supabase / **auth-js** Public archive

Code | Issues 13 | Pull requests 5 | Security and quality 1 | Insights

Commit 1bcb76e

hf authored on May 12, 2025 · 5 / 5 · Verified

```
feat: validate uuid and sign out scope parameters to functions (#1063)

You're not supposed to pass non-UUID values for parameters which are
meant to be UUID, as well as the sign out scope parameter.
```

master (#1063) · v2.72.0 ••• rc2.70.0-rc.7

1 parent eff5048 commit 1bcb76e

4 files changed +55 -19

Top

Filter files…

- src
  - GoTrueAdminApi.ts
  - lib
    - helpers.ts
    - types.ts
- test
  - GoTrueApi.test.ts

Search within code

src/GoTrueAdminApi.ts

```
@@ -5,7 +5,7 @@ import {
5  5    _request,
```

```
6   6     _userResponse,
7   7   } from './lib/fetch'
8       - import { resolveFetch } from './lib/helpers'
    8   + import { resolveFetch, validateUUID } from './lib/helpers'
9   9   import {
10  10    AdminUserAttributes,
11  11    GenerateLinkParams,
    ↕     @@ -19,6 +19,8 @@ import {
19  19    AuthMFAAdminListFactorsParams,
20  20    AuthMFAAdminListFactorsResponse,
21  21    PageParams,
    22  +   SIGN_OUT_SCOPES,
    23  +   SignOutScope,
22  24  } from './lib/types'
23  25  import { AuthError, isAuthError } from './lib/errors'
24  26
    ↕     @@ -59,8 +61,14 @@ export default class GoTrueAdminApi {
59  61      */
60  62    async signOut(
61  63      jwt: string,
62      -     scope: 'global' | 'local' | 'others' = 'global'
    64  +     scope: SignOutScope = SIGN_OUT_SCOPES[0]
63  65    ): Promise<{ data: null; error: AuthError | null }> {
    66  +     if (SIGN_OUT_SCOPES.indexOf(scope) < 0) {
    67  +       throw new Error(
    68  +         `@supabase/auth-js: Parameter scope must be one of
        ${SIGN_OUT_SCOPES.join(', ')}`
    69  +       )
    70  +     }
    71  +
64  72      try {
65  73        await _request(this.fetch, 'POST', `${this.url}/logout?scope=${scope}`, {
66  74          headers: this.headers,
    ↕     @@ -219,6 +227,8 @@ export default class GoTrueAdminApi {
219 227     * This function should only be called on a server. Never expose your
        `service_role` key in the browser.
220 228     */
221 229    async getUserById(uid: string): Promise<UserResponse> {
```

```
                 +     validateUUID(uid)
                 +
222  232           try {
223  233             return await _request(this.fetch, 'GET',
                   `${this.url}/admin/users/${uid}`, {
224  234               headers: this.headers,
                   @@ -241,6 +251,8 @@ export default class GoTrueAdminApi {
241  251          * This function should only be called on a server. Never expose your
                   `service_role` key in the browser.
242  252          */
243  253         async updateUserById(uid: string, attributes: AdminUserAttributes):
                   Promise<UserResponse> {
                 +     validateUUID(uid)
                 +
244  256           try {
245  257             return await _request(this.fetch, 'PUT',
                   `${this.url}/admin/users/${uid}`, {
246  258               body: attributes,
                   @@ -266,6 +278,8 @@ export default class GoTrueAdminApi {
266  278          * This function should only be called on a server. Never expose your
                   `service_role` key in the browser.
267  279          */
268  280         async deleteUser(id: string, shouldSoftDelete = false): Promise<UserResponse>
                   {
                 +     validateUUID(id)
                 +
269  283           try {
270  284             return await _request(this.fetch, 'DELETE',
                   `${this.url}/admin/users/${id}`, {
271  285               headers: this.headers,
                   @@ -286,6 +300,8 @@ export default class GoTrueAdminApi {
286  300         private async _listFactors(
287  301           params: AuthMFAAdminListFactorsParams
288  302         ): Promise<AuthMFAAdminListFactorsResponse> {
                 +     validateUUID(params.userId)
                 +
289  305           try {
290  306             const { data, error } = await _request(
291  307               this.fetch,
                   @@ -311,6 +327,9 @@ export default class GoTrueAdminApi {
```

```
311  327      private async _deleteFactor(
312  328        params: AuthMFAAdminDeleteFactorParams
313  329      ): Promise<AuthMFAAdminDeleteFactorResponse> {
     330  +     validateUUID(params.userId)
     331  +     validateUUID(params.id)
     332  +
314  333        try {
315  334          const data = await _request(
316  335            this.fetch,
```

src/lib/helpers.ts

```
     ...      @@ -1,6 +1,6 @@
1    1        import { API_VERSION_HEADER_NAME, BASE64URL_REGEX } from './constants'
2    2        import { AuthInvalidJwtError } from './errors'
3         -  import { base64UrlToUint8Array, stringFromBase64URL, stringToBase64URL } from
           './base64url'
     3    +  import { base64UrlToUint8Array, stringFromBase64URL } from './base64url'
4    4        import { JwtHeader, JwtPayload, SupportedStorage } from './types'
5    5
6    6        export function expiresAt(expiresIn: number) {
         ↓ ↑  @@ -357,3 +357,11 @@ export function getAlgorithm(alg: 'RS256' | 'ES256'):
           RsaHashedImportParams | Ec
357  357            throw new Error('Invalid alg claim')
358  358          }
359  359        }
     360  +
     361  +  const UUID_REGEX = /^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]
           {12}$/
     362  +
     363  +  export function validateUUID(str: string) {
     364  +    if (!UUID_REGEX.test(str)) {
     365  +      throw new Error('@supabase/auth-js: Expected parameter to be UUID but is
           not')
     366  +    }
     367  +  }
```

src/lib/types.ts

```
         ↑    @@ -1279,3 +1279,6 @@ export interface JWK {
```

```
1279  1279      kid?: string
1280  1280      [key: string]: any
1281  1281    }
      1282  +
      1283  + export const SIGN_OUT_SCOPES = ['global', 'local', 'others'] as const
      1284  + export type SignOutScope = typeof SIGN_OUT_SCOPES[number]
```

test/GoTrueApi.test.ts

```
      @@ -16,7 +16,7 @@ import {
16    16    import type { GenerateLinkProperties, User } from '../src/lib/types'
17    17
18    18    const INVALID_EMAIL = 'xx:;x@x.x'
19        - const INVALID_USER_ID = 'invalid-uuid'
      19  + const NON_EXISTANT_USER_ID = '83fd9e20-7a80-46e4-bf29-a86e3d6bbf66'
20    20
21    21    describe('GoTrueAdminApi', () => {
22    22      describe('User creation', () => {
      @@ -152,7 +152,7 @@ describe('GoTrueAdminApi', () => {
152   152       })
153   153
154   154       test('getUserById() returns AuthError when user id is invalid', async () =>
      {
155       -       const { error, data } = await
      serviceRoleApiClient.getUserById(INVALID_USER_ID)
      155 +       const { error, data } = await
      serviceRoleApiClient.getUserById(NON_EXISTANT_USER_ID)
156   156
157   157         expect(error).not.toBeNull()
158   158         expect(data.user).toBeNull()
      @@ -283,7 +283,7 @@ describe('GoTrueAdminApi', () => {
283   283       })
284   284
285   285       test('deleteUser() returns AuthError when user id is invalid', async () =>
      {
286       -       const { error, data } = await
      serviceRoleApiClient.deleteUser(INVALID_USER_ID)
      286 +       const { error, data } = await
      serviceRoleApiClient.deleteUser(NON_EXISTANT_USER_ID)
```

```
287  287
288  288          expect(error).not.toBeNull()
289  289          expect(data.user).toBeNull()
     ↕            @@ -479,7 +479,7 @@ describe('GoTrueAdminApi', () => {
479  479        test('listUsers() returns AuthError when page is invalid', async () => {
480  480          const { error, data } = await serviceRoleApiClient.listUsers({
481  481            page: -1,
482       -          perPage: 10
     482  +          perPage: 10,
483  483          })
484  484
485  485          expect(error).not.toBeNull()
     ↕            @@ -489,8 +489,8 @@ describe('GoTrueAdminApi', () => {
489  489
490  490      describe('Update User', () => {
491  491        test('updateUserById() returns AuthError when user id is invalid', async ()
                 => {
492       -        const { error, data } = await
                 serviceRoleApiClient.updateUserById(INVALID_USER_ID, {
493       -          email: 'new@email.com'
     492  +        const { error, data } = await
                 serviceRoleApiClient.updateUserById(NON_EXISTANT_USER_ID, {
     493  +          email: 'new@email.com',
494  494          })
495  495
496  496          expect(error).not.toBeNull()
     ↕            @@ -513,7 +513,7 @@ describe('GoTrueAdminApi', () => {
513  513          expect(uid).toBeTruthy()
514  514
515  515          const { error: enrollError } = await authClientWithSession.mfa.enroll({
516       -          factorType: 'totp'
     516  +          factorType: 'totp',
517  517          })
518  518          expect(enrollError).toBeNull()
519  519
     ↕            @@ -526,35 +526,41 @@ describe('GoTrueAdminApi', () => {
526  526
527  527          const factorId = data?.factors[0].id
528  528          expect(factorId).toBeDefined()
```

```
529       -       const { data: deletedData, error: deletedError } = await serviceRoleApiClient.mfa.deleteFactor({
530       -         userId: uid,
531       -         id: factorId!
532       -       })
      529 +       const { data: deletedData, error: deletedError } =
      530 +         await serviceRoleApiClient.mfa.deleteFactor({
      531 +           userId: uid,
      532 +           id: factorId!,
      533 +         })
533   534           expect(deletedError).toBeNull()
534   535           expect(deletedData).not.toBeNull()
535   536           const deletedId = (deletedData as any)?.data?.id
536   537           console.log('deletedId:', deletedId)
537   538           expect(deletedId).toEqual(factorId)
538   539
539       -       const { data: latestData, error: latestError } = await serviceRoleApiClient.mfa.listFactors({ userId: uid })
      540 +       const { data: latestData, error: latestError } = await serviceRoleApiClient.mfa.listFactors({
      541 +         userId: uid,
      542 +       })
540   543           expect(latestError).toBeNull()
541   544           expect(latestData).not.toBeNull()
542   545           expect(Array.isArray(latestData?.factors)).toBe(true)
543   546           expect(latestData?.factors.length).toEqual(0)
544   547         })
545   548
546       -
547   549       test('mfa.listFactors returns AuthError for invalid user', async () => {
548       -       const { data, error } = await serviceRoleApiClient.mfa.listFactors({ userId: INVALID_USER_ID })
      550 +       const { data, error } = await serviceRoleApiClient.mfa.listFactors({
      551 +         userId: NON_EXISTANT_USER_ID,
      552 +       })
549   553           expect(data).toBeNull()
550   554           expect(error).not.toBeNull()
551   555         })
552   556
553   557       test('mfa.deleteFactors returns AuthError for invalid user', async () => {
```

```
554      -       const { data, error } = await serviceRoleApiClient.mfa.deleteFactor({
           userId: INVALID_USER_ID , id: '1' })
     558 +       const { data, error } = await serviceRoleApiClient.mfa.deleteFactor({
     559 +         userId: NON_EXISTANT_USER_ID,
     560 +         id: NON_EXISTANT_USER_ID,
     561 +       })
555  562           expect(data).toBeNull()
556  563           expect(error).not.toBeNull()
557  564         })
558      -
559  565       })
560  566     })
```

Comments 0

This repository has been archived.