

PROJECT NAME

CITIZEN DIGITAL ACCOUNT PLATFORM (CDAP)

DOCUMENT NAME

INTEGRATION STANDARD OF CDAP





Dated: 06 January, 2022

Version: 4.0

Powered By



TABLE OF CONTENTS

Platform Integration Process	4
Step 1: Registration	5
Step 2: Login	5
Step 3: Account Setting	6
Step 4: Create Application	7
Step 4: Follow Integration Procedures	8
1. Web Integration:	8
 PHP	9
 .Net (JsonWebToken)	10
 .Net (System.IdentityModel.Tokens.Jwt)	11
 Java	12
2. Widget Integration:	13
3. API Integration	14
Token Create	14
Token Check	14
User Check	14
User Login	15
User Create	16
User Update	17
Send OTP	18
User Password Reset	18

LIST OF FIGURES

Figure 1: Platform Integration Process in CDAP	4
Figure 2: Go to https://idp-client.live.mygov.bd/registration	5
Figure 3: Login to https://idp-client.live.mygov.bd/login	5
Figure 4: Update account at https://idp-client.live.mygov.bd/account	6
Figure 5: Go to https://idp-client.live.mygov.bd/applications	7
Figure 6: Fill in Application Details (https://idp-client.live.mygov.bd/applications/create)	7
Figure 7: Web Integration Steps.....	8
Figure 8: Widget Integration Steps	13

PLATFORM INTEGRATION PROCESS

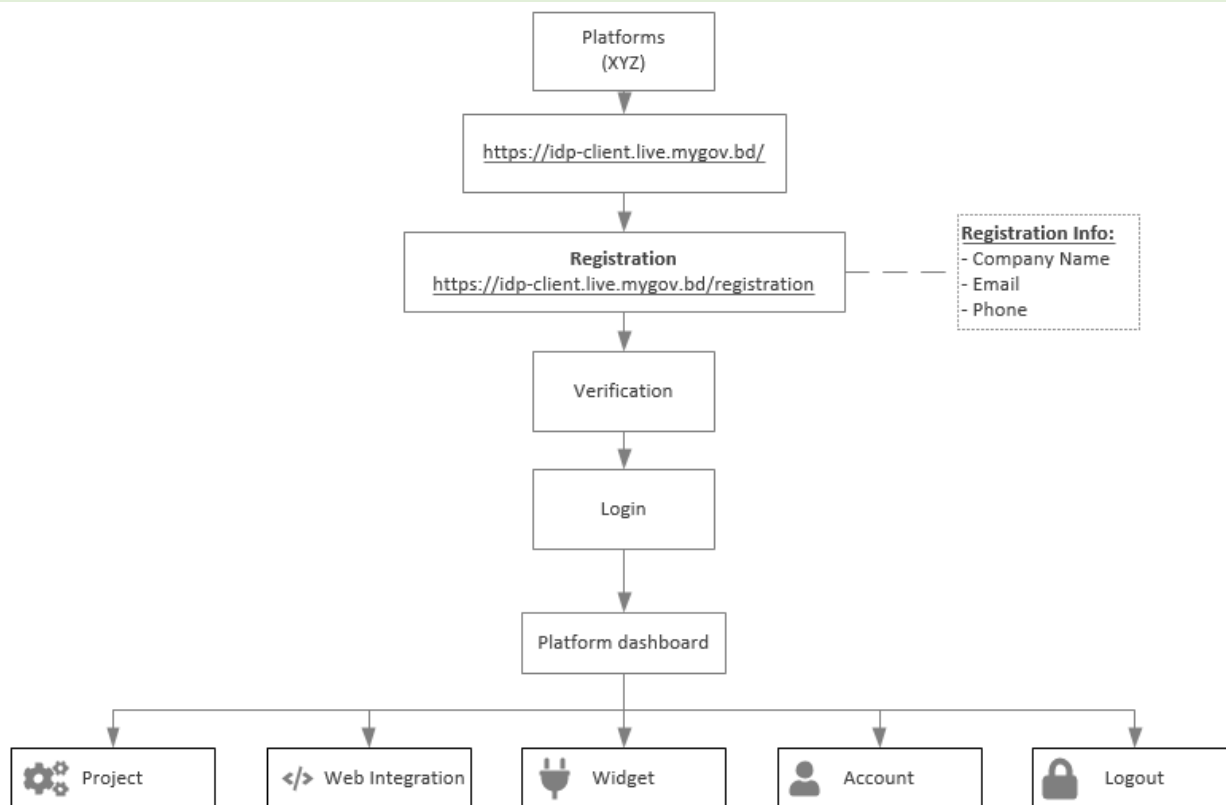
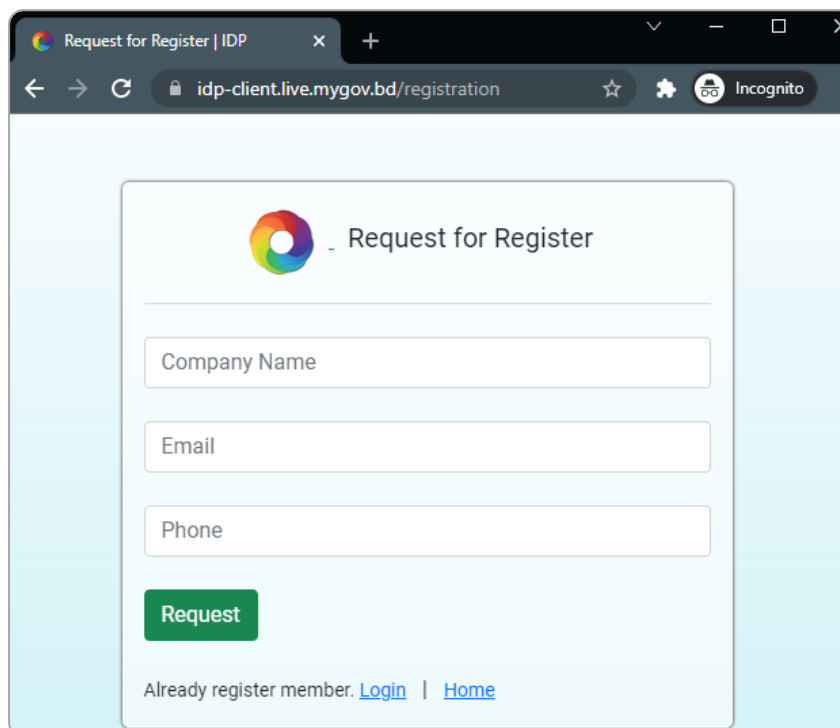


Figure 1: Platform Integration Process in CDAP

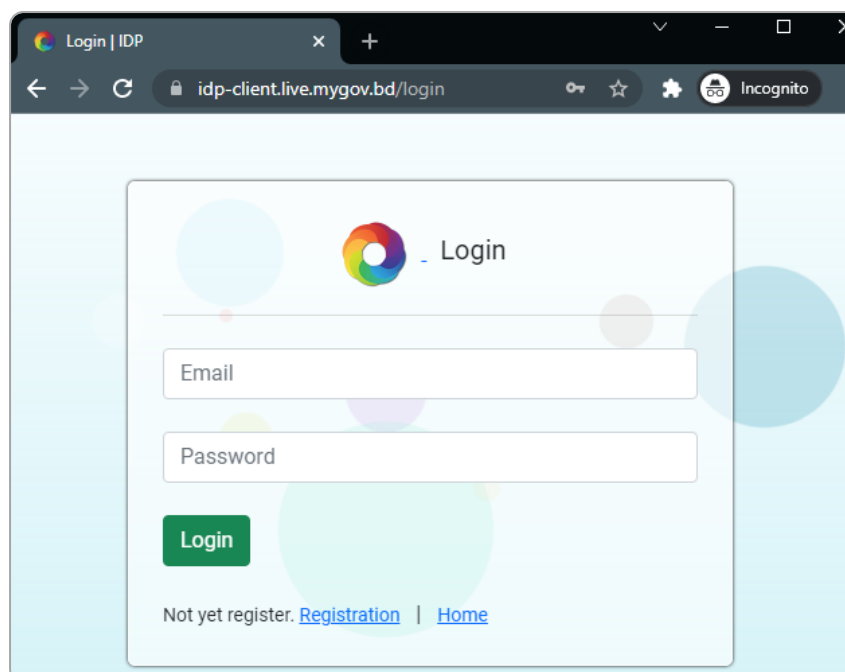
STEP 1: REGISTRATION



The screenshot shows a web browser window with the title "Request for Register | IDP". The address bar displays "idp-client.live.mygov.bd/registration". The page features a light blue background with a central white box containing the "Request for Register" form. The form includes a logo, a title, and three input fields: "Company Name", "Email", and "Phone". Below these fields is a green "Request" button. At the bottom of the form, there is a link for existing members: "Already register member. [Login](#) | [Home](#)".

Figure 2: Go to <https://idp-client.live.mygov.bd/registration>

STEP 2: LOGIN



The screenshot shows a web browser window with the title "Login | IDP". The address bar displays "idp-client.live.mygov.bd/login". The page features a light blue background with a central white box containing the "Login" form. The form includes a logo, a title, and two input fields: "Email" and "Password". Below these fields is a green "Login" button. At the bottom of the form, there is a link for new users: "Not yet register. [Registration](#) | [Home](#)".

Figure 3: Login to <https://idp-client.live.mygov.bd/login>

STEP 3: ACCOUNT SETTING

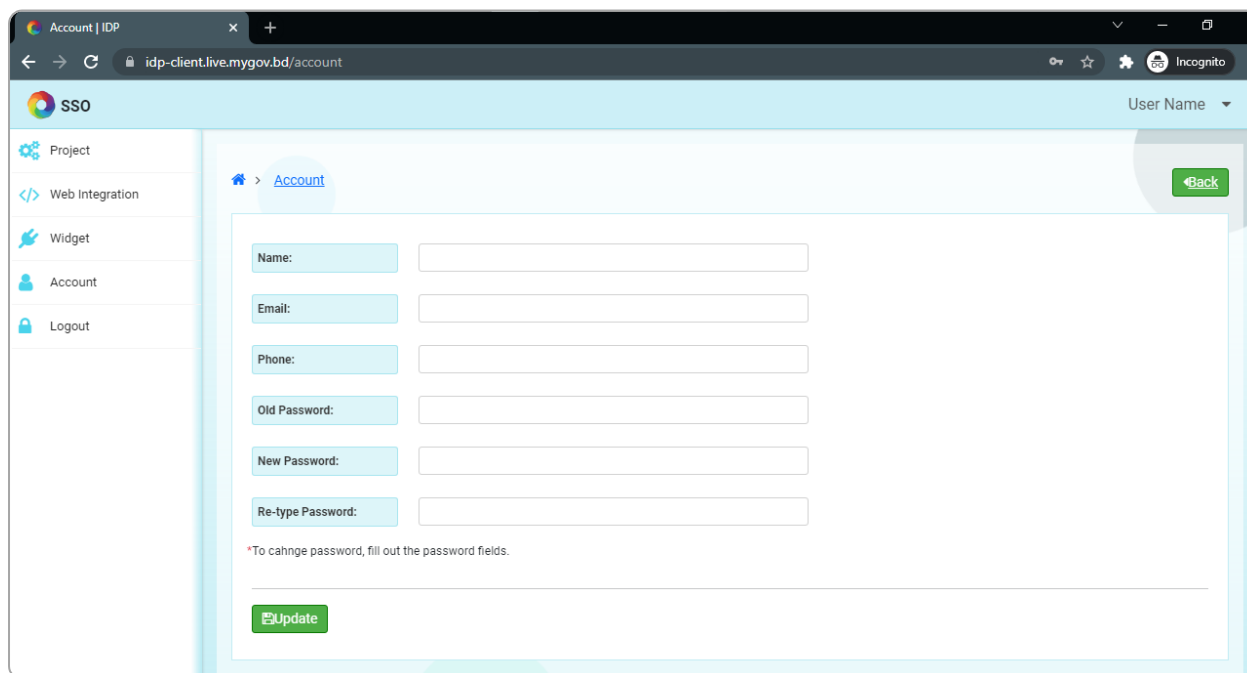


Figure 4: Update account at <https://idp-client.live.mygov.bd/account>

Account Details:

- Name
- Email
- Phone
- Old Password
- New Password
- Re-type Password

STEP 4: CREATE APPLICATION

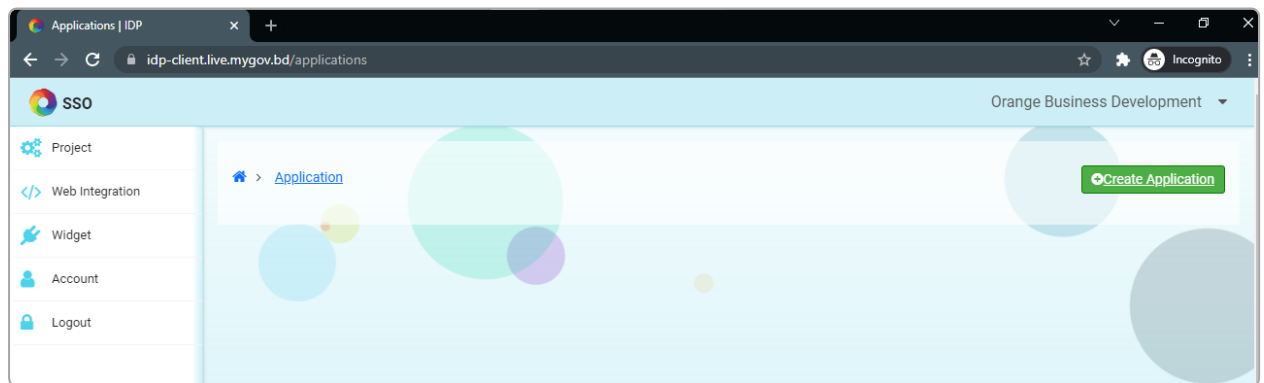


Figure 5: Go to <https://idp-client.live.mygov.bd/applications>

Under **Project** Tab Click **Create Application**.

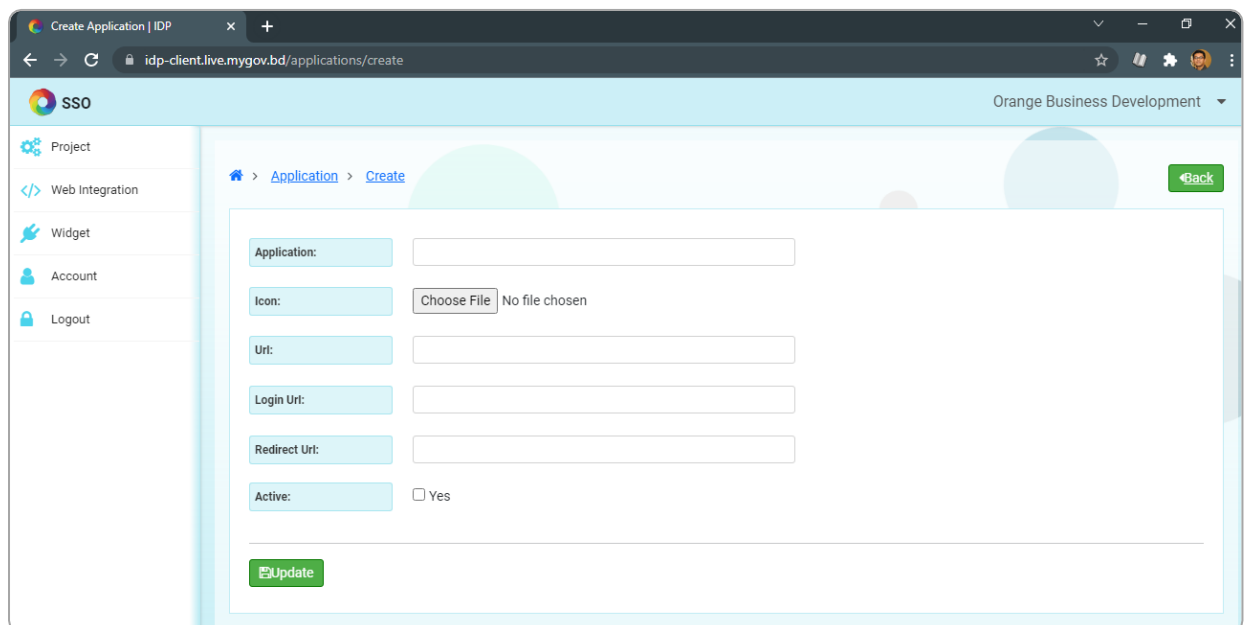


Figure 6: Fill in Application Details (<https://idp-client.live.mygov.bd/applications/create>)

Application Details:

1. Application Name
2. Icon (150 x 150)
3. URL
4. Login URL
5. Redirect URL
6. Active (Yes/No)

After Creating Application, by pressing **Update** button, the platform will receive their:

- Client ID
- Secret Key
- API Key

These can be Re-generated.

STEP 4: FOLLOW INTEGRATION PROCEDURES

1. WEB INTEGRATION:

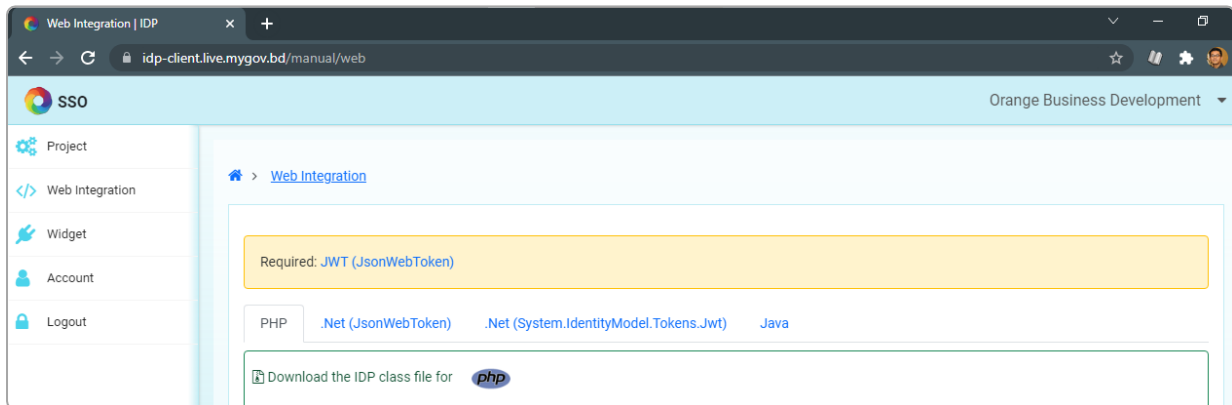


Figure 7: Web Integration Steps

Required: JWT (JsonWebToken)

Integration Module:

- Login
- Logout
- Data

Language Support:



PHP



.Net

(JsonWebToken)

(System.IdentityModel.Tokens.Jwt)



Java

[PHP IDP class file](#)

[.Net JsonWebToken IDP Class File](#)
[.Net System.IdentityModel.Tokens.Jwt IDP Class File](#)

[Java IDP Class File](#)



PHP

[PHP IDP class file](#)

Install JWT via composer: composer require firebase/php-jwt or [Click Here](#)

Login

```
require_once 'Idp/autoload.php';
$client = new \IDP_Client();
$client->setClientId('xxxxxxxxxxxxxxxx');
$client->setClientSecret('xxxxxxxxxxxxxxxx.xxxxxxxxxx');

$client->setRedirectUri('app_redirect_url');
$authUrl = $client->loginRequest();
header("Location: ".$authUrl);
```

Logout

```
$client->setRedirectUri('app_redirect_url');
$authUrl = $client->logoutRequest();
header("Location: ".$authUrl);
```

Data

```
$data = $client->responseRequest($_POST);
echo 'Log in as : '.$data->name;
```



.NET (*JSONWEBTOKEN*)
[.Net JsonWebToken IDP Class File](#)

*Install JWT via .Net CLI: dotnet add package JsonWebTokens --version 4.0.0 or
get [click here](#) for more information.*

Login

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
var uri = idp.loginRequest();
Response.Redirect(uri);
```

Logout

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
var uri = idp.logoutRequest();
Response.Redirect(uri);
```

Data

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
Object data = idp.responseRequest(Request.Form["key"], Request.Form["token"]);
/*Parse data as json (using Newtonsoft.Json;)/
dynamic json = JsonConvert.DeserializeObject(data.ToString());
string userId = json["id"];
```

**.NET** (SYSTEM.IDENTITYMODEL.TOKENS.JWT)[.Net System.IdentityModel.Tokens.Jwt IDP Class File](#)

Install JWT via .Net CLI: `dotnet add package System.IdentityModel.Tokens.Jwt --version 6.11.1` or get [click here](#) for more information.

Login

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
var uri = idp.loginRequest();
Response.Redirect(uri);
```

Logout

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
var uri = idp.logoutRequest();
Response.Redirect(uri);
```

Data

```
using IdpMygov;
IDP_Client idp = new IDP_Client();
idp.setClientId("xxxxxxxxxxxxxxxx");
idp.setClientSecret("xxxxxxxxxxxxxxxx.xxxxxxxxxx");
idp.setRedirectUri('app_redirect_url');
String data = idp.responseRequest(Request.Form["key"], Request.Form["token"]);
data = data.Replace("{\"typ\":\"JWT\",\"alg\":\"HS256\"}.", "");
/*Parse data as json (using Newtonsoft.Json)*/
dynamic json = JsonConvert.DeserializeObject(data.ToString());
string userId = json["id"];
```

[Java IDP Class File](#)

Login

```
IDP_Client idp = new IDP_Client();
idp.setClientId('xxxxxxxxxxxxxxxx');
idp.setClientSecret('xxxxxxxxxxxxxxxx.xxxxxxxxxx');
idp.setRedirectUri('app_redirect_url');
String url = idp.loginRequest();
redirect(url);
```

Logout

```
IDP_Client idp = new IDP_Client();
idp.setClientId('xxxxxxxxxxxxxxxx');
idp.setClientSecret('xxxxxxxxxxxxxxxx.xxxxxxxxxx');
idp.setRedirectUri('app_redirect_url');
String url = idp.logoutRequest();
redirect(url);
```

Data

```
IDP_Client idp = new IDP_Client();
idp.setClientId('xxxxxxxxxxxxxxxx');
idp.setClientSecret('xxxxxxxxxxxxxxxx.xxxxxxxxxx');
String data = idp.responseRequest(request);
```

2. WIDGET INTEGRATION:

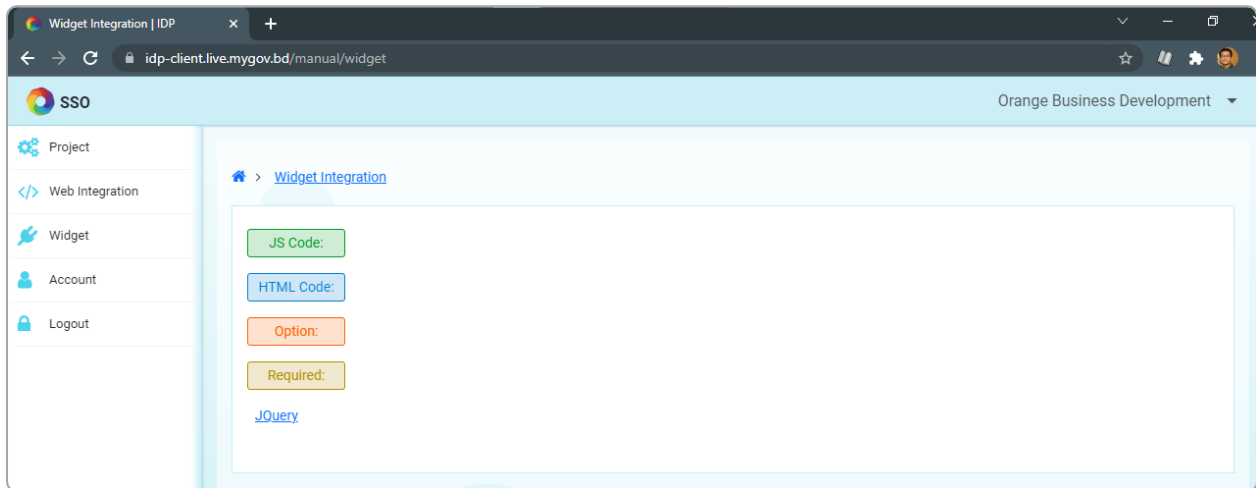


Figure 8: Widget Integration Steps

Required:  **jQuery**
write less, do more.

JS Code:

```
<script src="https://idp-client.live.mygov.bd/js/widget.js?v=1.0.2">
</script>
$('#sso-widget').ssoWidget({});
```

HTML Code:

```
<div id="sso-widget"></div>
```

Option:

Customized the widget icon

```
icon-url="url to widget icon"
align:"left"
```

3. API INTEGRATION

TOKEN CREATE

Url: https://idp-api.live.mygov.bd/token/create

Method: POST

Parameter:

- client_id = xxxxxxxxxx
- api_key = xxxxxxxxxx.xxxxxxxx

Output:

```
{
  "status": "success",
  "token": "1f8959e503f7ef1c0f86681188feb828",
  "expire_time": 1592717538
}
```

TOKEN CHECK

Url: https://idp-api.live.mygov.bd/token/check

Method: POST

Parameter:

- token = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Output:

```
{
  "status": "success",
  "token": "1f8959e503f7ef1c0f86681188feb828",
  "valid": true,
  "expire_time": 1592717538
}
```

USER CHECK

Url: https://idp-api.live.mygov.bd/token/check

Method: POST

Parameter:

- token = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- mobile = 01xxxxxxxx

* Note: email, mobile, NID, BRN or passport

Output:

```
{
  "status": "success",
  "user_exists": "true",
  "verify_by": "mobile"
}
```

or

```
{
  "status": "success",
  "user_exists": "false"
}
```

USER LOGIN

Url:

<https://idp-api.live.mygov.bd/user/login>

Method:

POST

Parameter:

- token = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- password = xxxxxxxx
- mobile = 01xxxxxxxx

* Note: email, mobile, NID, BRN or passport

Output:

```
{
  "status": "success",
  "data": {
    "id": "5e92a03a46804",
    "email": null,
    "mobile": "01xxxxxxxx",
    "nid": "xxxxxxxx",
    "brn": null,
    "passport": null,
    "user_group": null,
    "name": "",
    "name_en": "",
    "mother_name": "",
    "mother_name_en": "",
    "father_name": "",
    "father_name_en": "",
    "spouse_name": "",
    "spouse_name_en": "",
    "gender": "male",
    "date_of_birth": "YYYY-MM-DD",
    "photo": ""
  }
}
```

USER CREATE

Url: https://idp-api.live.mygov.bd/user/create

Method: POST

Parameter:

- token = xx
- password = xxxxxxxx
- mobile = 01xxxxxxxxx
- email = 01xxxxxxxxx
- name = xx xxxxx xxxxx

* Note: mobile, name is required

* Note: email is optional

* Note: name, name_en, mother_name, mother_name_en, father_name, father_name_en, spouse_name, spouse_name_en, gender (male or female), date_of_birth (yyyy-mm-dd), photo is optional

* Note: photo as file

Output:

```
{
  "status": "success",
  "message": "User create successfully.",
  "data": {
    "id": "5e92a03a46804",
    "email": null,
    "mobile": "01xxxxxxxxx",
    "nid": "xxxxxxxxxx",
    "brn": null,
    "passport": null,
    "user_group": null,
    "name": "",
    "name_en": "",
    "mother_name": "",
    "mother_name_en": "",
    "father_name": "",
    "father_name_en": "",
    "spouse_name": "",
    "spouse_name_en": "",
    "gender": "male",
    "date_of_birth": "YYYY-MM-DD",
    "photo": ""
  }
}
```


USER UPDATE

Url: https://idp-api.live.mygov.bd/user/update

Method: POST

Parameter:

- token = xx
- user_id = 5e92a03a46804
- name = xx xxxxx xxxxx

* Note: user_id is required

* Note: current_password, new_password is optional

* Note: name, name_en, mother_name, mother_name_en, father_name, father_name_en, spouse_name, spouse_name_en, gender (male or female), date_of_birth (yyyy-mm-dd), photo is optional

* Note: photo as file

Output:

```
{
  "status": "success",
  "message": "User update successfully.",
  "data": {
    "id": "5e92a03a46804",
    "email": null,
    "mobile": "01xxxxxxxxx",
    "nid": "xxxxxxxxxx",
    "brn": null,
    "passport": null,
    "user_group": null,
    "name": "",
    "name_en": "",
    "mother_name": "",
    "mother_name_en": "",
    "father_name": "",
    "father_name_en": "",
    "spouse_name": "",
    "spouse_name_en": "",
    "gender": "male",
    "date_of_birth": "YYYY-MM-DD",
    "photo": ""
  }
}
```

SEND OTP

Url:

<https://idp-api.live.mygov.bd/user/sendOtp>

Method:

POST

Parameter:

- token = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- mobile = 01xxxxxxxx or example@mail.com
- otp = xxxx

* Note: Mobile or email and OTP is required

Output:

```
{
  "status": "success",
  "message": "OTP Send."
}
```

USER PASSWORD RESET

Url:

<https://idp-api.live.mygov.bd/user/resetPassword>

Method:

POST

Parameter:

- token = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- mobile = 01xxxxxxxx or mymail@mail.com
- password = xxxxxx

* Note: Mobile or email and Password is required

* Note: Password minimum 8

Output:

```
{
  "status": "success",
  "message": "OTP Send."
}
```

Thank You