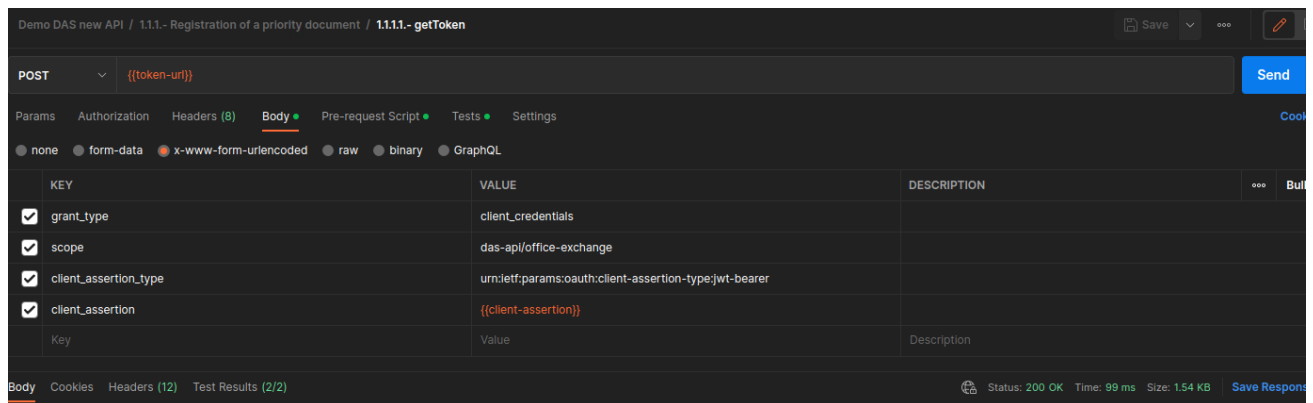


## b.2.2.1.- getToken

[ [Request configuration](#) ] [ [a.- Headers](#) ] [ [b.- Body](#) ] [ [c.- Pre-request script](#) ] [ [d.- Tests script](#) ] [ [Expected response](#) ] [ [e.- Body](#) ] [ [f.- Headers](#) ] [ [g.- Test results](#) ]

The Postman request:



### Request configuration

#### a.- Headers

Standard headers, nothing to update here.

	KEY	VALUE
<input checked="" type="checkbox"/>	Postman-Token	<calculated when request is sent>
<input checked="" type="checkbox"/>	Content-Type	application/x-www-form-urlencoded
<input checked="" type="checkbox"/>	Content-Length	<calculated when request is sent>
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>
<input checked="" type="checkbox"/>	User-Agent	PostmanRuntime/7.30.0
<input checked="" type="checkbox"/>	Accept	/*/*
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br
<input checked="" type="checkbox"/>	Connection	keep-alive

#### b.- Body

Values that are used for this demo:

Key	Value	Notes
grant_type	client_credentials	
scope	das-api/office-exchange	type of access to give to the user
client_assertion_type	urn:ietf:params:oauth:client-assertion-type:jwt-bearer	standard jwt bearer authentication
client_assertion	{{client-assertion}}	value to be retrieved from the Postman <a href="#">environment</a> . Before executing this request, please generate your assertion as per indicated in these pages: <a href="#">e.3.- Executing the sample scripts under linux</a> or <a href="#">e.4.- Executing the sample scripts via docker</a> . Then update the Postman <a href="#">environment</a> with the obtained JWT assertion

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL		
	KEY	VALUE
✓	grant_type	client_credentials
✓	scope	das-api/office-exchange
✓	client_assertion_type	urn:ietf:params:oauth:client-assertion-type:jwt-bearer
⋮	client_assertion	{{client-assertion}}
	Key	Value

c.- Pre-request script

Nothing here.

d.- Tests script

The following script is executed after the request is sent, to evaluate the response and also to set up environment variables (if they are needed for following requests)

```
// following test will verify the expected response code for this request
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});

// tests validate the reception of the access code from das
pm.test("Received ACCESS-TOKEN for DAS access", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.access_token).not.eql(undefined);
  if (jsonData.access_token !== undefined) {
    console.log("access-token received:"+jsonData.access_token)
  }
  //the new access token is saved in the postman environment for use in the other requests.
  pm.environment.set("access-token", jsonData.access_token);
});
```

Expected response

e.- Body

In this case the body will mainly show the access token provided y the time left for its use.

Body Cookies Headers (12) Test Results (2/2) Status: 200 OK Time: 2.21 s Size: 1.54 KB [Save Response](#)

Pretty Raw Preview Visualize JSON

```
1
2  "access_token": "eyJ0eXAiOiJKV1QiLCJraWQiOiJmVWRmbEJSa3c5bm1tejcrL3BmMmM5d2RYdXc9IiwiaWxnIjoilMyNTYifQ.eyJzdWIiOiJkYXNtaWItYXBpLWlkMSIsImN0cyI6Iik9BVVRIM19TVEFURUxhbnR1JBTlQlLCJhdwRpdFRyYWNaaw5nSwQiOiJlMmU3N2M2Zi00NTQ0LTQ1NDMtYjkwZiDg1NjczZGY5ZDgtNjA3MjM5IiwiaXNzIjoiaHR0cHM6Ly9sb2dpbmRldi53aXBvLmIudDo0NDMvYW0vb2F1dGgyIiwidG9rZW50YW11IjoiyWNjZXNzX3Rva2VuIiwidG95fdHlwZSI6IjYXJlciIsImF1dG8hcmFudElkIjoibk1ja2FhMjFkbG9yQkZkSkV0czJYNHk0Q1JFIiwiaWF0Ijoi2020-01-01T00:00:00.000Z",sImdyYW50X3R5cGU0IjoibGllbnRfY3JlZGVudGlhbHMlLCJzY29wZSI6WyJkYXNtaWxnbnR1JBTlQlLCJhdwRpdFRyYWNaaw5nSwQiOiJlMmU3N2M2Zi00NTQ0LTQ1NDMtYjkwZiIjoilYsImV4cCI6MTY3NzA2NTcwMCIwF0IjozNjc3MDYyMTAwLCJleHBpcnVzX2luIjozNjAwLCJqdGkiOiJqb0VRdV10bWwGR051Ym1id3NqMDNYbHRkM2cifQ.mor13vdkv3hqVsEb0-nm7zMPWxsUJ-YCW2n7RE2Jo5dTGMyVfX-02NaAK6B5e9-trLhGBNyEM2-YgFnNd9bBUJQZ6yMV1-436umzhV09naLgW6k4X-Upe2gragcnjS6CBfdFvksDXbcuE0tuASDgxbXFF7idWnnjFr8-ELRDVn7qckCm5PLsUXCU46kevZy8Uj9TFFUIEg_GNRYuKGbnY2eCEq4E47ahd6iUzefXrw5Mqwx_RqhHqyWI360bgozfaUHhGdrO4a9wPCfvpFReyQfsQr5073W3Gyboi69YX1qERxNcDxbenBd9Ed6dQ5tMDtQbF033g096qQ",
3  "scope": "das-api/office-exchange",
4  "token_type": "Bearer",
5  "expires_in": 3599
6
```

f.- Headers

Normal headers.

BodyCookiesHeaders (12)Test Results (2/2)

Status: 200 OKTime: 2.21 sSize: 1.54 KB

Save Response

KEY	VALUE
Date	Wed, 22 Feb 2023 10:35:00 GMT
Server	Apache
Strict-Transport-Security	max-age=31536000
X-Frame-Options	SAMEORIGIN
X-Content-Type-Options	nosniff
Pragma	no-cache
Cache-Control	no-store
Accept-Ranges	bytes
Vary	Accept-Charset,Accept-Encoding,Accept-Language,Accept,Origin
Content-Type	application/json
Connection	close
Transfer-Encoding	chunked

g.- Test results

This depends on the test we set up in the test script

BodyCookiesHeaders (12)Test Results (2/2)

AllPassedSkippedFailed

PASS

Status code is 200

PASS

Received ACCESS-TOKEN for DAS access