JJM - Web API

Help Guide | Version 1.0

सत्यमेव जयते

Ministry of Jal Shakti,
Department of Drinking Water & Sanitation

Table of Contents

1. Introduction

The Jal Jeevan Mission is based on a community approach to water and demands extensive Information, Education, and communication as a crucial part of the mission.

Therefore, every little contribution is appreciated towards its successful accomplishment by millions. Also, it will be remembered for many decades when the last-mile rural family of the country will be able to get drinking water in their own house.

1.1. Why do we have this Web-API?

To speed up the implementation of the Jal Jeevan Mission across the country, relevant data should be able to exchange between its various stakeholders simultaneously across the country.

For this purpose, technical team of Jal Jeevan Mission has provided a robust Web-API solution. It enables a registered user to use it for ‘data communication’ effectively according to his requirement.

Now statistical data of Jal Jeevan Mission can be used in real-time at the pan-India level using the Internet, while its consistency and availability are not compromised.

1. Workflow to Web API

This guide is intended for fellow developers who have requirement to include API in their web applications. To get an API, a developer is supposed to follow the given workflow:

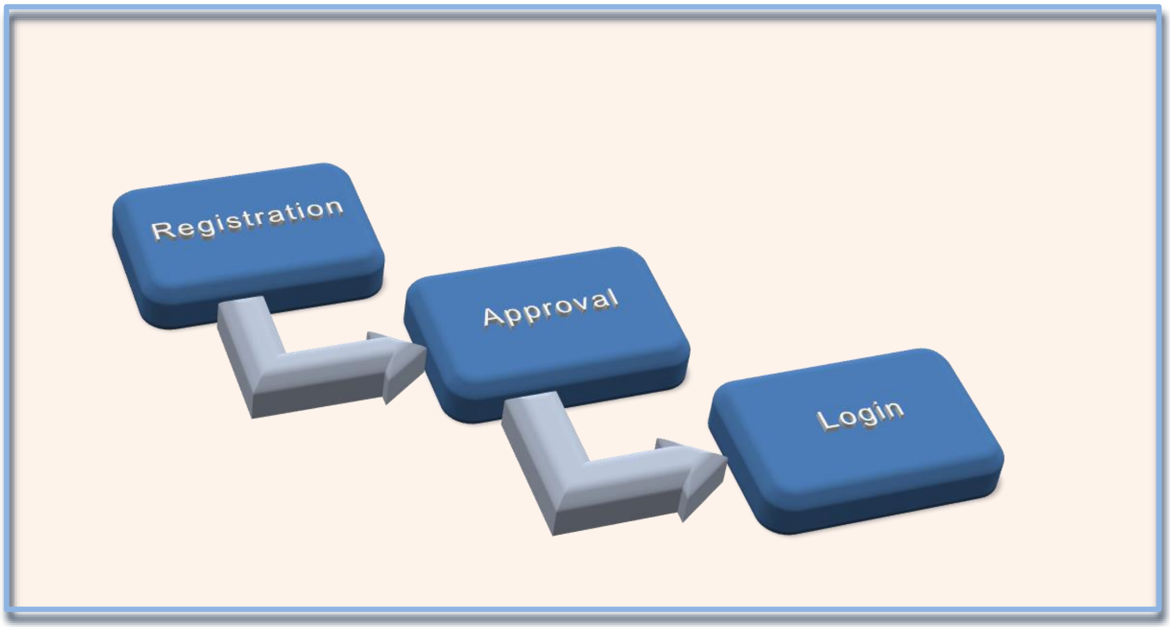


Figure 1: Work flow to get access to API(s)

Figure 1 depicts a high-level summary of the entire process of getting access to API from the website.

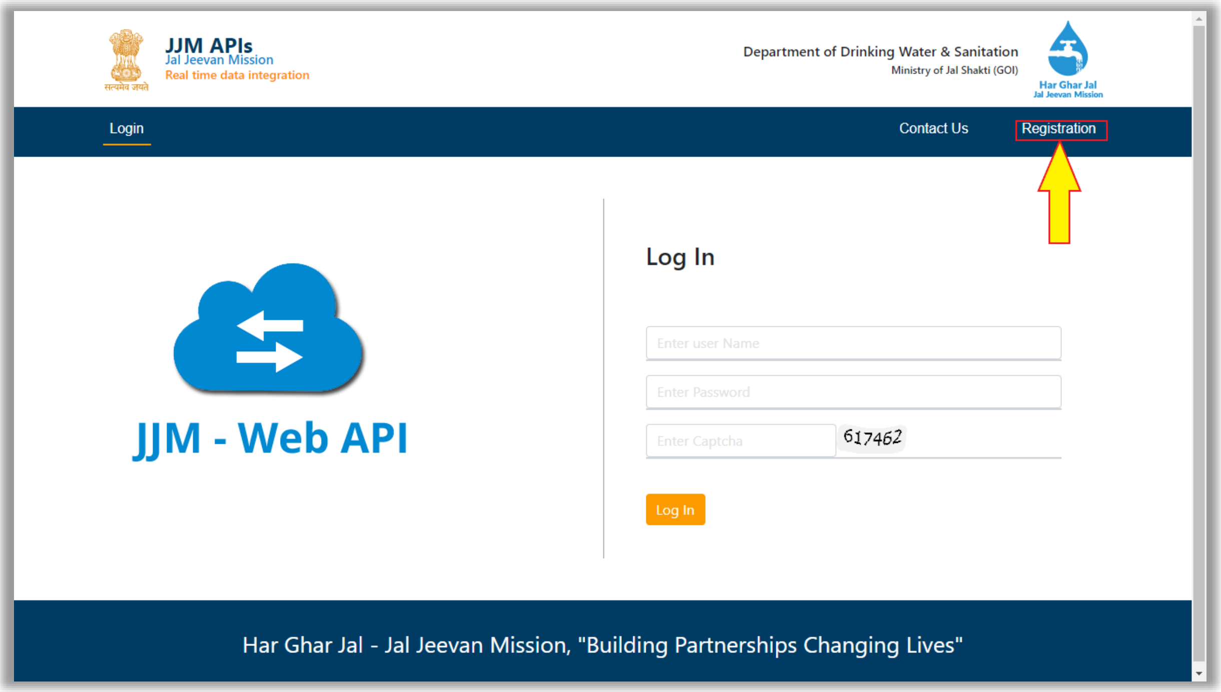
In the next section, we will undergo this process step-by-step.

The very first step to access or to get APIs one should visit the following website:

https://ejalshakti.gov.in/webapi

Note:

1. *This webpage looks similar to the following figure 2 as of now, however it may change in future as per the requirements.*



JJM APIs
Jal Jeevan Mission
Real time data integration

Department of Drinking Water & Sanitation
Ministry of Jal Shakti (GOI)

JJM - Web API

Log In

Enter user Name

Enter Password

Enter Captcha 617462

Har Ghar Jal - Jal Jeevan Mission, "Building Partnerships Changing Lives"

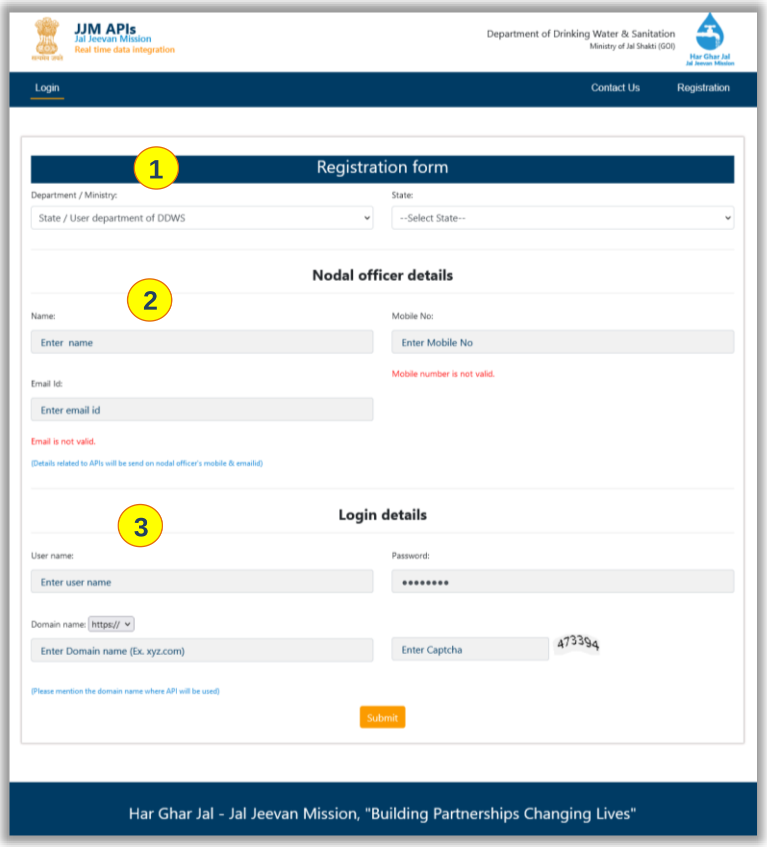
Figure 2 – Home page for reference purpose.

If a user is visiting this website for the first time, then he should go to Registration process, first. Else, he can proceed to Log In.

The registration process is explained in the next part. Let’s click on the Registration button. *(Refer to figure 2, please)*

(P.T.O.)

1.1.Registration Process


JJM APIs
Jal Jeevan Mission
Real time data integration
Department of Drinking Water & Sanitation
Ministry of Jal Shakti (GOI)
Har Ghar Jal
Jal Jeevan Mission
Login
Contact Us
Registration
1
2
3
Submit

Registration form

Department / Ministry: State / User department of DDWS

State: --Select State--

Nodal officer details

Name: Enter name

Mobile No: Enter Mobile No

Mobile number is not valid.

Email Id: Enter email id

Email is not valid.

(Details related to APIs will be send on nodal officer's mobile & emailid)

Login details

User name: Enter user name

Password: ••••••••

Domain name: https://

Enter Domain name (Ex. xyz.com)

Enter Captcha 473394

(Please mention the domain name where API will be used)

Har Ghar Jal - Jal Jeevan Mission, "Building Partnerships Changing Lives"

Figure 3 – Registration Process (web page)

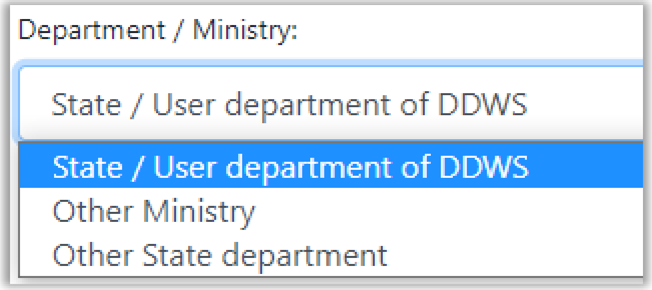
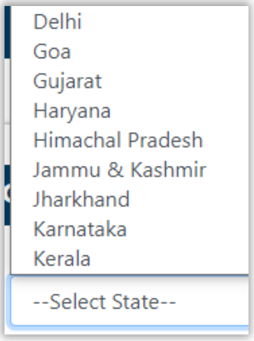
It starts with a click on a button named ‘Registration’ as shown in the figure 2. *(Refer to the figure 2, please.)*

A page similar to the figure 3 is rendered in the web browser. *(Refer to figure 3, please.)*

In the following table description of the indicators added are available. Please go through it for an understanding of it.

(P.T.O.)

Indicator	Web page label	Purpose

1	Department/ Ministry	 Department / Ministry: State / User department of DDWS State / User department of DDWS Other Ministry Other State department Figure 4 A user is supposed to select the appropriate Department or Ministry. *(Refer to the screenshot (figure4) given above.)*
	State	 Delhi Goa Gujarat Haryana Himachal Pradesh Jammu & Kashmir Jharkhand Karnataka Kerala --Select State-- Figure 5 A user is supposed to select name of his state. *(Refer to the screenshot given above.)*
Provide Nodal Officer Details		
2	Enter Name	Please fill the name of Nodal officer. Note: 1. *The chosen name can be in between 3 to 20 characters.* 2. *It must contain alphabets only.*
	Mobile No.	Enter a valid mobile number. (Details of API will be available on Mobile number including email id.) Note: 1. *A valid number should contain 10 digits number without country code.*
	Email ID	Enter a valid Email ID. Note:

		An invalid or inactive Email ID may result in failure to get API(s) credentials.
	Please furnish Login details	
3	Username	Please, choose a username and ensure the following guidelines while creating one. Note: 1. *The chosen name can be in between 3 to 20 characters.* 2. *It must contain alphabets only.*
	Password	Create the password. Note: 1. *The chosen password can be in between 8 to 20 characters.* 2. *It may be alphanumeric, though it must contain minimum one uppercase letter, one lower case letter, and one special character too.*
	Domain name	Select one of the protocols between *https* and *http*. Then enter the name of website in full where it is supposed to use it.
	Enter CAPTCHA	Enter the same figure as written in CAPTCHA code.
4	Submit	After providing the valid details, please click on the Submit button.

Table 1: Registration Process Explained

1.2. Approval process

At this stage, details are getting validated by a competent authority. In this process, the following steps are involved:

1.2.1. Request generation

After submitting the details, an email is sent to the competent authority (on behalf of the applicant.) The applicant should wait for the response. It usually takes 24 - 48 hours.

1.2.2. Review of Details

In this stage, the competent authority verifies the details provided by the applicant.

1.2.3. Approval

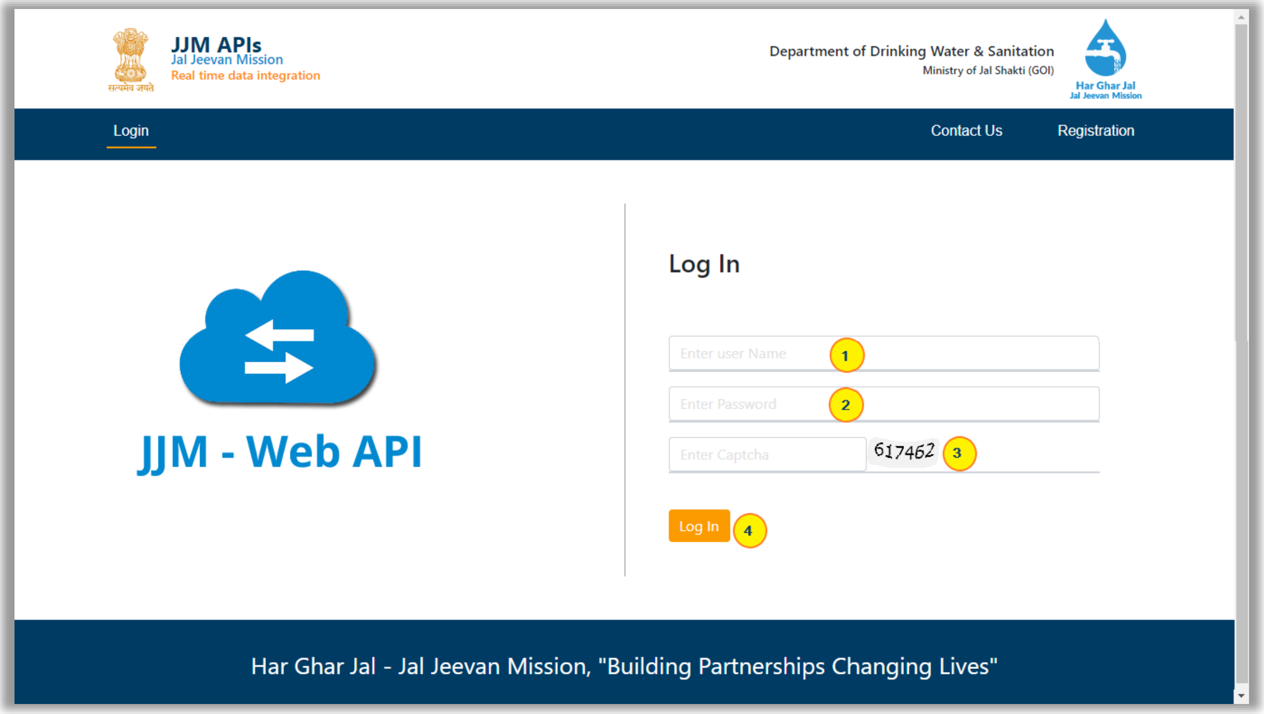
After successful verification of the details, a concerned authority approves it. Approval/ Disapproval will be sent to the applicant via email and on a mobile number.

1.3.Login Process

Once the user received login credentials on his email, he can login and access required Web-API.

Demonstration of Log-in process:

Go to the login page (https://ejalshakti.gov.in/webapi/Home/Login). It looks similar to the screenshot presented below. *(Refer to Table 2, followed by figure 6, please.)*



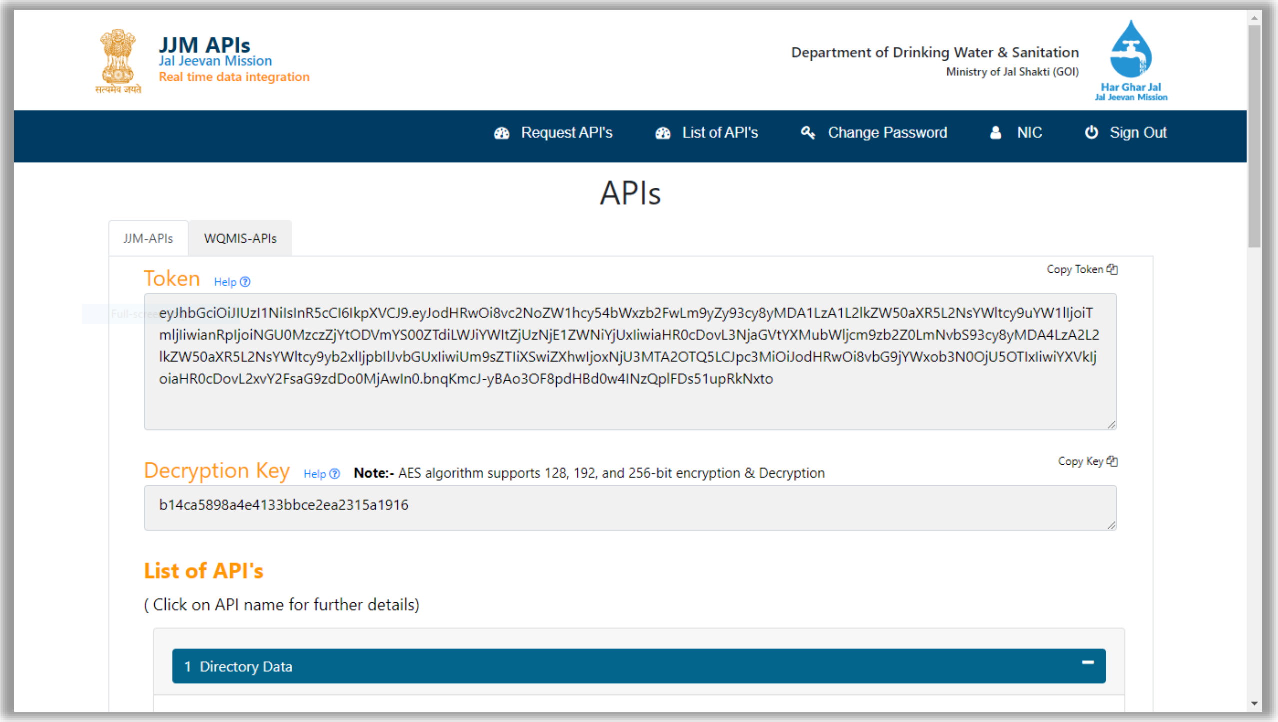
Har Ghar Jal - Jal Jeevan Mission, "Building Partnerships Changing Lives"

Figure 6 – Login page

Indicator	Web page label	Purpose
1	Enter Username	A user should fill the username exactly as provided in the email.
2	Enter Password	A user should fill the password. He should fill it same as provided in email.

3	Enter CAPTCHA	Enter the numbers as shown in the image. Even a slight mismatch would result in reentering of CAPTCHA code.
4	Log In	Click on the button to proceed to web API page. *(Refer to figure, please. It depicts the web page after successful log in.)*

Table 2: Login Process Explained



APIs

Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiT
mJjIiwianRpIjoiNGU0MzczZjYtODVmYS00ZTdiLWJiYWItZjUzNjE1ZWNiYjUxIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L2
lkZW50aXR5L2NsYWltcy9yb2xlIjpbIlJvbGUxIiwiUm9sZTIiXSwiZXhwIjoxNjU3MTA2OTQ5LCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjU5OTIxIiwiYXVkIj
oiaHR0cDovL2xvY2FsaG9zdDo0MjAwIn0.bnqKmcJ-yBAo3OF8pdHBd0w4INzQplFDs51upRkNxto
```

Decryption Key **Note:-** AES algorithm supports 128, 192, and 256-bit encryption & Decryption

```
b14ca5898a4e4133bbce2ea2315a1916
```

List of API's

(Click on API name for further details)

1 Directory Data

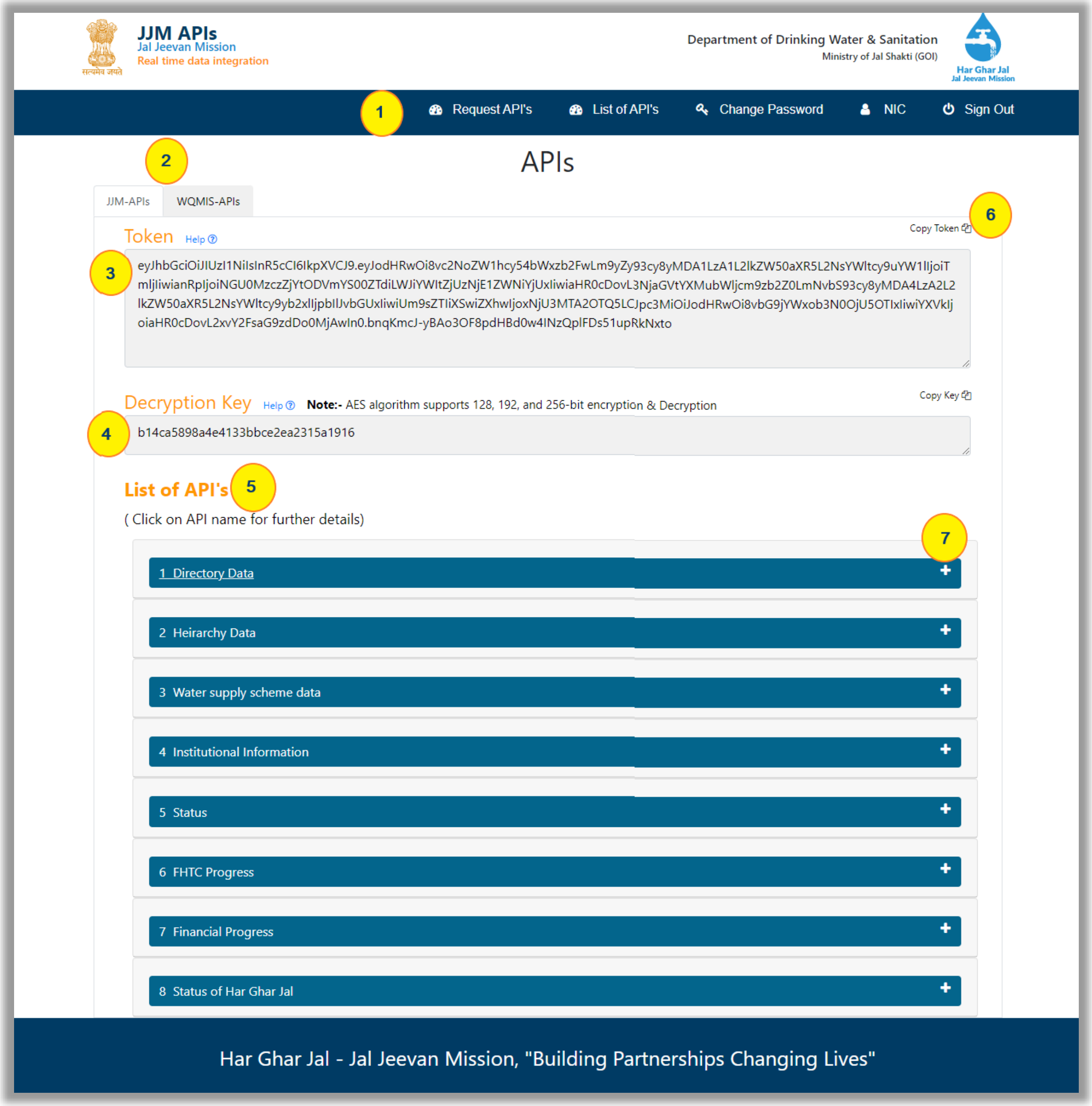
Figure 7

In the next chapter, we will walk through Home page, menu bar and other features (& their functionality) available for a user. Let's move on to it.

3. Working with web API (website)

Let's start with the home page, this page has many helpful features to work with an API. Please read this chapter carefully before start using any API.

3.1.Home Page



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiTmljIiwianRpIjoiNGU0MzczZjYtODVmYS00ZTdiLWJiYWItZjUzNjE1ZWNiYjUxIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L2lkZW50aXR5L2NsYWltcy9yb2xlIjpbIlJvbGUxIiwiUm9sZTIiXSwiZXhwIjoxNjU3MTA2OTQ5LCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjU5OTIxIiwiYXVkIjoiaHR0cDovL2xvY2FsaG9zdDo0MjAwIn0.bnqKmcJ-yBAo3OF8pdHBd0w4INzQplFDs51upRkNxto

Decryption Key Help Note:- AES algorithm supports 128, 192, and 256-bit encryption & Decryption

Copy Key

b14ca5898a4e4133bbce2ea2315a1916

List of API's

(Click on API name for further details)

1 Directory Data +

2 Heirarchy Data +

3 Water supply scheme data +

4 Institutional Information +

5 Status +

6 FHTC Progress +

7 Financial Progress +

8 Status of Har Ghar Jal +

Har Ghar Jal - Jal Jeevan Mission, "Building Partnerships Changing Lives"

Figure 8: Home Page

On a Home Page (figure 8), one can see important information and functions related to APIs. A description of these is made available in upcoming sections.

Let’s begin with the Menu bar.

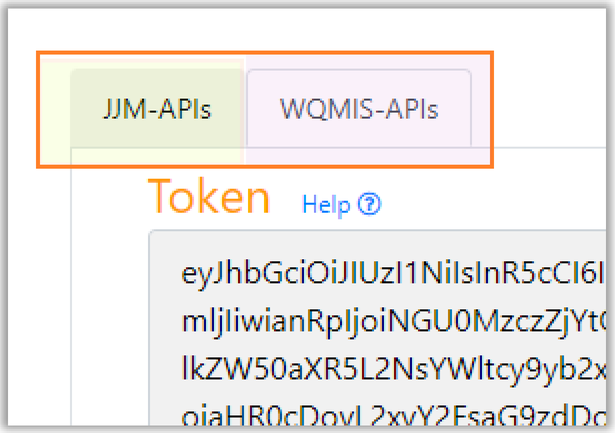
3.1.1. Menu bar

The menu bar is available at Indicator 1. It offers multiple functionalities. For brevity purposes, its description is available in section 3.2. Click here to read about it.

3.1.2. JJM-API/ WQMIS-API

Indicator	Web page label	Purpose
2	JJM-API/ WQMIS-API	The user should click the concerned button to see the required APIs for JJM-API or WQMIS-API. Both offer a similar UI.

Table 3: Switch between JJM-API and WQMIS-API



```
eyJhbGciOiJIUzI1NiIsInR5cCI6I
mIjIiwianRpIjoiNGU0MzczZjYt
IkZW50aXR5L2NsYWltcy9yb2x
oiaHR0cDovL2xvY2FsaG9zdDo
```

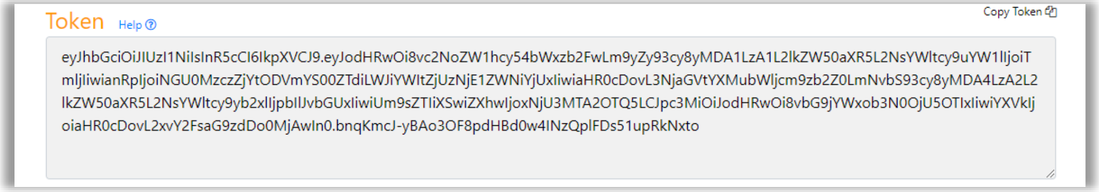
Figure 9

3.1.3. Token

Indicator	Web page label	Purpose
3	Token	This string of characters is popularly known as Token or JWT (JSON Web Token.) It is very significant during journey of fetching a resource using web API. This mechanism (JSON Web Token) helps the JJM server determine whether an incoming request is valid. A user is supposed to attach this token at a specified place in their web application. The process to add this

		token is given in a PDF which is available with a click of the Help button given on RHS. This button is shown below in section 3.1.3.1. for your references.

Table 4: Token description



Token Help ⓘ Copy Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiT
mljIiwianRpIjoiNGU0MzczZjYtODVmYS00ZTdiLWJiYWItZjUzNjE1ZWNiYjUxIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L2
lkZW50aXR5L2NsYWltcy9yb2xlIjpbIlJvbGUxIiwiUm9sZTIiXSwiZXhwIjoxNjU3MTA2OTQ5LCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjU5OTIxIiwiYXVkIj
oiaHR0cDovL2xvY2FsaG9zdDo0MjAwIn0.bnqKmcJ-yBAo3OF8pdHBd0w4INzQplFDs51upRkNxto
```

Figure 10

3.1.3.1. Help button on Token

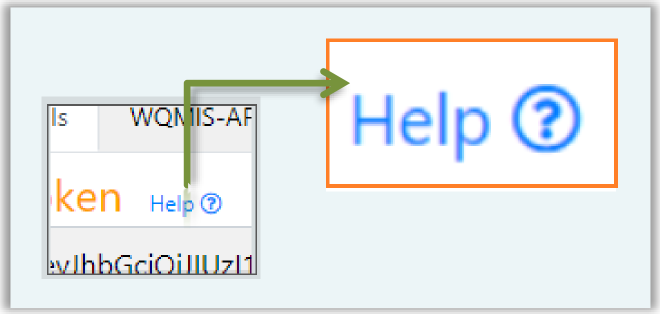
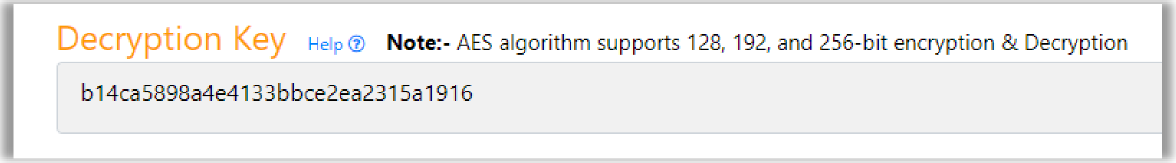


Figure 10.1: Help button

3.1.4. Decryption Key

Indicator	Web page label	Purpose
4	Decryption Key	This key is used when an encoded response is being sent, and that response is needed to be read in plain English/ or other language form (at user end). It supports 128, 192, and 256-bit encryption & Decryption algorithm.

Table 5: Description of Decryption Key



Decryption Key Help ⓘ **Note:-** AES algorithm supports 128, 192, and 256-bit encryption & Decryption

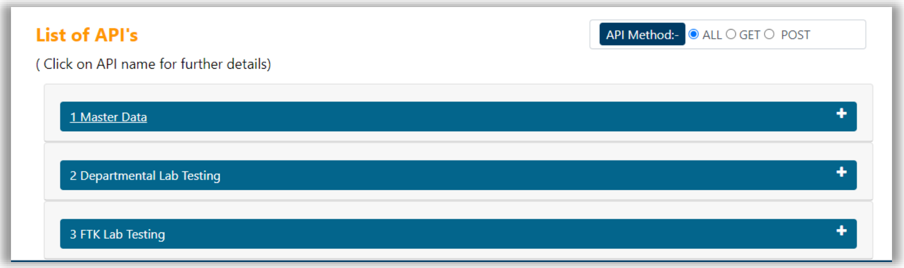
b14ca5898a4e4133bbce2ea2315a1916

Figure 11

3.1.5. List of API's

Indicator	Web page label	Purpose
5	List of API's	It is available at Indicator number 5. It shows a group of APIs for which a user has got usage-permission. This section contains a set of APIs that a user has permission to use. Here a user can filter list of APIs based on GET or POST APIs. It looks similar to figure 12 given below.

Table 6 – Description of List of APIs section



List of API's

(Click on API name for further details)

API Method:- ● ALL ○ GET ○ POST

1 Master Data +

2 Departmental Lab Testing +

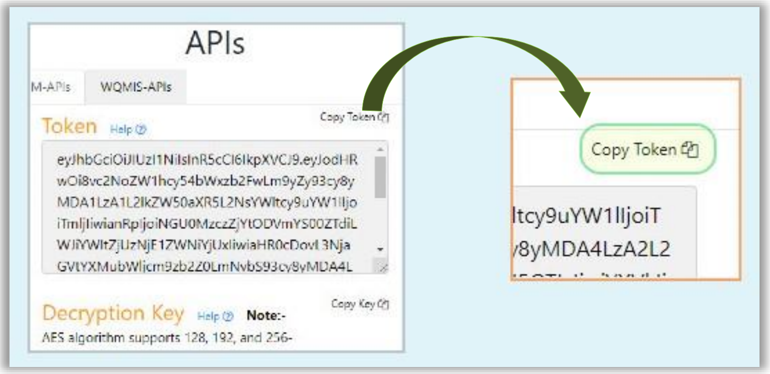
3 FTK Lab Testing +

Figure 12: Display of Collection of APIs category-wise

3.1.6. Copy Token/ Copy Key

Indicator	Web page label	Purpose
6	Copy Token/ Copy Key (button)	Either of the buttons lets a user to copy desired Token or Key to his clipboard and enables him to use according to his requirement. (on the PC)

Table 7 – Description of Copy Token/ Copy Key button



APIs

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHR
wOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8y
MDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjo
iTmljIiwianRpIjoiNGU0MzczZjYtODVmYS00ZTdiL
WJiYWItZjUzNjE1ZWNiYjUxIiwiaHR0cDovL3Nja
GVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4L
```

Note:- AES algorithm supports 128, 192, and 256-

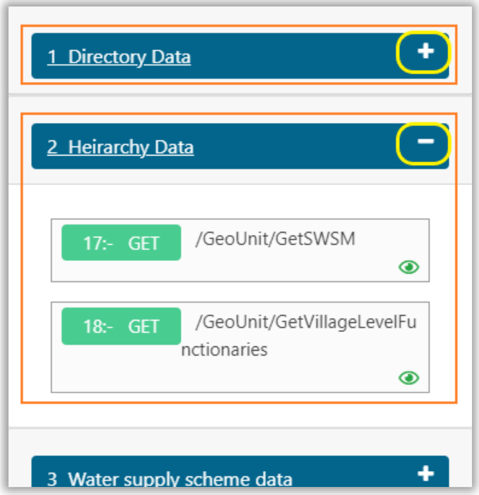
```
ltcy9uYW1lIjoiT
/8yMDA4LzA2L2
```

Figure 13: Demonstration of Copy button

3.1.7. +/ – (To expand or collapse view of API categories)

Indicator	Web page label	Purpose
7	+/ – (button)	This button helps expand a category. Once an API category expands, this + plus changes to a – button. This – (minus) button helps a user to collapse the expanded view, and the works for another category of APIs. *(Please refer to figure 14 for better understanding.)*

Table 8: Description of – (Collapse)/ + (Expand) button



1 Directory Data +

2 Heirarchy Data −

17:- GET /GeoUnit/GetSWSM

18:- GET /GeoUnit/GetVillageLevelFunctionaries

3 Water supply scheme data +

Figure 14

3.2.Menu bar (Explained)

In this section, we will be covering menu items one by one. That is available in the header section of the Home page.

3.2.1. Request APIs

Objective:

On this web page user can observe those APIs for which usage-permission is unavailable, and he can request for them according to the requirement.

(Refer to figure 16, please.)

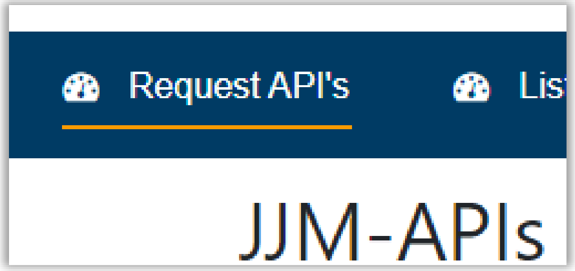
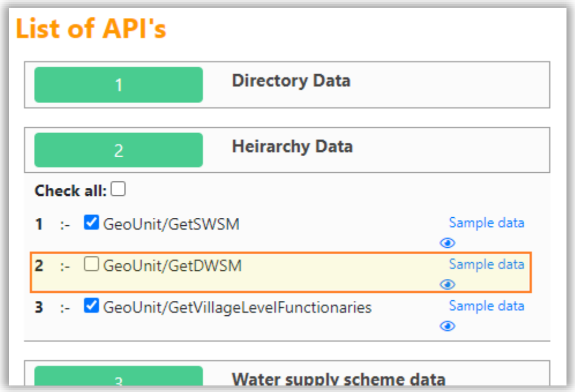


Figure 15: Request APIs menu



List of API's

1 Directory Data

2 Heirarchy Data

Check all: ☐

1 :- ☑ GeoUnit/GetSWSM Sample data

2 :- ☐ GeoUnit/GetDWSM Sample data

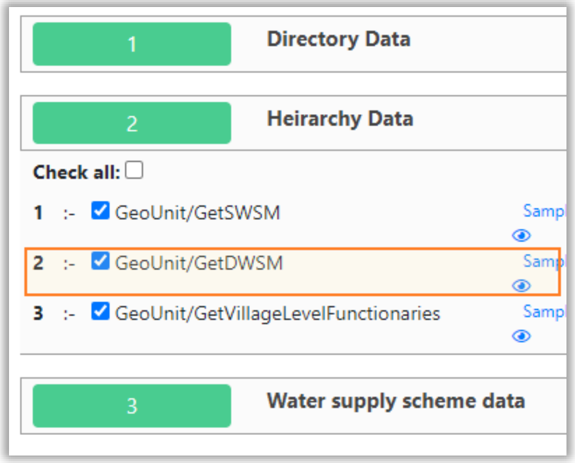
3 :- ☑ GeoUnit/GetVillageLevelFunctionaries Sample data

3 Water supply scheme data

Figure 16

Procedure to request an API for which a user has not received permission:

Step 1) Mark/ check the API and click on the Submit button. *(Refer to figure 17 given on the right)*



1 Directory Data

2 Heirarchy Data

Check all: ☐

1 :- ☑ GeoUnit/GetSWSM Samp

2 :- ☑ GeoUnit/GetDWSM Samp

3 :- ☑ GeoUnit/GetVillageLevelFunctionaries Samp

3 Water supply scheme data

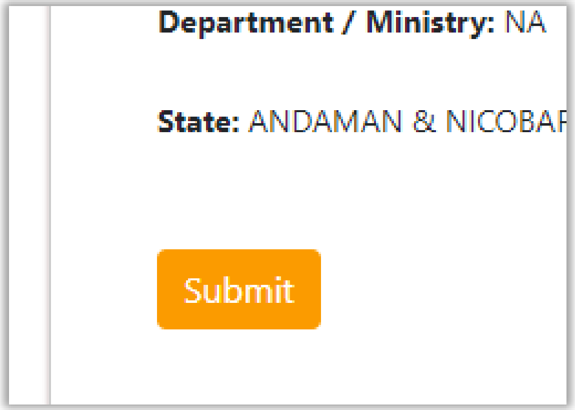
Figure 17

Step 2) Click on the Submit button.

Note:

This is how a request for an API has been generated and a user should wait for 24-48 hours for approval.

(Refer to figure 18 given on the right)



Department / Ministry: NA

State: ANDAMAN & NICOBAR

Figure 18

3.2.2. List of APIs

Objective:

This web page has the same output when a user logs-in on the website. It looks similar to the figure 8 of section 3.1.

(Refer to figure 16.)

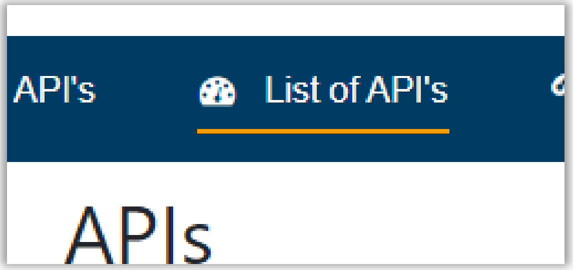


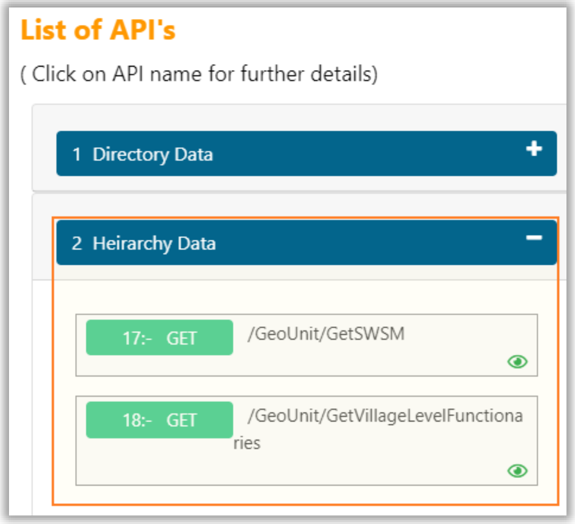
Figure 19

Here, those APIs are available for which he has received usage-permission.

(Refer to figure 20 and figure 16)

A user should compare the *figure 20* with figure 16 for a better understanding.

The figure 16 lists three APIs where one out of three is unchecked, i.e., unavailable for user.



List of API's

(Click on API name for further details)

1 Directory Data +

2 Heirarchy Data −

17:- GET /GeoUnit/GetSWSM

18:- GET /GeoUnit/GetVillageLevelFunctionaries

Figure 20

3.2.3. Change Password

Objective:

To change password a user is supposed work on this web page. A user is presented with the following figure 22.

(Refer to the figure 22 given below, please)

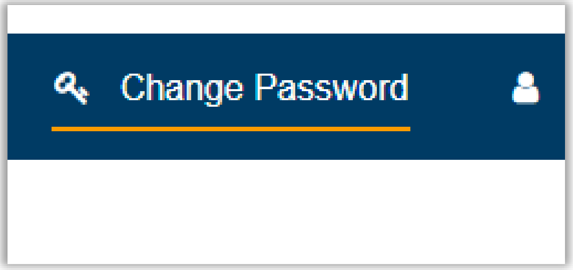


Figure 21

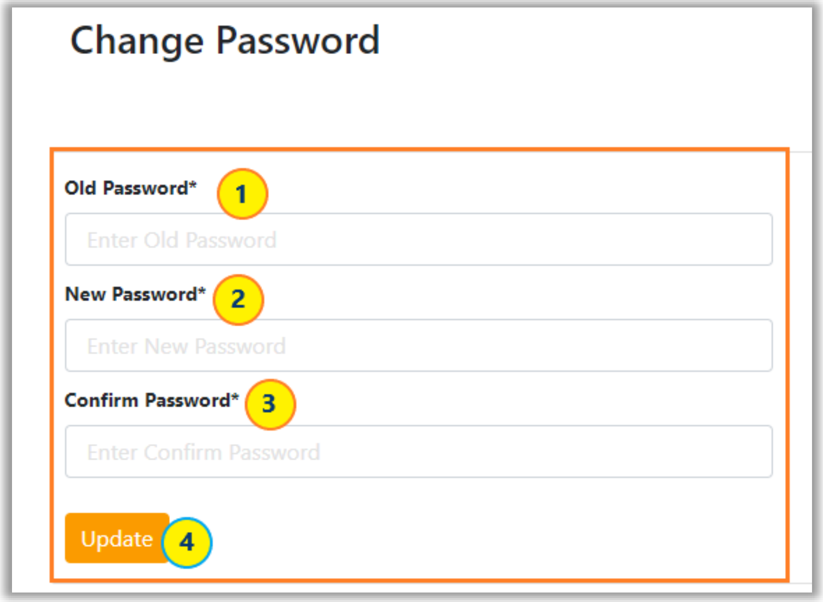


Figure 22: Demo of Change Password process

Indicator	Web page label	Purpose
1	Old Password	Here old password should be entered.
2	New Password	New password must conform the following rules: 1. *The chosen password can be in between 8 to 20 characters.* 2. *It may be alphanumeric, though it must contain minimum one uppercase letter, one lower case letter, and one special character too.*
3	Confirm Password	Please reenter the password chosen here. Ensure both passwords match; otherwise application may reject the change.
4	Update (button)	Click on the button to save the changes.

Table 9: Demonstration of Change Password feature

3.2.4. User-Name

This menu item depicts a username.

3.2.5. Sign Out

With this menu item a user can end his current work session. The user is transferred back to the Login-cum-welcome page of JJM APIs (website) once he clicks on this menu-item.

Note:

1. *Please ensure changes made by a user should save using 'Submit/ Update' button before Sign Out.*
2. *Please, contact to system admin if data is lost due to power failure or some other external/ internal factor.*

In the next chapter, we'll be learning about displaying (and reviewing) the response of an API without having it attached to a web page. Let's move forward to chapter 4, 'Reviewing a Response.'

4. Preview a Response

We have discussed about the features of this website until now.

In this section, we talk about how a user can sneak into (preview) response of an API (that he is allowed to use.)

How to review a response

A response can be viewed in the following ways/ places.

- At Try it Yourself (feature) at List of APIs section
- At Request API (menu bar)

Note:

1. *We hereby assume a user has logged in with his provided credentials.*
2. *We also assume that a User is aware of his web API requirement.*

Let’s get familiar with the above in the following sections.

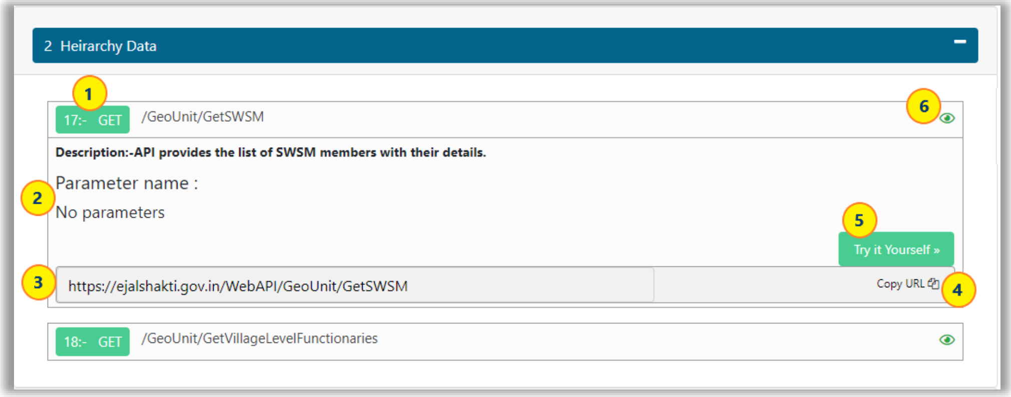
4.1. Using “Try it Yourself” (feature) at List of APIs section

This menu option lets a user to preview those APIs for which are registered for his account. Here a user can preview the response using the following workflow:

4.1.1. Expand an API by clicking on it or on plus (+) button. This can also be used for checking major details of an API. (Demonstration of it is provided at figure 23 and Table 10 given below.)

4.1.2. Click on the ‘Try it Yourself’ button.

4.1.3. Click on the Execute button.



2 Heirarchy Data

17:- GET /GeoUnit/GetSWSM

Description:-API provides the list of SWSM members with their details.

Parameter name :

No parameters

https://ejalshakti.gov.in/WebAPI/GeoUnit/GetSWSM

18:- GET /GeoUnit/GetVillageLevelFunctionaries

Figure 23: Review of details of an API

Indicator	Web page label	Purpose
1	Type of API	This label informs a user if it is a GET API or POST API. A GET type of API is reflected in Green whereas a POST API is reflected as Orange.
2	Parameter name	A Parameter helps an API to fetch the exact resource from a large pool of data.
3	URL	This URL tells the JJM server about the resource a user is looking for.
4	Copy URL (button)	This button copies the URL given on number 3 to clipboard.
5	Try it Yourself (button)	This button lets a user to see the output of this API. To observe the output a user is supposed to click on this button and it opens the same in new browser tab. Where he may select a few more options according to the requirement to get preview of resources it (API) would return.
6	Eye (button)	This is a toggle button. Which expands description of an API if it is in closed mode, and does the opposite if it has been already expanded.

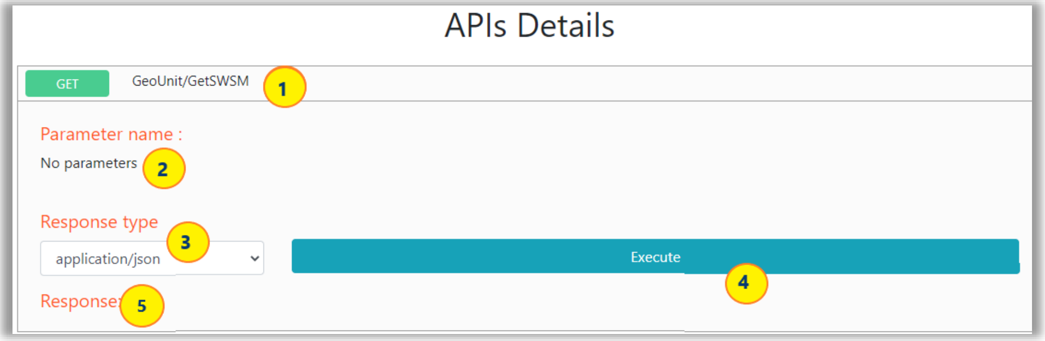
Table 10: UI description of an API detail feature

Demonstration of UI of the Try It Yourself feature:

Indicator	Web page label	Purpose

1	Name of API	Name of the API.
2	Parameter name	It depicts name of the Parameter that is required.
3	Response type	A user is supposed to select the response format.
4	Execute (button)	This button displays the result that helps the user to determine output for his various requirements. *(Refer to figure 24.)*
5	Response	Upon click of the Execute button, it prints the response that a user would get during real-time operation. *(Refer to figure 25 given below, please.)*

Table 11: Demonstration of the 'Try It Yourself' feature's UI



APIs Details

GET GeoUnit/GetSWSM

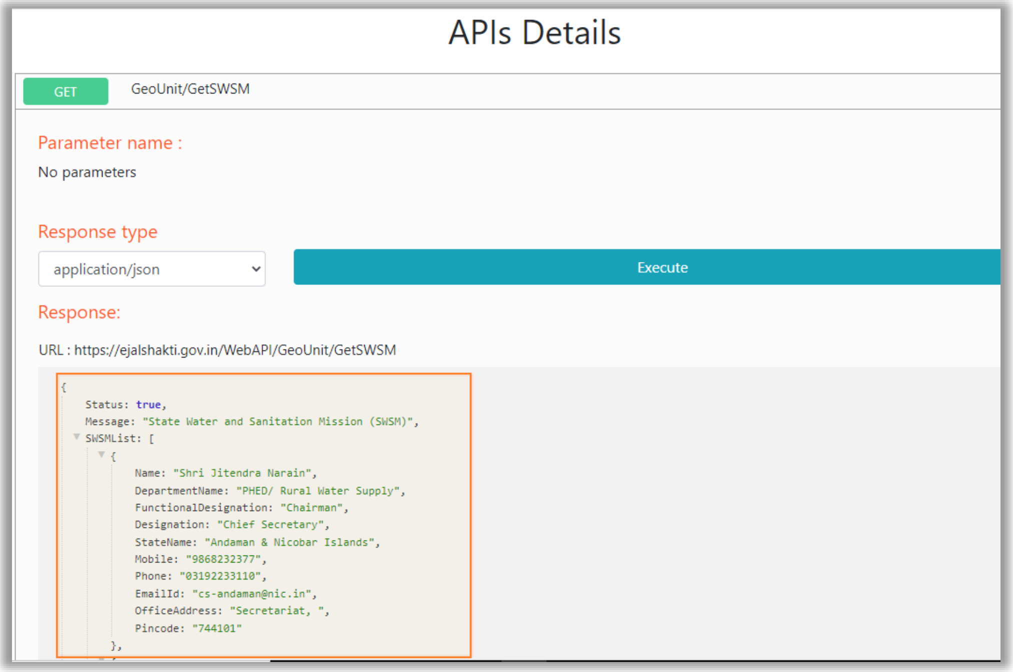
Parameter name :
No parameters

Response type
application/json

Response

Figure 24: UI of APIs Details

Once a user clicks on the Execute button, response of the API is printed as demonstrated in figure given below. It basically confirms the output a user would get. *(Refer to figure 26.)*



APIs Details

GeoUnit/GetSWSM

Parameter name :

No parameters

Response type

application/json

Response:

URL : https://ejalshakti.gov.in/WebAPI/GeoUnit/GetSWSM

```
{
    Status: true,
    Message: "State Water and Sanitation Mission (SWSM)",
  ▼ SWSMList: [
      ▼ {
            Name: "Shri Jitendra Narain",
            DepartmentName: "PHED/ Rural Water Supply",
            FunctionalDesignation: "Chairman",
            Designation: "Chief Secretary",
            StateName: "Andaman & Nicobar Islands",
            Mobile: "9868232377",
            Phone: "03192233110",
            EmailId: "cs-andaman@nic.in",
            OfficeAddress: "Secretariat, ",
            Pincode: "744101"
        },
```

Figure 25: Demonstration of output of Response of an API

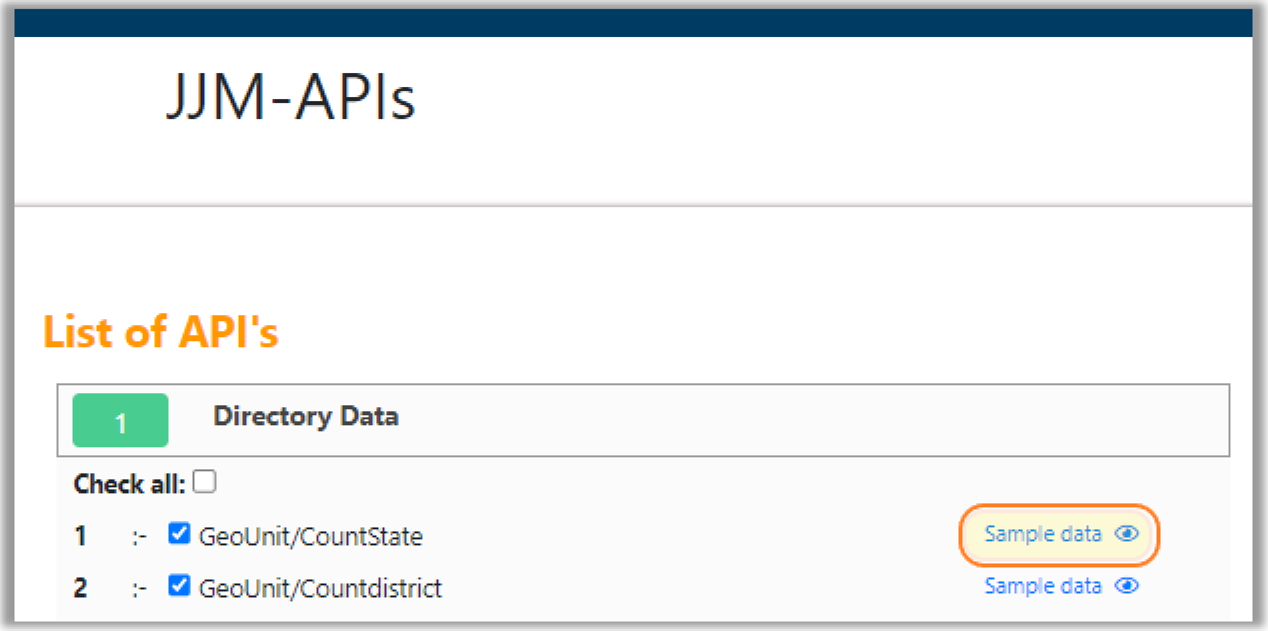
4.2. Review in Request API menu

The Request API feature is helpful when a user needs an API which is not available in his list/ account, and he wants to view the output of it. Then he can request it by following the procedure explained in the section 3.2.1.

In the section 3.2.1., we became familiar with the process and in the current section we'll be covering how we can preview an API that seems significant for our requirements.

Let's take a brief look on this process now:

(Refer to figure 28 given below)



JJM-APIs

List of API's

1 Directory Data

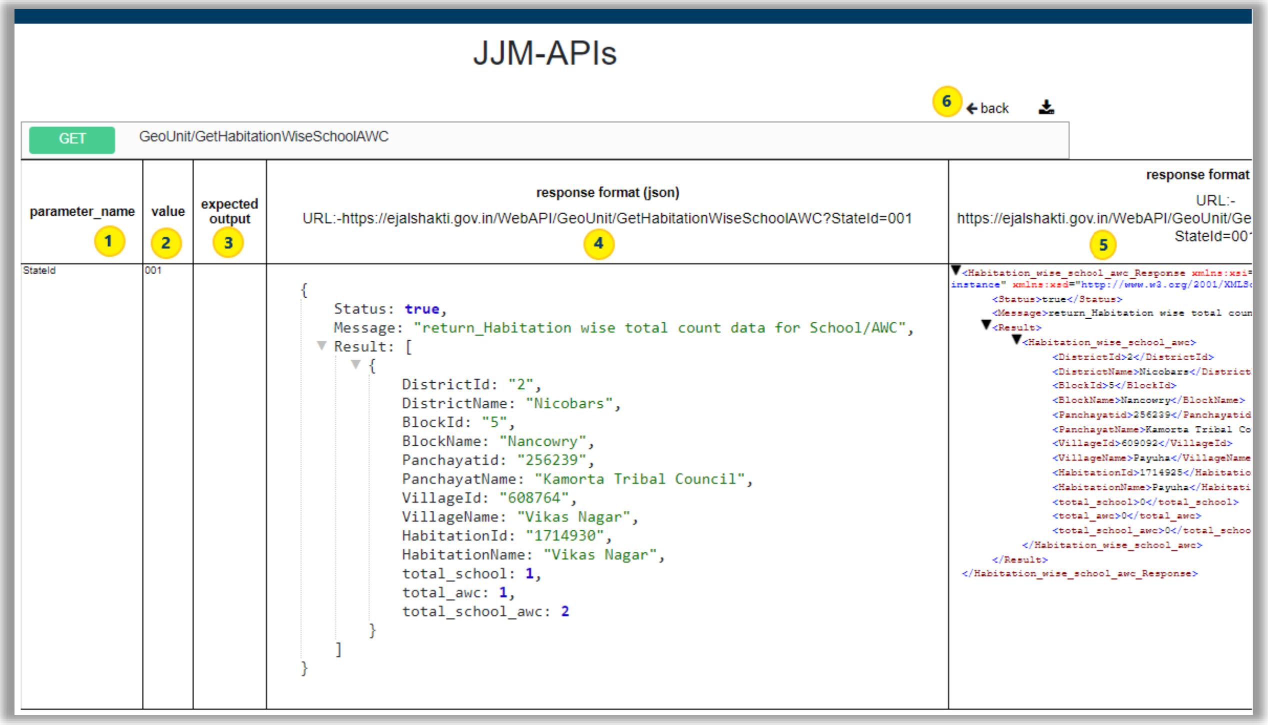
Check all: □

1 :- ☑ GeoUnit/CountState Sample data

2 :- ☑ GeoUnit/Countdistrict Sample data

Figure 26

To get the details of an API, a user should click on the ‘Sample Data’ button. It takes him to a web page that looks similar to figure 27. *(Please refer to figure 27.)*



JJM-APIs

GET GeoUnit/GetHabitationWiseSchoolAWC

parameter_name	value	expected output	response format (json) URL:-https://ejalshakti.gov.in/WebAPI/GeoUnit/GetHabitationWiseSchoolAWC?StateId=001	response format URL:- https://ejalshakti.gov.in/WebAPI/GeoUnit/Ge StateId=00
StateId	001			

```
{
   Status: true,
   Message: "return_Habitation wise total count data for School/AWC",
 ▼ Result: [
     ▼ {
           DistrictId: "2",
           DistrictName: "Nicobars",
           BlockId: "5",
           BlockName: "Nancowry",
           Panchayatid: "256239",
           PanchayatName: "Kamorta Tribal Council",
           VillageId: "608764",
           VillageName: "Vikas Nagar",
           HabitationId: "1714930",
           HabitationName: "Vikas Nagar",
           total_school: 1,
           total_awc: 1,
           total_school_awc: 2
       }
   ]
}
```

```
▼<Habitation_wise_school_awc_Response xmlns:xsi=
instance" xmlns:xsd="http://www.w3.org/2001/XMLS
     <Status>true</Status>
     <Message>return_Habitation wise total coun
   ▼<Result>
      ▼<Habitation_wise_school_awc>
          <DistrictId>2</DistrictId>
          <DistrictName>Nicobars</District
          <BlockId>5</BlockId>
          <BlockName>Nancowry</BlockName>
          <Panchayatid>256239</Panchayatid
          <PanchayatName>Kamorta Tribal Co
          <VillageId>609092</VillageId>
          <VillageName>Payuha</VillageName
          <HabitationId>1714925</Habitatio
          <HabitationName>Payuha</Habitati
          <total_school>0</total_school>
          <total_awc>0</total_awc>
          <total_school_awc>0</total_schoo
       </Habitation_wise_school_awc>
     </Result>
  </Habitation_wise_school_awc_Response>
```

Figure 27: Description of UI of Details an API

Description of API detail page as visible in figure 27 above.

Indicator	Web page label	Purpose
1	Name of Parameter	It depicts the name of parameter which a user should pass to this API. If an API doesn't require a parameter, then the parameter name will be NA.
2	Value	It depicts the value that it requires. If no parameter required then value is 0. (zero)
3	Expected output	It displays the output (purpose) that a user is supposed to get.
4	Response format (JSON)	The actual output of this API is demonstrated in JSON format.
5	Response format (XML)	The output of this API is demonstrated in XML format.
6	Back (button)	Using this button a user can go to Request API page.

7	Download (button)	Functionality of this button is demonstrated in the next section. *(Please refer to section 4.3 given below.)*

Table 12: Description of API detail page

4.3. Download output in a PDF file:

The output of an API can be saved (download) as PDF on the machine of user. To do user need to click on the download button ⬇ provided at the top.

This file is available in “Downloads” folder of the system with name of short description (purpose) of API. *(Refer to figure 28, please.)*

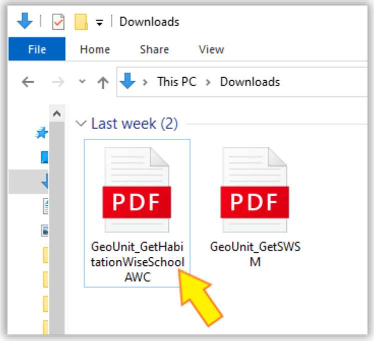


Figure 28

I. Appendix

Here, a user learns to add a Token in his web application, step-by-step.

1) How to Use a JWT Token

- Once the user has received the approval, he should work on to attach it to this web application.
- The Authorization header must be specified as demonstrated in the code given below:

```
httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer",
"Your JWT token");
```

Note:

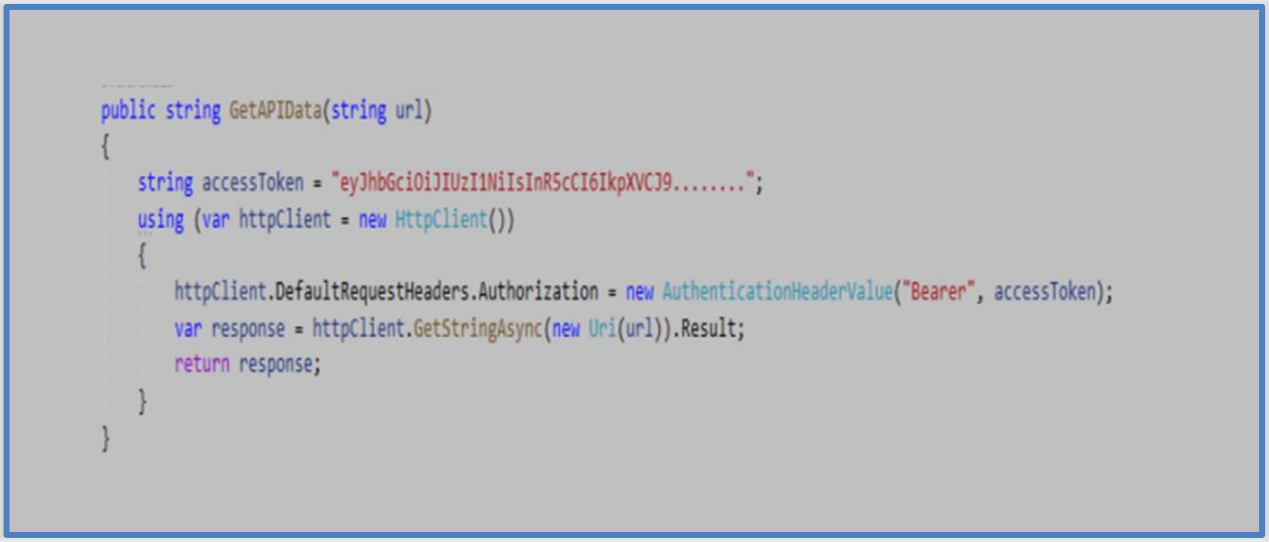
1. *After a token has been created, it cannot be modified.*
2. *API token is valid for one year.*
3. *Create a new access token when the old one is about to expire.*

2) Code Example

While making a request to the JJM server, a user is supposed to include the acquired token in the <head> section of his web application.

Example

The following example (code) returns a JSON version of the list of available REST API endpoints for his environment:



```
public string GetAPIData(string url)
{
    string accessToken = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9........";
    using (var httpClient = new HttpClient())
    {
        httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", accessToken);
        var response = httpClient.GetStringAsync(new Uri(url)).Result;
        return response;
    }
}
```

We hope this document answers queries about working with a JJM API excellently.

Using this document, a user can successfully access an API from the JJM server for his various requirements.