BRAND SHELTER

DNS API

Documentation and Workflow

BRAND SHELTER

Introduction

GraphQL API Endpoint

We offer a GraphQL endpoint to interact with the DNS API:

https://secure.brandshelter.com/graphql

A great place to learn about GraphQL is here:

https://graphql.org/learn/

Authorization

Access Tokens

You can manage your access tokens in the Brandshelter frontend application. A access token´s “secret” is needed to create a valid JWT which is used to access the DNS API.

https://secure.brandshelter.com/access_tokens

JWT (JSON Web Token)

To get a JWT suitable for access to the GraphQL API, you need a user with a valid, not expired access token. Using the secret from that token, you can then create and sign a JWT with content:

- jti as a “globally unique” identifier.
- iss and sub as the user's login name.
- aud as the string “BrandShelter”.
- nbf must be a timestamp in the past. The "nbf" (not before) claim identifies the time before which the JWT is not valid.
- iat must be a timestamp in the past. The "iat" (issued at) claim identifies the time at which the JWT was issued.
- exp must be a timestamp in the future. The "exp" (expiration time) claim identifies the expiration time on or after which the JWT is invalid.

You then sign that JWT using the access token’s “secret” with algorithm “HS512” (HMAC with SHA-512). Doing so, you must make sure that some JWT headers are set appropriately:

- kid must be the user’s login name and the access token’s name separated by a forward slash, e.g., "john.doe@brandshelter.com/ExampleToken".
- alg must be “HS512”.

Example JWT header:

```
{
 "kid": "john.doe@brandshelter.com/ExampleToken",
 "alg": "HS512"
}
```

Example JWT payload:

```
{
 "iss": "john.doe@brandshelter.com",
 "sub": "john.doe@brandshelter.com",
"aud": "BrandShelter",
 "nbf": 1680522050,
 "exp": 1712102400,
 "iat": 1680594434,
 "jti": "BrandShelter-7a589414-d004-46b4-a95f-50b763f678d3"
}
```

You can use JWT.IO to create or verify JWTs.

HTTP Bearer Authentication

For authentication you must send the JWT in the Authorization header when making requests to DNS API:
Authorization: Bearer <JWT>

Introspection & Auto Documentation

The GraphQL introspection system allows to query information about the available schema. There are various tools and clients which are able to generate an API documentation from the live endpoint using a introspection query. You can use for example Altair GraphQL Client to easily generate and browse the API documentation.

Examples

```
query CurrentUser {
 currentUser {
   firstname
   lastname
   email
  }
}
```

```
query DnsZone {
  dnsZone(domainName: "test-domain-1001.net") {
   id
   name
   zoneType
  }
}
```

```
query DnsZones {
  dnsZones {
   nodes {
    id
    name
    zoneType
   }
 }

```

```
query DnsRecordsByZone {
 dnsRecordsByZone(domainName: "test-domain-1001.net") {
   bindSyntax
   errors
   fqdn
   host
  id
  locked
  Rdata
ttl
type
}
}
```

```
mutation CreateDnsRecord {
createDnsRecord(
input: { type: "MX", dnsZoneId: 2, rdata: "6 mail.foo-1.com." }
) {
id
bindSyntax
}
}
```

```
mutation DeleteDnsRecord {
deleteDnsRecord(id: 2076) {
id
}
}
```

```
mutation PublishDnsZone {
publishDnsZone(domainName: "test-domain-1001.net") {
name
 }
}
```