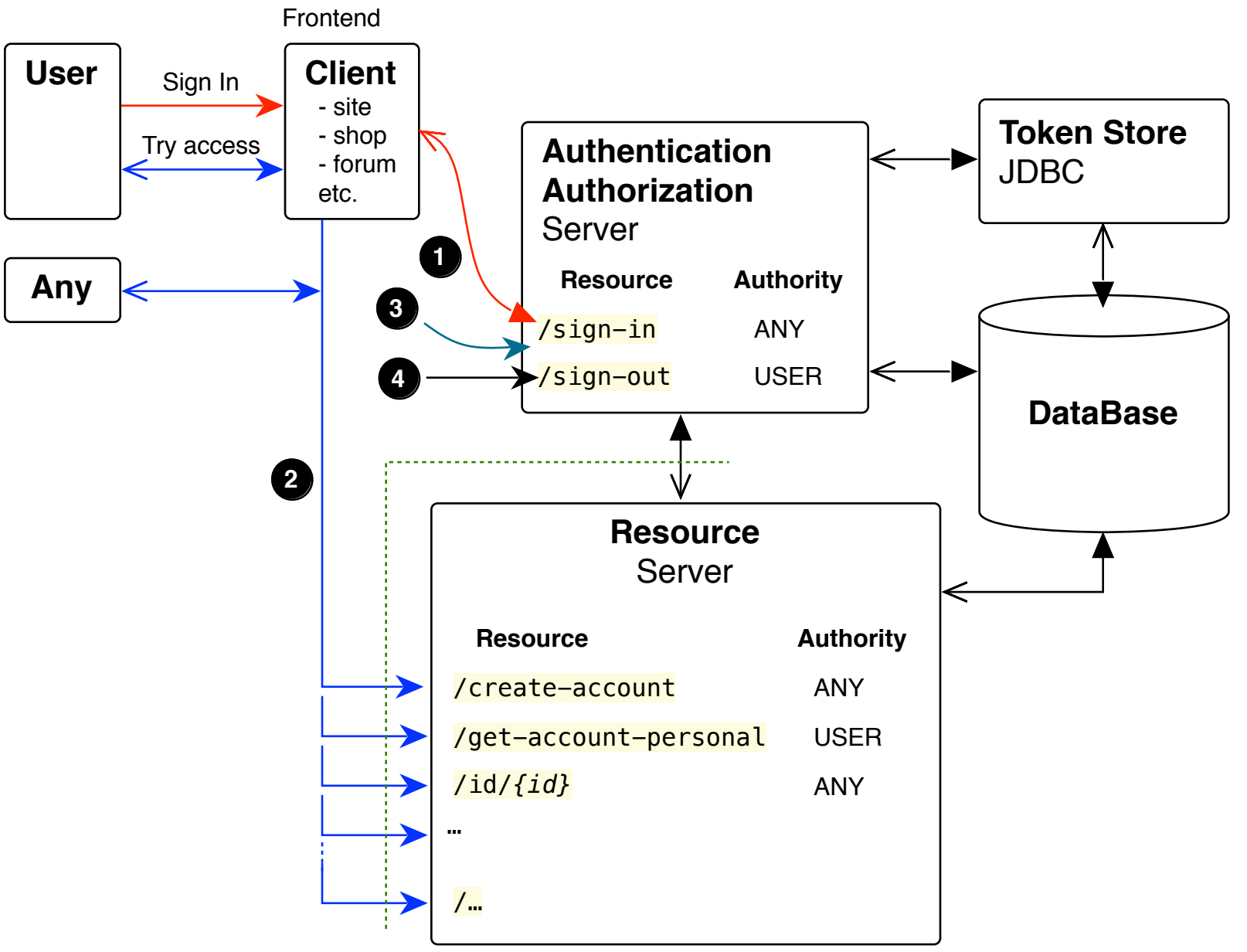
Lifecycle



1 Sign In

POST

```
Req: curl client_id : client_password @ host:port /sign-in
             -d username=username            required
             -d password=password            required
             -d grant_type=password          required
             -d usernameType=usernameType    optional
```

If you need to define a table where stored an ***username***, then, when sign-in, you need to pass an additional parameter, for example, this assumes that the parameter is named `usernameType`, its possible values are: - ***EMAIL***
- ***PHONE***
- ***NICKNAME***
- ***ID***

To avoid problems the values ***username*** and ***password*** passed in the request must contains characters from US-ASCII, and must be percent-encoded.

```
Resp: {
       access_token  : access token
       token_type    : "bearer"
       refresh_token : refresh token
       expires_in    : token expiration time in seconds
       scope         : list of scopes
       data: {
         id          : account id
         username
         authorities : list of authorities: [USER, ADMIN, etc.]
         createdOn   : YYYY-MM-DDTHH:MM:SSZ
       }
     }
```

2 Resource accessing

POST or GET

```
Req: curl host:port /path/to/resource -H "Authorization: Bearer ACCESS_TOKEN" ...
```

Resp: *resource content* or *error*

3 Token refreshing

POST

```
Req: curl client_id : client_password @ host:port /sign-in
            -d refresh_token=...
            -d grant_type=refresh_token
```

```
Resp: {
       access_token  : new access token
       token_type    : "bearer"
       refresh_token : refresh token
       expires_in    : token expiration time in seconds
       scope         : list of scopes
       data: {
         id          : account id
         username
         authorities : list of authorities: [USER, ADMIN, etc.]
         createdOn   : YYYY-MM-DDTHH:MM:SSZ
       }
     }
```

4 Sign Out

GET

```
Req: curl host:port /sign-out -H "Authorization: Bearer ACCESS_TOKEN"
```

Resp: *success* or *error*