    params.set('state', state);
20    params.set('scope', scope);
21    if (!!codeChallenge) {
22      params.set('code_challenge_method', codeChallengeMethod);
23      params.set('code_challenge', codeChallenge);
24    }
25    return `${authorizationUrl}?${params.toString()}`;
26  }
27
```

code_challenge_method

- plain
- S256

<https://www.rfc-editor.org/rfc/rfc7636.txt>

pkce

THE DEVELOPER'S CONFERENCE

```
31
32  export async function generatePkcePair(): Promise<[string, string]> {
33    const codeVerifier = await generateRandomString();
34    const codeVerifierHash = await sha256(codeVerifier).then(buffer => base64URLEncode(buffer));
35    return [codeVerifier, codeVerifierHash];
36  }
37
```

<https://www.rfc-editor.org/rfc/rfc7636.txt>

pkce

THE DEVELOPER'S CONFERENCE

```
4
5   const PKCE_DICT = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~';
6
7   export async function generateRandomString(size: number = 128) {
8     const buffer = new Uint8Array(128);
9     crypto.getRandomValues(buffer);
10    return Array.from(buffer).map(x => PKCE_DICT[x % 64]).join('');
11  }
12
```

<https://www.rfc-editor.org/rfc/rfc7636.txt>

pkce

THE DEVELOPER'S CONFERENCE

```
15
16  export function base64URLEncode(buffer: ArrayBuffer) {
17    return btoa(Array.from(new Uint8Array(buffer)).map(bytes => String.fromCharCode(bytes)).join(''))
18      .replace(/\+/g, '-')
19      .replace(/\//g, '_')
20      .replace(/=+$/, '');
21  }
22
23  export async function sha256(phrase: string): Promise<ArrayBuffer> {
24    const encoder = new TextEncoder();
25    const buffer = encoder.encode(phrase);
26    return crypto.subtle.digest('SHA-256', buffer);
27  }
28
```

<https://www.rfc-editor.org/rfc/rfc7636.txt>

pkce


```
5
6   export function startAuthorizationFlow({
7     authorizationUrl,
8     clientId,
9     redirectUri,
10    state,
11    scope,
12    codeChallenge,
13    codeChallengeMethod = 'S256'
14  }) {
15    const params = new URLSearchParams();
16    params.set('response_type', 'code');
17    params.set('client_id', clientId);
18    params.set('redirect_uri', redirectUri);
19    params.set('state', state);
20    params.set('scope', scope);
21    if (!!codeChallenge) {
22      params.set('code_challenge_method', codeChallengeMethod);
23      params.set('code_challenge', codeChallenge);
24    }
25    return `${authorizationUrl}?${params.toString()}`;
26  }
27
```

code_challenge_method

- plain
- S256

<https://www.rfc-editor.org/rfc/rfc7636.txt>

Token Request

THE DEVELOPER'S CONFERENCE

```
24
25  export async function fetchAccessTokenByAuthCode({
26    authCode, clientId, codeVerifier, redirectUri, tokenUrl
27  }): Promise<AccessTokenResponse> {
28    const body = new URLSearchParams();
29    body.set('client_id', clientId);
30    body.set('grant_type', 'authorization_code');
31    body.set('code', authCode);
32    body.set('redirect_uri', redirectUri);
33
34    if (codeVerifier) {
35      body.set('code_verifier', codeVerifier);
36    }
37
38    return fetch(tokenUrl, {
39      headers: { 'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8' },
40      body,
41      method: 'POST'
42    })
43      .then(handleOAuthErrorResponse)
44      .then(asJsonResponse);
45  }
```

grant_type

- authorization_code
- client_credentials
- refresh_token
- urn:ietf:params:oauth:grant-type:device_code

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#section-4.1>

Token Response

THE DEVELOPER'S CONFERENCE

```
12
13  export interface AccessTokenResponse {
14    access_token: string,
15    refresh_token?: string,
16    token_type: string,
17    expires_in: number,
18    refresh_expires_in?: number,
19    scope?: string,
20    session_state?: string,
21    id_token?: string
22  }
23
```

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#section-4.1>

THE DEVELOPER'S CONFERENCE

E quando não houver Browser?

Device Flow

THE DEVELOPER'S CONFERENCE

```
37
38  export async function startDeviceAuthorizationFlow({
39    clientId, deviceUrl, scope
40  }): Promise<DeviceAuthorizationResponse> {
41    const body = new URLSearchParams();
42    body.set('client_id', clientId);
43    body.set('scope', scope);
44    return fetch(deviceUrl, {
45      headers: { 'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8' },
46      body,
47      method: 'POST'
48    })
49      .then(handleOAuthErrorResponse)
50      .then(asJsonResponse);
51  }
52
```

<https://www.rfc-editor.org/rfc/rfc8628.txt>

Device Flow Response

THE DEVELOPER'S CONFERENCE

```
25
26  export interface DeviceAuthorizationResponse {
27    device_code: string;
28    user_code: string;
29    verification_uri: string;
30    interval: number;
31    expires_in: number;
32  }
33
```

<https://www.rfc-editor.org/rfc/rfc8628.txt>

Token Request

THE DEVELOPER'S CONFERENCE

```
55
56  export async function fetchAccessTokenByDeviceCode({
57    clientId, deviceCode, tokenUrl
58  }): Promise<AccessTokenResponse> {
59    const body = new URLSearchParams();
60
61    body.set('client_id', clientId);
62    body.set('grant_type', 'urn:ietf:params:oauth:grant-type:device_code');
63    body.set('device_code', deviceCode);
64
65    return fetch(tokenUrl, {
66      headers: { 'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8' },
67      body,
68      method: 'POST'
69    })
70      .then(handleOAuthErrorResponse)
71      .then(asJsonResponse);
72  }
```

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#section-4.1>

Error Response

THE DEVELOPER'S CONFERENCE

```
5
6   export interface OAuthErrorResponse {
7     error: string,
8     error_description?: string,
9     error_uri?: string,
10    state?: string
11  }
12
```

error
- slow_down
- authorization_pending

<https://www.rfc-editor.org/rfc/rfc8628.txt>

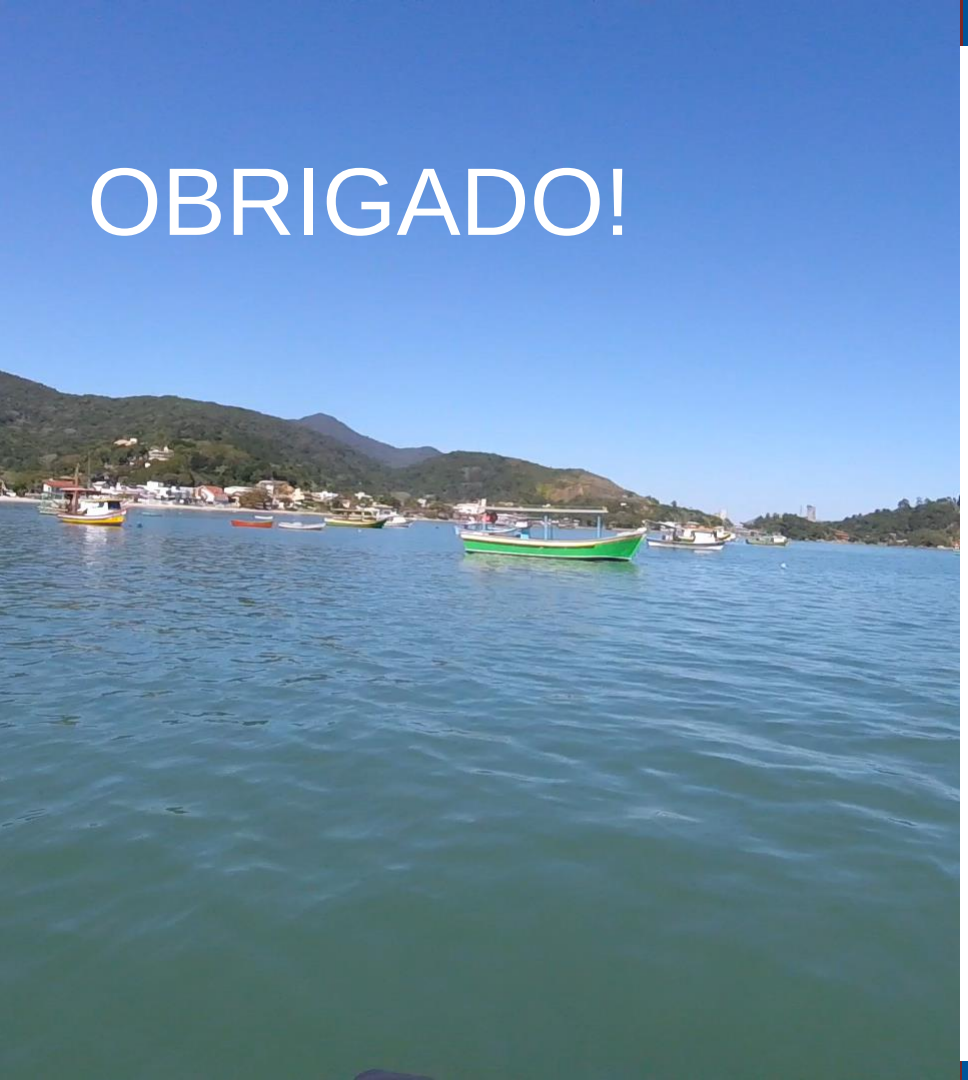
Token Response

THE DEVELOPER'S CONFERENCE

```
12
13  export interface AccessTokenResponse {
14    access_token: string,
15    refresh_token?: string,
16    token_type: string,
17    expires_in: number,
18    refresh_expires_in?: number,
19    scope?: string,
20    session_state?: string,
21    id_token?: string
22  }
23
```

<https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-08.html#section-4.1>

OBRIGADO!



THE DEVELOPER'S CONFERENCE

https://github.com/maxandriani/tdc-2023-oauth
https://twitter.com/maxandriani
https://www.instagram.com/maxandriani

THE
DEVELOPER'S
CONFERENCE

THE DEVELOPER'S CONFERENCE