SAMS oAuth 2.0 Information Guide

Version: 1.0
Date: 02/16/2017
Remarks: Initial version

End Points:

1.	Authorization	**Production:** https://apigw.cdc.gov:8443/auth/oauth/v2/authorize **Staging:** https://apigw-stg.cdc.gov:8443/auth/oauth/v2/authorize
2.	Token	**Production:** https://apigw.cdc.gov:8443/auth/oauth/v2/token **Staging:** https://apigw-stg.cdc.gov:8443/auth/oauth/v2/token

3.	UserInfo	**Production:** https://apigw.cdc.gov:8443/openid/connect/v1/userinfo
		
		Staging: https://apigw-stg.cdc.gov:8443/openid/connect/v1/userinfo
		
		--- 	
		This userinfo endpoint provides information about an authenticated Person account (e.g. HHS users or SAMS external users). This endpoint can't be used with System account.
		
		Client needs to call Userinfo service along with a valid oAuth Token. It returns user information based on approved scopes for the oAuth Token.
		
		Following are the sample output for different scope values:
		
		openid:
		
		{
		"sub":"_Nnr2npeTv00Ae9wsNjcxUPeUb6T6qIOGy9EV0Id1gs"
		}
		
		openid profile :
		
		{
		"sub":"_Nnr2npeTv00Ae9wsNjcxUPeUb6T6qIOGy9EV0Id1gs",
		"profile":{
		"account_type":"person",
		"account_id":"7453",
		"name":"John K Doe",
		"family_name":"Doe",
		"middle_name":"K",
		"given_name":"John",

		<pre>"preferred_name":"John",
		"name_suffix":"Jr."
		}
		}</pre>
		openid profile email :
		<pre>{
		"sub":"_Nnr2npeTv00Ae9wsNjcxUPeUb6T6qIOGy9EV0Id1gs",
		"profile":{
		"account_type":"person",
		"account_id":"7453",
		"name":"John K Doe",
		"family_name":"Doe",
		"middle_name":"K",
		"given_name":"John",
		"preferred_name":"John",
		"name_suffix":"Jr."
		},
		"email":"jdoe@gmail.gov"
		}</pre>

4.	Token Validation	**Production:** https://apigw.cdc.gov:8443/sams/oauth/tokenvalidate
		
		Staging: https://apigw-stg.cdc.gov:8443/sams/oauth/tokenvalidate
		
		Clients can call this service to validate an oAuth Token. The service will return following output:
		
		If oAuth Token id valid:
		<pre>{
		"status":"ok",
		"Reason":"Valid Token"
		}</pre>
		If oAuth Token id not valid:
		<pre>{
		"status":"fail",
		"Reason":"<Error Message>"
		}</pre>

5.	UserInfosys	**Production:** https://apigw.cdc.gov:8443/openid/connect/v1/userinfosys **Staging:** https://apigw-stg.cdc.gov:8443/openid/connect/v1/userinfosys --- This endpoint provides information about authenticated system accounts. This endpoint can't be used with Person accounts. Clients' needs to provide a valid oAuth token to access it. This endpoint doesn't require any scope. Following is a sample output of this endpoint:

		{
		"sub":"_Nnr2npeTv00Ae9wsNjcxUPeUb6T6qIOGy9EV0Id1gs",
		"profile":{
		"account_type":"system",
		"account_id":"SYS-7453",
		"name":"John K Doe",
		"family_name":"Doe",
		"middle_name":"K",
		"given_name":"John",
		"preferred_name":"John",
		"name_suffix":"Jr."
		},
		"email":"jdoe@gmail.gov"
		}

oAuth Request Scenarios:

The following scenarios describe requests for access_token using a specified grant_type:

1. grant_type=authorization code

Exchange the authorization_code for an access_token. A client has received the authorization_code attached to a redirect URI. The client now exchanges the authorization_code for an access_token by using grant_type 'authorization_code'.

Request

Method: POST

Header: content-type: application/x-www-form-urlencoded

Header: authorization: Basic base64(client_id:client_secret) (This header can only be used if 'client_id' and 'client_secret' are **NOT** found within the message body and vice versa!)

Endpoint: /auth/oauth/v2/token

Parameters: grant_type=authorization_code&code=the-received-authorization-code&client_id=a-client_id&client_secret=a-client_secret&redirect_uri

Optional: **redirect_uri:** The value has to be included if it has been used in the initial request. It also has to match the original value

Response

Header: status: 200

Header: content-type: application/json

Body: { "access_token":"115b8c ... 11a5", "token_type":"Bearer", "expires_in":3600, "refresh_token":"74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

If the client included 'openid' as SCOPE in his request, additional keys are included in the response:

..."id_token":"eyJ0eXAiO1v8 ... JZu_LsN851VtfC5pcIqJc", "id_token_type":"urn:ietf:params:oauth:grant-type:jwt-bearer" ...

The id_token (JWT) can be used with grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer.

2. grant_type=client_credentials

This grant_type can be used if the client is acting on its own behalf. No user authentication is required.

Request

Method: POST

Header: content-type: application/x-www-form-urlencoded

Header: authorization: Basic base64(client_id:client_secret) (This header can only be used if 'client_id' and 'client_secret' are **NOT** found within the message body and vice versa!)

Endpoint: /auth/oauth/v2/token

Parameters: Parameters: grant_type=client_credentials&client_id=a-client_id&client_secret=a-client_secret&scope=a-list-of-scope-values

Optional: **scope:** Only SCOPE values that have been registered for the client will be granted by the OAuth server

---

Response

Header:status: 200

Header:content-type: application/json

Body: { "access_token":"115b8c ... 11a5", "token_type":"Bearer", "expires_in":3600, "scope":"openid email" }

---

3. grant_type=password

This grant type can only be used with **SAMS System Accounts**, it can't be used for HHS or external users. The client passes userID and password for the system accounts.

Request

Method:	POST
Header:	content-type: application/x-www-form-urlencoded
Header:	authorization: Basic base64(client_id:client_secret) (This header can only be used if 'client_id' and 'client_secret' are **NOT** found within the message body and vice versa!)
Endpoint:	/auth/oauth/v2/token
Parameters:	grant_type=password&username=a-username&password=a-users-password&client_id=a-client_id&client_secret=a-client_secret&scope=a-list-of-scope-values
Optional:	**scope:** Only SCOPE values that have been registered for the client will be granted by the OAuth server

Response

Header:	status: 200
Header:	content-type: application/json
Body:	Example: { "access_token":"115b8c ... 11a5", "token_type":"Bearer", "expires_in":3600, "refresh_token":"74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

---

4. grant_type=urn:ietf:params:oauth:grant-type:jwt-bearer

This grant_type can be used if the client is in possession of an id_token (represented as JWT) of an authenticated user. Only id_token (JWT) that were issued by the OAuth server are accepted.

Request

Method:	POST
Header:	content-type: application/x-www-form-urlencoded
Header:	authorization: Basic base64(client_id:client_secret) (This header can only be used if 'client_id' and 'client_secret' are **NOT** found within the message body and vice versa!)
Endpoint:	/auth/oauth/v2/token

Parameters: grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&assertion=a-jwt&client_id=a-client_id&client_secret=a-client_secret&scope=a-list-of-scope-values

Optional: **scope:** Only SCOPE values that have been registered for the client will be granted by the OAuth server

---

Response

Header:status: 200

Header:content-type: application/json

Body: { "access_token":"115b8c ... 11a5", "token_type":"Bearer", "expires_in":3600, "refresh_token":"74b29d19-8b ... 7bb6bd1", "scope":"openid email" }

---

5. grant_type=refresh_token

This grant_type can be used if the client is in possession of a refresh_token. The request will only be successful if the refresh_token has not expired. The parameter 'SCOPE' can only include the same or a subset of values that were originally requested. The refresh_token can only be use once.

Request

Method: POST

Header: content-type: application/x-www-form-urlencoded

Header: authorization: Basic base64(client_id:client_secret) (This header can only be used if 'client_id' and 'client_secret' are **NOT** found within the message body and vice versa!)

Endpoint: /auth/oauth/v2/token

Parameters: Parameters: grant_type=refresh_token&refresh_token=a-refresh-token&client_id=a-client_id&client_secret=a-client_secret&scope=a-list-of-scope-values

Optional: **scope:** Only SCOPE values that have been registered for the client will be granted by the OAuth server

---

Response

Header:status: 200

Header:content-type: application/json

Body: { "access_token":"115b8c ... 11a5", "token_type":"Bearer", "expires_in":3600, "refresh_token":"74b29d19-8b ... 7bb6bd1", "scope":"openid email" }