འབྲུག་རྒྱལ་འཛུགས་རིག་དང་བརྒྱུད་འཕྲིན་ལས་ཁུངས།
འབྲུག་དང་བརྒྱུད་འཕྲིན་ལྷན་ཁག རྒྱལ་གཞུང་།
Department of Information Technology & Telecom
Ministry of Information & Communications
Royal Government of Bhutan

Data Exchange Standard Operating Protocol

Purpose:

The purpose of this document is to help system developers and integrators in understanding the requirements of the API(Application Programming Interface) standards for sharing data over the Government DataHub platform.

Sl#	Description	Remarks
1	Name of Doc: **Data Exchange SoP**	*The SOP is a living document, so it will be updated as and required.*
2	Version: **1.0**	
3	Last Date modified: **Jan 2022**	

I. Terminology

The keywords of the document "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" should be interpreted as specified by the Internet Engineering Task Force (Internet Engineering Task Force (IETF) RFC 2119: Keywords for use in RFCs to indicate requirements levels). To highlight the relevance of these words, they have been provided in block capitals and their meaning is as follows:

Meaning	Words expressing the meaning
Required/obligatory (absolute requirement or prohibition)	MUST, REQUIRED, SHALL
Recommendation (full implications must be understood and carefully weighed before choosing a different course)	SHOULD, RECOMMENDED

Acceptable/allowed	MAY, OPTIONAL
Not recommended (acceptable only for specific reasons or under specific circumstances)	SHOULD NOT, NOT RECOMMENDED
Prohibited (absolute prohibition)	MUST NOT, SHALL NOT

II. Data Consumer

1. The Data Consumer(s) MUST finalize the **Data element requirement** for their application in consultation with the Data owner(s).
    a. The Data Consumer (s) MUST check whether Data owners have the fields in their database.
    b.
2. The Data Consumer(s) SHOULD check if the **Data Elements required** for their application can be fulfilled by the existing API in the National Data Exchange Platform from APIM at link https://staging-datahub-apim.dit.gov.bt/store from agency network (Gov net) or visit with DITT
3. To use the existing API, the request must be sent to data owners for permission to use their data in the letter template provided in Annexure I.
4. If the data owner does not have a database, the data owner and data consumer can have an agreement that the database will be made available with the required fields and if both party agrees, the DATAHUB team can develop a dummy API with dummy data for seamless integration which will be replaced once database is up. The time for the database to be made available should not take more than 3 months.
5. Once the data consumer receives the API, the data consumer should be responsible to check the working of it and solve any issues before they integrate to have minimum issues during integration.

III. Data Owner

1. The Data Owner MUST consent the usage of Data Element(s) by the Data Consumer(s) as per the consent template Annexure II.

2. Once the Data Owner Consent the Access to Data Consumers but API is not developed, the data owner(s) MUST create the SQL query/Stored Procedure(s) with all the Data element(s) agreed and Database access (Server IP address, DB users with execute rights and DB version) or their API gateway access with endpoints(If their system has API gateway for internal services) and shared National Data Exchange team to Develop new API with it.
3. The data owner MUST make sure that they create a user for National Data Exchange platform with only read access ensure security of Data Owner's DB
4. The data owner(s) MUST NOT change the access credentials (Server IP, Username, Access right) unilaterally.
5. The Data Owner(s) MUST work with National Data Exchange Platform for any changes to their DB (Server IP, Username, Access right and new data elements)

IV. Annexure I: Consent Seeking Template for Data Consumers

Agency letterhead
Letter No……

Address……..[]

Subject : Seeking Consent for Usage Data from your agency System

Sir/Mam,
We would like to seek permission to use _____{Agency Name}______________________ data via National Data Exchange Platform for our _______(system name)_______________ system that will be used in our __________________. (own agency name)

Our system would require the following data :-
1. Name
2. DOB
3. …
4. …....

The above data will be used for __________________________________(state the reason)

Looking forward to hearing from you soon.

Thank you,

(........................)

----Agency Name----------
Cc:

1. DITT for kind information.

V. Annexure II: Data Usage Consenting Template for Data Owner

Agency letterhead
Letter No.

Adddres…

<u>Subject : Data owners Response (If the required data exists in API in NDE)</u>

Sir/Mam,
We would like to consent to the use following data elements for ________(system name) _______ system via National Data Exchange Platform from ------------ DB name/System name------------- for ------------------------ (State reason) only.

The following data can be used by your system :-

1. Name
2. DOB
3. …
4. …...

Thank you,
Sincerely,

(........................)
----Agency Name----------
Cc:

a. DITT for necessary action.

VI. Data owner’s Response (If the required data does not exist as API in NDE)

Agency letterhead
Letter No.

Address…[
To,
The Director,
Department of Information Technology & Telecom
Thimphu
]

Sir/Mam,

The requested data elements by your agency do not exist as API in NDE . We would like to therefore request DITT to develop an API based on the details provided :-

Query :- {SELECT * FROM ….}

DB details :-
IP:-
User:-
Password:-

Thank you,
Sincerely,

(........................)
----Agency Name----------

Cc:

a. DITT for necessary action.

VII. SSO(Single Sign On) Implementation and API consumption Guide:

1. SSO and API Invocation flow (language independent

To integrate client applications with the Datahub platform, we need an unique client id and secret for each and every application. Development team suppose to provide login redirect url and logout redirect url (if it has separate urls for login and logout) to DITT Datahub Team and make a req

uest to register the client application in datahub side.

[https://tools.ietf.org/html/rfc6749#section3.1.2.2]
[https://tools.ietf.org/html/rfc6749#section10.6]

Once DITT Datahub team provided the Client ID and Client secret, developers can start integrating the application with sso platform. Please refer the OIDC SSO Flow diagram sectio n below

NOTE: You can try the sample playground app [https://docs.wso2.com/display/IS530/OAuth+2.0+with+WSO2+Playground] to get some ida about how the SSO and access token generation happens.

Integrating SSO

2. Login Request(authorize request)

[https://tools.ietf.org/html/rfc6749#section4.1]

[https://tools.ietf.org/html/rfc6749#section4.1.1]

Staging URL	https://stgsso.dit.gov.bt/oauth2/authorize
Production URL	https://sso.dit.gov.bt/oauth2/authorize

Once the client application detects that the user is not authorized, it should initiate the authorize request to login with Datahub identity platform.

query parameters	response_type=code&client_id=<client_id>&scope=PRODUCTI O N&redirect_uri=<application_callback_url>
headers	application/xwwwformurlencoded

Application_callback_url : URl which get redirected to after login with the code value

Redirected url will look like bellow

[https://tools.ietf.org/html/rfc6749#section4.1.2]

Sample callback with parameters

```
http://xxx.xxx.xxx.xxx:xxxx/ViewCitizenDetails/sso/acs?code=20258599d43630aebf7bc33
1fdca7cdf
&session_state=fe505a62a3d313db043670185ebde767e2956a96c5e7ba081974f8bdb67884
8c.Uq0tOhKc 3avTdKU2V2_w
```

With the callback after successful login, Client application will receive a token as a request parameter called code. Client application is suppose to read the value of the code , session_sta te parameters and store in client session.

3. Generation access token for SSO users

[https://tools.ietf.org/html/rfc6749#section4.1.3]

NOTE: Following requests should be done in server side (using languages such as Java, php , csharp, nodejs, etc.) and never send the call from browser.

To access the API, we need an access_token. We can use the authorization code, which we got at the time of login in in previous step, to generate an access token. Soon after client application receive the code, Application can make a token request and get an access token.

Endpoints	
Staging URL	https://stgsso.dit.gov.bt/oauth2/token
Production URL	https://sso.dit.gov.bt/oauth2/token

query parameters	grant_type=authorization_code&code=<code retrieved by previo us step> &redirect_uri=<application_callback_url>
headers	application/xwwwformurlencoded TODO issue with header in client code

```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic
SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMT
GxDNmpzbEFDVzhh Content-Type:
```

```
application/x-www-form-urlencoded
grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA&redirect_uri=https%
3A %2F%2Fclient%2Eexample%2Ecom%2Fcb
```

Expecting a response like follows
[https://tools.ietf.org/html/rfc6749#section4.1.4]
[tokenid spec TODO]

```
{
 "access_token": "fad53b4a24f03a52add3b4298379bcdb",
 "refresh_token": "e2e9a402e9ea3a52864eeaa73603233a",
 "scope": "openid",
 "id_token":
"eyJ4NXQiOiJObUptT0dVeE16WmxZak0yWkRSaE5UWmxZVEExWXpkaFpUUmlPV
0
UwTldJMk0ySm1PVGMxWkEiLCJraWQiOiJkMGVjNTE0YTMyYjZmODhjMGFiZDE
y
YTI4NDA2OTliZGQzZGViYTlkIiwiYWxnIjoiUlMyNTYifQ.eyJhdF9oYXNoIjoiZkcyO
EJ
KR0NpcndzZFVwSkhiaXlUUSIsInN1YiI6ImRwZXJlcmEiLCJhdWQiOlsiZlhnWDUybE
V
Rb0lObXlOX1NVUFJVbjBMbUNnYSJdLCJhenAiOiJmWGdYNTJsRVFvSU5teU5fU1
V
QUlVuMExtQ2dhIiwiYXV0aF90aW1lIjoxNTE4MjQxNzg2LCJpc3MiOiJodHRwczpcL1
w
vbG9jYWxob3N0Ojk0NDRcL29hdXRoMlwvdG9rZW4iLCJleHAiOjE1MTgyNDUzOD
csI
mlhdCI6MTUxODI0MTc4N30.GN6gQKLxI3DqVLMR2U53JZHJ12HCFSFcqXPZ2N0
N
NQrXyHViKabWlPpBq32SGWsfxeRxQqsLCrnyeiVmO7QkkW5tz8rpGxnhfBulFSglM0
4
v5wOnbJbVRNzGWWzHGFa71H8O4TwETb_LedJ_cGxPVyADWPnMn2QoazHeww
4",   "token_type": "Bearer",
 "expires_in": 3235
}
```

Information retrieving from above response need to saved in the user session. These information are unique for each user , hence naver save these information in application session.

access_token	Token that client apps need to use for API Calls as a Bearer Authorizati on header.
refresh_token	Token which we used to generate new access token when the curr ent access token is expired.
id_token	There will be three base64 encoded values in this section separated by dot (.). Second portion of the string will contain following data in JSON format. NOTE: You need to decode it first after splitting and parse the JSON

	● username ● http://wso2.org/claims/role (Roles associated with the user as) ● firstName ● middleName ● lastName ● mobileNo ● http://wso2.org/claims/emailaddress (Email address of the user) ● authorizedCID [including children under the legal age(18)] [https://medium.com/@hasiniwitharana/openidconnect532465308090] [http://openid.net/specs/openidconnectcore1_0.html#IDToken]

4. Renewing the access token

Endpoints	
Staging URL	https://stgsso.dit.gov.bt/oauth2/token
Production URL	https://sso.dit.gov.bt/oauth2/token

query parameters	grant_type=refresh_token&refresh_token=<refresh token>
headers	ContentType: application/xwwwformurlencoded Authorization: Basic <Base64encodedclient_key:client_secret>

For more details : https://docs.wso2.com/display/AM210/Refresh+Token+Grant

5. Check session

[https://docs.wso2.com/display/IS540/OpenID+Connect+Logout+URL+Redirection]

Client application should use the following URL to logout the user from the SSO session. Once user clicks the URL, user will be redirected to the SSO portal and asking for logout confirmation. Once confirmed, user will be logout from the SSO portal side and redirect back to the application to handle the logout in the application side.

Endpoints	
Staging URL	https://stgsso.dit.gov.bt/oidc/logout
Production URL	https://sso.dit.gov.bt/oidc/logout

query parameters	post_logout_redirect_uri=<logoutCallbackURL>&id_token _hint=< ID Token>
headers	ContentType: application/xwwwformurlencoded Authorization: Basic <Base64encodedclient_key:client_secret>

6. Generating token for internal users (Application to Application)

[https://docs.wso2.com/display/AM210/Client+Credentials+Grant]

[https://tools.ietf.org/html/rfc6749#section4.4]

We use client credential grant type to generate access tokens for internal users. The token we use in this case will be shared within the client application (All the internal users will be using the same access token at a any given time). This token need to be saved in the application session in client application.

Endpoints	
Staging URL	https://stgsso.dit.gov.bt/oauth2/token
Production URL	https://sso.dit.gov.bt/oauth2/token

query parameters	grant_type=client_credentials
headers	ContentType: application/xwwwformurlencoded Authorization: Basic <Base64encodedclient_key:client_secret>

7. Access API Using Access token.

Use the correct access token to access the APIs which are subscribed to the token generated Application.

query parameters	As required by the API
headers	Accept: application/json Authorization: Bearer {access_token}

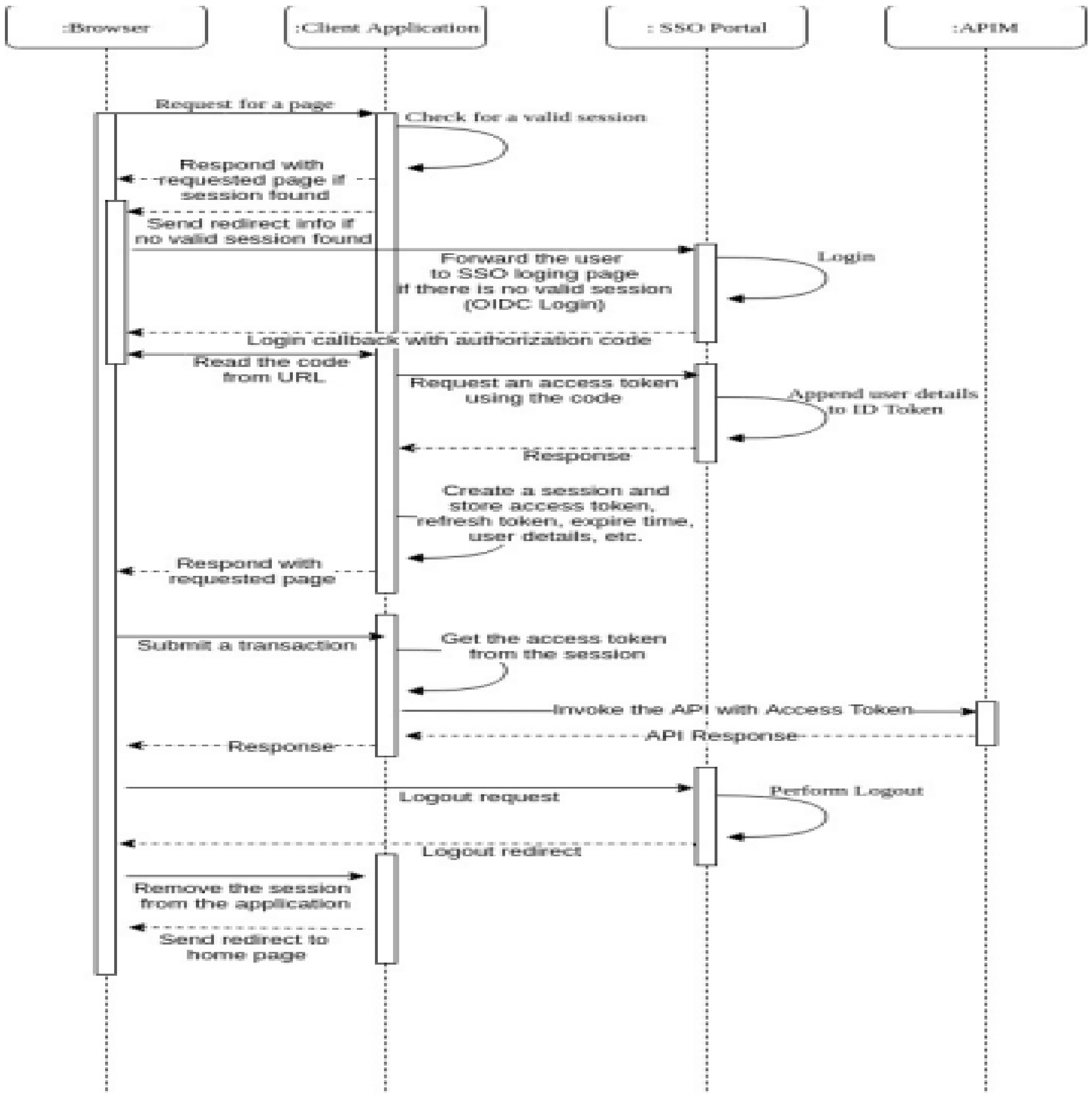
Endpoints	
Staging URL	https://stgapi.dit.gov.bt/ <api_context>/<api_version>/<apiresource>
Production URL	https://api.dit.gov.bt/ <api_context>/<api_version>/<apiresource>

* If you are using Swagger generate SDK client, the api resource will be wrapped by the sdk client and exposure as methods.

Ex: of using JAVA SDK client

```
                  OkHttpClient client = new OkHttpClient();
                client.setReadTimeout(15, TimeUnit.SECONDS);
                client.setConnectTimeout(15, TimeUnit.SECONDS);

                ApiClient apiClient = new ApiClient();
                apiClient.setHttpClient(client);
                apiClient.setBasePath("https://localhost:8245/nlcs_landdetailapi/1.0.0");
                apiClient.setAccessToken("<access token>");
                apiClient.setVerifyingSsl(false);

                DefaultApi api = new DefaultApi(apiClient);

                 LandDetailResponse detailResponse =
                 api.landdetailsbycidCidGet("11705002040", null);

List<LanddetailObj> landdetailObjs = detailResponse.getLandDetails().getLandDetail();

                if (!landdetailObjs.isEmpty()) {

                        for (LanddetailObj landdetailObj : landdetailObjs) {
                         System.out.print("DzongkhagOrThromde :" +
landdetailObj.getDzongkhagOrThromde());
                 System.out.print(", GewogOrThromdeVillage :" +
landdetailObj.getGewogOrThromdeVillage());
        System.out.print(", LandLocationFlag :" + landdetailObj.getLandLocationFlag());
       System.out.print(", LandTypeOrPrecinct :" + landdetailObj.getLandTypeOrPrecinct());
           System.out.print(", OwnerCid :" + landdetailObj.getOwnerCid());
          System.out.print(", OwnerName :" + landdetailObj.getOwnerName());
         System.out.print(", OwnershipType :" + landdetailObj.getOwnershipType());
          System.out.print(", PlotAreaUnit :" + landdetailObj.getPlotAreaUnit());
             System.out.print(", PlotId :" + landdetailObj.getPlotId());
         System.out.print(", ThramNumber :" + landdetailObj.getThramNumber());
          System.out.print(", PlotNetArea :" + landdetailObj.getPlotNetArea());

           System.out.print(", LapNam :" + landdetailObj.getLapName());
         System.out.print(", PlotNetArea :" + landdetailObj.getDemkhongName());
                        System.out.println("");
          System.out.println("########################################");
                        }

                } else {
                        System.out.println("No land detail Data");
  System.out.println("########################################");
                }
```

8. OIDC SSO Flow diagram


:Browser
:Client Application
: SSO Portal
:APIM
Request for a page
Check for a valid session
Respond with requested page if session found
Send redirect info if no valid session found
Forward the user to SSO loging page if there is no valid session (OIDC Login)
Login
Login callback with authorization code
Read the code from URL
Request an access token using the code
Append user details to ID Token
Response
Create a session and store access token, refresh token, expire time, user details, etc.
Respond with requested page
Submit a transaction
Get the access token from the session
Invoke the API with Access Token
API Response
Response
Logout request
Perform Logout
Logout redirect
Remove the session from the application
Send redirect to home page