Integration Standard of LISF (Land Information and Services Framework)

Cabinet division | Ministry of Land | a2i, Prime Minister’s Office

Technical assistance by:

Soft BD Ltd.

Flat No:B5, Panthavila,
58/11 Box Culvert, Panthapath
Dhaka 1205 (Opposite of Bashundhara City).
Mobile: +880-1710334756, +880-1917778999
Email: info@soft-bd.com
Website: SoftBD Ltd.

Version **1.4**

Status **Waiting For Final Approval**

	NAME	SIGNATURE	DATE
Author	**Atiqul Islam Khan**		
Reviewer	**Hafijur Rahman**		
Authoriser	**Arfe Elahi Manik**		

Effective Date:	

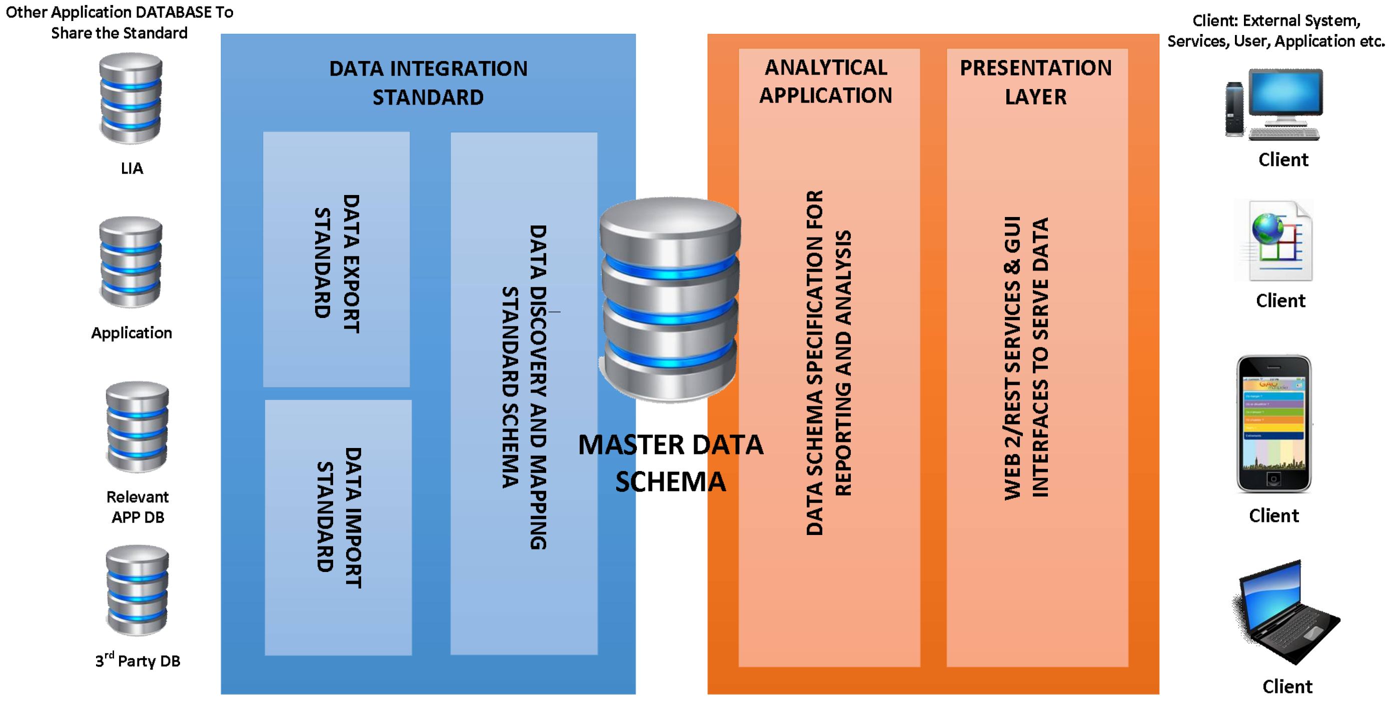
	READ BY			
NAME	**Version**	**Team**	**Signature**	**Date**
Tanvir Ahmed	**0.4**	**SoftBD**		**15/12/2015**
Hafijur Rahman	**0.5**	**A2I**		**16/12/2015**
Mahmudur Rahman	**0.8**	**SoftBD**		**03/02/2016**
Mahmudur Rahman	**1.1**	**SoftBD**		**21/02/2016**
Atiqul Islam	**1.4**	**SoftBD**		**24/02/2016**

Table of Contents

Description

The LISF concept and principle of “**One Central Land Framework**” has strong and clear correlation with the integration principles and standards. As such the integration standards in this document are designed to ensure all Land related applications are able to share data with LISF core database in a standardized format.

By Data Integration (DI) we mean using data abstraction through data definitions to provide a unified interface to facilitate uniform data access for viewing and utilizing data.


Other Application DATABASE To Share the Standard
LIA
Application
Relevant APP DB
3rd Party DB
DATA INTEGRATION STANDARD
DATA EXPORT STANDARD
DATA IMPORT STANDARD
DATA DISCOVERY AND MAPPING STANDARD SCHEMA
MASTER DATA SCHEMA
ANALYTICAL APPLICATION
DATA SCHEMA SPECIFICATION FOR REPORTING AND ANALYSIS
PRESENTATION LAYER
WEB 2/REST SERVICES & GUI INTERFACES TO SERVE DATA
Client: External System, Services, User, Application etc.
Client
Client
Client
Client

Integration Methods (In order of preference)

1. Utilizing the Services Oriented Architecture (SOA) by invocation of a web service (W3C compliant).

2. Public or Private both data access requires service/application registration into LISF.

3. LISF provides only method level integrations. LISF integration manager control this list of method.

4. Flat file Transfers/ Direct SQL access are strictly prohibited.

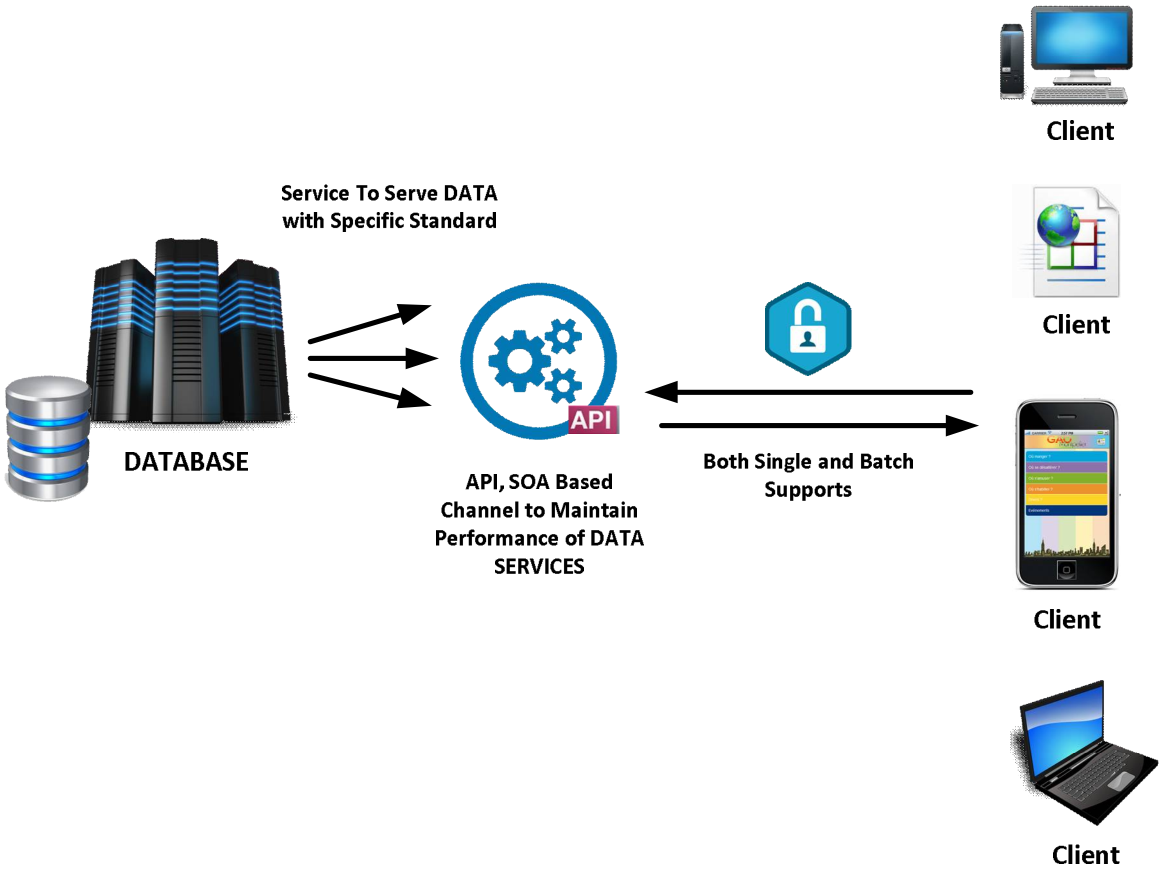
5. Access to private data via LISF published method, all applications or services required secure (HTTPS) connection.

6. Data Transfer using web services follow REST architecture with protocol RFC 6690.

7. LISF Data integration can be categorized as either event based not batch load/processing.

8. No Storing of Any private LISF Core data content.

9. Other Point to Point (application/service to other application/service) methods are not covered or no issue's for LISF data integration.


Client
Service To Serve DATA with Specific Standard
Client
API
DATABASE
API, SOA Based Channel to Maintain Performance of DATA SERVICES
Both Single and Batch Supports
Client
Client

Bi-directional Data integration

In a bi-directional, data-driven integration, data is synced any time it is changed, not just when a particular event occurs. As LISF nature or business point of view, LISF wish & allow integrating such Applications/services which are qualified to update/create LISF core data definition.

With necessary permission & fulfillment of pre-requisites, only approved services/applications can use bi-directional data integration method in following way.

a. Initiate Communication.
b. Authenticating the service/application.
c. Authorization of specific request from that service/application.

d. If all parameters are valid & having proper authorization of data, for specific that request new set of access token issued by LISF to that application/service.
e. Before finalizing new/update Core Data Definition, Prompting re-consent by matching new access token & old token.
f. Issue Update/Insert confirmation receipt to service/application.
g. Records each & every action for future inspection.
h. This operation must be done within fixed time limit.

Application Development and Acquisition

It is a requirement of all applications/services to be able to support 1 to 6 and 8 of the above methods for both consuming and contributing data in the process.

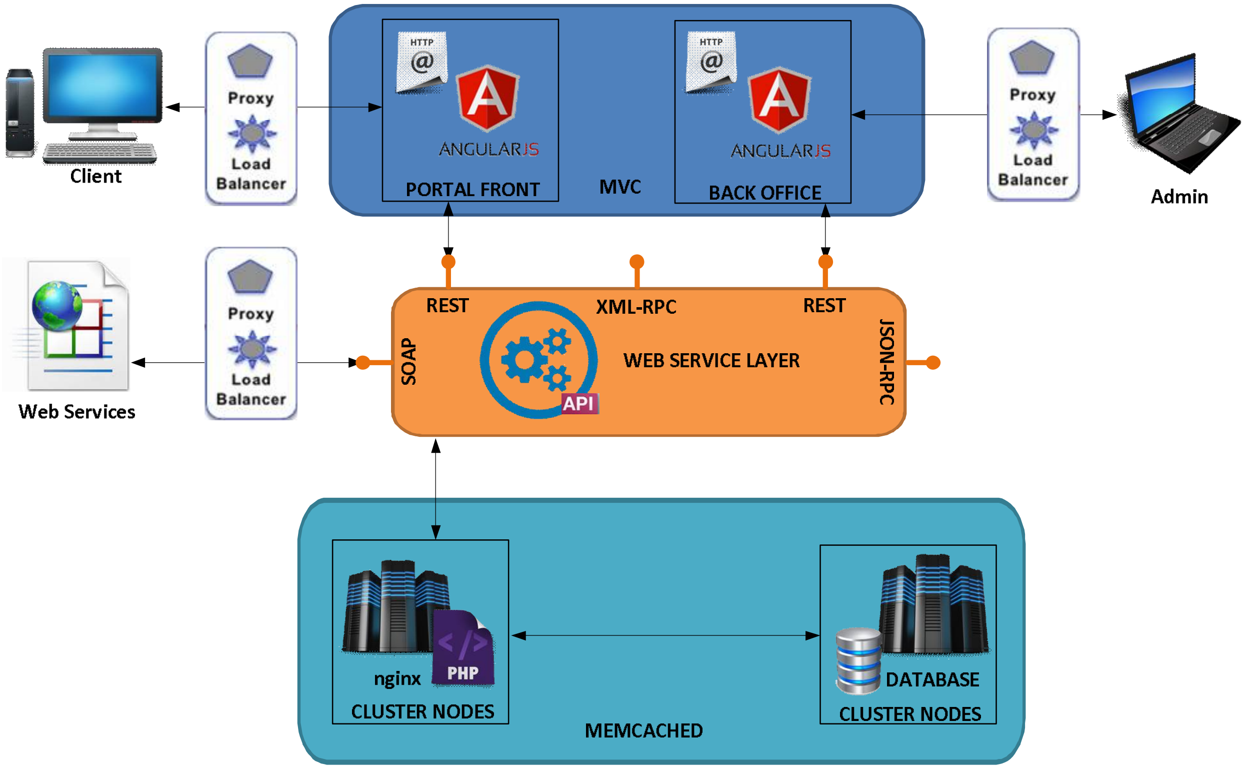
Applications/services that are unable to any standards may be reviewed at regular intervals and an alternative solution that offers a more sustainable integration solution may be adopted, particularly if the application is or has been identified as an origin data host otherwise LISF may restrict or cancel those applications/services registration.

Core Data Formats

Data must only be transferred from the middleware layer (webMethods) and the destination system in a format which enforces constraints of the appropriate CDD. Once each Core Data record has passed through the middleware layer and been validated by it may then be remapped to the required destination system data format (i.e. Transformed Core Data)

Hosting Models

Internally as well as externally hosted solutions are supported by LISF webMethods integration. Consideration should be given the potential latency involved with using an externally hosted solution as a subscriber of LISF data for a web service. A tolerance of no more than X milliseconds is seen as acceptable for propagation of data, which is separate to the execution time of the service.



Connectivity

Where an application/service has no platform independent API for webMethods to interact with and LISF ensure no permission is given to direct access to platforms supported in the Infrastructure standards document.

LISF integration policies always prefer secure connection whatever data type (private/public) access. In that Case LISF data protocol will give highest priority to serve those methods call which comes from secured connection.

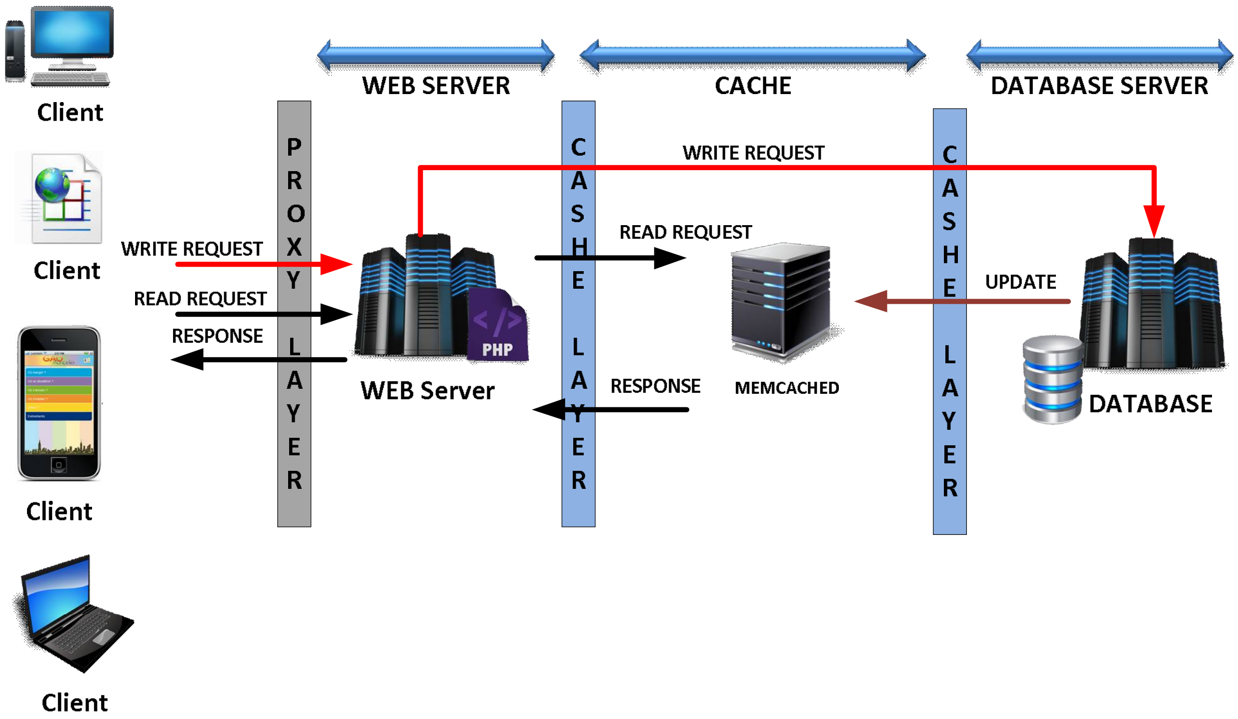
Protocols

Data transferred between internally hosted and externally hosted systems is only permissible if transport layer supports https for Private Data access or HTTP for Public Data access.

Data Caching

When any LISF published method with specific parameters serves very frequently, a copy must be provisioned in the LISF Core database in order to provide faster and alternate read only source for web services to use.

This cache copy will be removed if that specific method’s popularity decreasing.



Multi-Mastering & Consistent Data Bridge with Other Data source

LISF core database requires persistent & consistent **Data BRIDGE** with other potential **DATA SOURCE**. As in this process LISF core database uses Multi-mastering for that selected portion of data source.

Such LISF core database need to be persistent **synced** with **E-Filling** Core database for office, user data source. Any Changes on that data source will be reflected within a millisecond in LISF Core database.

Same methodology requires for **NID** access for user information & **BBS code** access for Area data information in LISF Core database architecture.

Directory Services / Authentication / Authorization

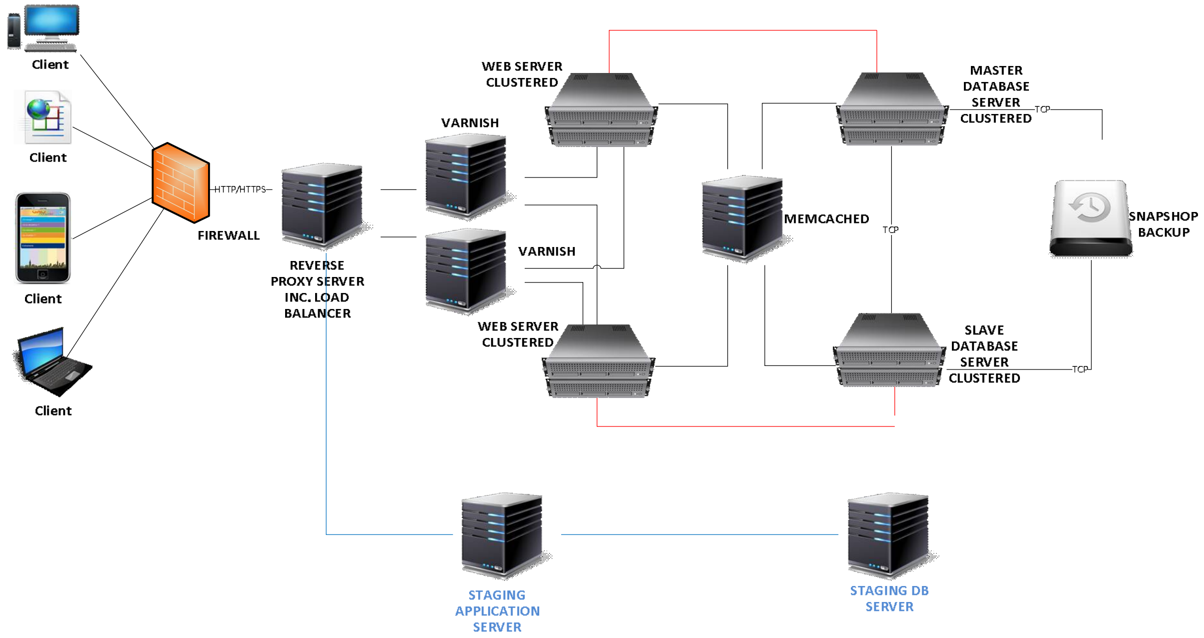
There are no supported directory services for the integration of applications/services such as Active Directory (AD) and LDAP.

Whilst LISF primary & only focus is to facilitate authentication, attributes considered core to authentication/authorization process may be accessible to applications as a form of basic integration.

For that purpose all applications/services need to obtain specific service code & each method access code for using to integrate systems by registered & approved by LISF Core system.

Business Continuity / Disaster recovery

Any origin or destination system's business continuity measures must remain transparent to webMethods. Only in extreme DR cases will a webMethods configuration be altered to accommodate a change necessitated by a disaster recovery plan being enacted.



Exceptions

As the level diversity within the application/services portfolio increases at a rapid pace, new applications requiring LISF published method may exhibit certain behavior which may not be explicitly covered in the integration standards. If this is the case it should be raised in an invitational meeting where a decision will be made to either incorporate the requirements into the integration standards to encompass new technologies and/or methodologies which adhere to the integration principles or grant a one off exception. In the event the standards cannot be modified to accommodate an application requiring LISF, then an exception to the integration standards can be considered on a case by case basis.

1. A compelling business case is presented to LISF
2. Data involved is classified as Level 1 in the Information standards.

Some Popular & Major Land Service/Application

This application/service may integrate into LISF Framework.

	Service/Application Name	Office Level
1	Allotment of agricultural government land to landless people	District/Upazila
2	Chandina Viti	District/Upazila
3	Mutation & Joma Vag	Upazila
4	Porcha/Records Correction instructed by Court	District/Upazila
5	Allotment of non-agricultural government land	District/Upazila
6	Lease extension of vested property	District/Upazila
7	Change lease owner of vested property	District/Upazila

8	Management of Land development tax	Upazila
9	Miss case Management	Upazila
10	Resolve Land type change request	District/Upazila
11	Land allotment to ethnic people	District/Upazila
12	Management of Abandoned Property	District/Upazila
13	Correction of khatian clerical mistake	Upazila
14	Management of Government Owned Land	District/Upazila
15	Khatian/Porcha Management from DC office record room	District
16	Mouja Map Management from DC office record room	District
17	Wetland, Pond & swamp Management	District/Upazila

Method Level Integration Overview

All land related applications/services needs to be pre-registered with LISF to integrate any of its core data.

During the successful registration process each & every application/service will receive a unique Service ID and different (unique) Access-code of selected method from LISF published integration methods list.

All applications/services may register for more than one integration method in LISF concurrently using their assigned Service ID and method base specific Access-code.

Integration methods can be categorized into two groups Public and Private by data access point of view.

	Name	Req. Type	Work Type	Access Type
1	{LISF BASE URL}/api/user-check	Read	User	Private
2	{LISF BASE URL}/api/user-login-office	Read	User	Private
3	{LISF BASE URL}/api/user-profile	Read	User	Private
4	{LISF BASE URL}/api/office-profile	Read	Office	Public
5	{LISF BASE URL}/api/office-tree	Read	Office	Public
6	{LISF BASE URL}/api/office-organogram	Read	Office	Public
7	{LISF BASE URL}/api/office-user-organogram	Read	Office, User	Private
8	{LISF BASE URL}/api/office-branch-list	Read	Office	Public
9	{LISF BASE URL}/api/office-branch-user-list	Read	Office, User	Private
10	{LISF BASE URL}/api/area-office-count	Read	Office	Public
11	{LISF BASE URL}/api/khatian-generate	Read	Record	Private
12	{LISF BASE URL}/api/dag-info-mouja	Read	Record	Private
13	{LISF BASE URL}/api/khatian-dag-info	Read	Record	Private
14	{LISF BASE URL}/api/khatian-owner-info	Read	Record	Private
15	{LISF BASE URL}/api/land-owner-list	Read	Record	Private
16	{LISF BASE URL}/api/land-owner-in-mouja	Read	Record	Private
17	{LISF BASE URL}/api/khas-land-mouja	Read	Record	Private
18	{LISF BASE URL}/api/khatian-count-mouja	Read	Record, Area	Public
19	{LISF BASE URL}/api/khatian-count-upazila	Read	Record, Area	Public
20	{LISF BASE URL}/api/reference-khatian	Read	Record	Private
21	{LISF BASE URL}/api/reference-dag	Read	Record	Private
22	{LISF BASE URL}/api/khatian-tax-info	Read	Record	Private
23	{LISF BASE URL}/api/division-list	Read	Area	Public
24	{LISF BASE URL}/api/district-list	Read	Area	Public
25	{LISF BASE URL}/api/Upazila-list	Read	Area	Public
26	{LISF BASE URL}/api/Mouja-list	Read	Area	Public

27	{LISF BASE URL}/api/union-list	Read	Area	Public
28	{LISF BASE URL}/api/pouroshova-list	Read	Area	Public
29	{LISF BASE URL}/api/Pouroshova-ward-list	Read	Area	Public
30	{LISF BASE URL}/api/city-corporation-list	Read	Area	Public
31	{LISF BASE URL}/api/citycorporationward-list	Read	Area	Public
32	{LISF BASE URL}/api/area-info	Read	Area	Public
33	{LISF BASE URL}/api/area-change-info	Read	Area	Public
34	{LISF BASE URL}/api/plot-draw-2d-map	Read	Area	Private
35	{LISF BASE URL}/api/deed-info-by-khatian	Read	Deeds	Private
36	{LISF BASE URL}/api/deed-info-by-dag	Read	Deeds	Private
37	{LISF BASE URL}/api/ deed-info-by-owner	Read	Deeds	Private
38	{LISF BASE URL}/api/deed-info-by-seller	Read	Deeds	Private
39	{LISF BASE URL}/api/ deed-info-by-deed-id	Read	Deeds	Private
40	{LISF BASE URL}/api/ deed-info-by-area	Read	Deeds, Area	Private
41	{LISF BASE URL}/api/ deed-info-by-office	Read	Deeds, Office	Private
42	{LISF BASE URL}/api/list-ap-land-lease	Read	Record	Private
43	{LISF BASE URL}/api/list-vp-land-lease	Read	Record	Private
44	{LISF BASE URL}/api/list-expired-land-lease	Read	Record	Private
45	{LISF BASE URL}/api/write-new-khatian	Write	Record	Private
46	{LISF BASE URL}/api/update- khatian	Write	Record	Private
47	{LISF BASE URL}/api/new--lease-owner	Write	Record	Private
48	{LISF BASE URL}/api/update- lease-owner	Write	Record	Private

A) USER & OFFICE Related (View Only)

1) User Existence Check / User Login without OFFICE

Method	URL
GET	{LISF BASE URL}/api/usercheck/

Input Parameter

Name	Type	Description
service_id	String	**Required**. The service ID you received from LISF when you registered
username	String	**Required**.
password	String	**Required**.
access_code	String	**Required**. Access code defined for each API binding with service.
return_type	Integer	**Required**. A. 1 use for JSON, b. 2 use for XML
ip_address	String	**Optional.**
user_info	Integer	**Optional.** A. 1 use for Only True, b. 2 use for Detail Information

Return Response

Code	String
200	OK! [“1”] [“full name BD”, ”full name EN”, “profile pic”, ”designation name”, ”office name”]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
603	{"error":" Please provide username."}
604	{"error":" Please provide user password."}
625	{"error":" Incorrect username or password."}
606	{"error":" Return Type Missing."}
641	{"error":" Blocked User ID."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call this authentication API again.

Service List

Every Service related to LISF can use this API.

2) User Existence Check / User Login with OFFICE

Method	URL
POST	{LISF BASE URL}/api/userloginoffice/

Input Parameter

Name	Type	Description
service_id	String	**Required**. The service ID you received from LISF when you registerec
username	String	**Required**.
Password	String	**Required**.
access_code	String	**Required**. Access code defined for each API binding with service.
office_id	String	**Required.**
return_type	Integer	**Required**. A. 1 use for JSON, b. 2 use for XML
ip_address	String	**Optional.**

Return Response

Code	String
200	[“full name”, “profile pic”, ”designation”, ”office name”]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
603	{"error":" Please provide username."}
604	{"error":" Please provide password."}
605	{"error":" Please provide office id."}
625	{"error":" Incorrect username or password."}
606	{"error":" Return Type Missing."}
641	{"error":" Blocked User ID."}
701	{"error":" Wrong Office ID."}
702	{"error":" Inactive Office."}

Data Validity Duration

No warranty provided.

Payment

N/A

Service List

Every Service related to LISF can use this API.

3) User Profile Information

Method	URL
GET	{LISF BASE URL}/api/userprofile/

Input Parameter

Name	Type	Description
service_id	String	**Required**. The service ID you received from LISF when you registerec
username	String	**Required**.
Password	String	**Required**.
access_code	String	**Required**. Access code defined for each API binding with service.
return_type	Integer	**Required**. A. 1 use for JSON, b. 2 use for XML
ip_address	String	**Optional.**

Return Response

Code	String
200	[“full name BD”, ”full name EN”, “mother name”, “father name”, “dob”, “NID”, “email”, “mobile number”, “present address”, “permanent address”, “job info”, “profile pic”, ”designation name”, ”office name”]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide username."}
604	{"error":" Please provide password."}
625	{"error":" Incorrect username or password."}
606	{"error":" Return Type Missing."}
641	{"error":" Blocked User ID."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

4) Office Information

Method	URL
GET	{LISF BASE URL}/api/officeprofile/

Input Parameter

Name	Type	Description
service_id	String	**Required** The service ID you received from LISF when you registered
access_code	String	**Required** Access code defined for each API binding with service.
office_id	String	**Required**
office_name	String	**Optional**
office_address	String	**Optional**
return_type	Integer	**Required** A. 1 use for JSON, b. 2 use for XML
ip_address	String	**Optional**

Return Response

Code	String
200	[“Title BD”, “Title EN”, “Level”, “Parent Office Title”, “Address”, “Phone”, “Fax”, “Mobile”, “Email”, “Website”]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
606	{"error":" Return Type Missing."}
710	{"error":" Blocked Office."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

5) Office Hierarchy

Method	URL
GET	{LISF BASE URL}/api/officetree/

Input Parameter

Name	Type	Description	
service_id	String	Required	The service ID you received from LISF when you registerec
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
direction	Integer	Required	A: 1 use for Upward, B: 2 use for Downward
office_name	String	Optional	
office_address	String	Optional	
return_type	Integer	Required	A. 1 use for JSON, b. 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	Case 1, Upward - [{“id1 , title1”} {“id2 , title2”} {“given id , return title”}] Case 2, Downward - [{“given id , return title”} {“id1 , title1”} {“id2 , title2”}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
606	{"error":" Return Type Missing."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment - N/A

Service List

Every Service related to LISF can use this API.

6) Office Organogram

Method	URL
GET	{LISF BASE URL}/api/officeorganogram/

Input Parameter

Name	Type	Description	
service_id	String	Required	The service ID you received from LISF when you registered
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
office_name	String	Optional	
office_address	String	Optional	
return_type	Integer	Required	A. 1 use for JSON, b. 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	[{“id1”, “level1”, “Designation1”, “Number”}{“id2”, “level2”, “Designation2” ,“Number”}{“id3”, “level2”, “Designation2” , “Number”} {“id4”, “level3”,“Designation3” “Number”} . . .]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
606	{"error":" Return Type Missing."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

7) Office Organogram with User List

Method	URL
GET	{LISF BASE URL}/api/officeuserorganogram/

Input Parameter

Name	Type	Description	
service_id	String	Required	The service ID you received from LISF when you registered
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
office_name	String	Optional	
office_address	String	Optional	
return_type	Integer	Required	A. 1 use for JSON, b. 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	[{“id1”, “level1”, “name1”, “Designation1”}{“id2”, “level2” “name2”, “Designation2”}{“id3”, “level2”, “name3”, “Designation2”} {“id4”, “level3”, “name4”, “Des3”} . . .]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
606	{"error":" Return Type Missing."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

8) Office Branch/Department List

Method	URL
GET	{LISF BASE URL}/api/officebranchlist/

Input Parameter

Name	Type	Description	
service_id	String	Required	The service ID you received from LISF when you registered
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
office_name	String	Optional	
office_address	String	Optional	
return_type	Integer	Required	A. 1 use for JSON, b. 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	[{“id1”, “title1” } {“id2”, “title2”} {“id3”, “title3”} {“id4”, “title4”} . . .]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
606	{"error":" Return Type Missing."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

9) Office-Department wise User List

Method	URL
GET	{LISF BASE URL}/api/officebranchuserlist/

Input Parameter

Name	Type		Description
service_id	String	Required	The service ID you received from LISF when you registered.
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
office_department_id	String	Required	
office_name	String	Optional	
office_address	String	Optional	
return_type	Integer	Required	A. 1 use for JSON, b. 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	[{“id1”, “level1”, “name1”, “Designation1”} {“id2”, “level2”, “name2”, “Designation2”} {“id3”, “level3”, “name3” ,“Designation3”} . . .]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
606	{"error":" Please provide office department id."}
701	{"error":" Wrong Office ID."}
606	{"error":" Return Type Missing."}
710	{"error":" Blocked Office."}

Data Validity Duration

No warranty provided. It will be recommended that, before taking any important decision it should be mandatory to call login authentication API again.

Payment

N/A

Service List

Every Service related to LISF can use this API.

10)Office Number Count Based on Area & Status

Method	URL
GET	{LISF BASE URL}/api/areaofficecount/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
area_type	String	Required	A) 1 use for whole country B) 2 use for Division C) 3 use for Zone D) 4 use for District E) 5 use for Upazila F) 8 use for Union
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
area_code	String	Required.	Respective Area code based on type & code type
office_status	String	Required.	A) 1 use for ALL B) 2 use for Active C) 3 use for Blocked D) 4 use for In-Active
access_code	String	Required.	Access code defined for each API binding with service.
return_type	Integer	Required.	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{“area title”, “Number of Office”, “status”}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
801	{"error":" Area Type Missing."}
802	{"error":" Area Code Type Missing."}
803	{"error":" Area Code Missing."}
851	{"error":" Invalid Area Type."}
852	{"error":" Invalid Area Code Type."}
853	{"error":" Invalid Area Code."}
606	{"error":" Return Type Missing."}
710	{"error":" Blocked Office."}

Data Validity Duration

No warranty provided.

Service List

Every Service related to LISF can use this API.

B) RECORDS/Khatian/Porcha Related (View Only)

11) Generate Khatian [Formatted View]

Method	URL
GET	{LISF BASE URL}/api/khatiangenerate/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	Respective Area code based on code type
district_code	String	Required	Respective Area code based on code type
upazila_code	String	Required	Respective Area code based on code type
mouja_code	String	Required	Respective Area code based on code type
khatian_number	String	Required	
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200 If single khatian needs multiple page dataset come with multiple record set. And for various survey type data return type would b different set	/--------------- BRS RESPONSE --------------- \\ [{“Survey type: BRS”, “page width”, “page height”, “page number”, “page title”, “khatian number”, “tafsil/area Bar”, “table Number”, “table width”, “table height”, “table header” “column 1”,“ column 2”,“ column 3”,“ column 4”, “column 5”,“ column 6”,“ column 7”,“ column 8”,“ column 9”,“ column 10”,“ column 11”,“ column 12”,“table footer”, “additional info”, “payment info”}] /--------------- CS RESPONSE --------------- \\ [{“Survey type: CS”, “page width”, “page height”, “page number:1”, “page title”, “khatian number”, “tafsil/area Bar”, “table Number:1”, “table width”, “table height”, “table header”, “column 1”, “column 2”, “column 3”, “column 4”,

	“column 5”, “column 6”, “table footer”, “table Number:2”, “table width”, “table height”, “table header”, “column 1”, “column 2”, “column 3”, “column 4”, “column 5”, “table footer”, “page width”, “page height”, “page number:2”, “page title”, “table Number:3”, “table width”, “table height”, “table header”, “column 1”, “column 2”, “column 3”, “column 4”, “column 5”, “column 6”, “column 7”, “column 8”, “column 9”, “table footer”, “table Number:4”, “table width”, “table height”, “table header”, “column 1”, “column 2”, “table footer”, “additional info”, “payment info”}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
804	{"error":" Any Area Code is missing."}
911	{"error":" Khatian Number Missing."}
854	{"error":" Invalid/Missing Division Code."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
951	{"error":" Invalid Khatian Number."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
912	{"error":" Khatian Type Missing."}
964	{"error":" Invalid Khatian Type."}
962	{"error":" Khatian Not Found/stored in System."}
963	{"error":" Restricted Khatian, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

This information is provided to the recipient as-is.
For survey type data, LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Specific Service those need formatted khatian view such DC office Record Room can use this API.

12)Dag/Plot Information in Mouja

Method	URL
GET	{LISF BASE URL}/api/daginfomouja/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF LISF Area Code with documentation available in Portal
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
dag_number	String	Required	
dag_origin_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS G) 7 use for PETI
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Area”, “Land Type”, “Agricultural Type”, “Government Owned”, “Is Road”, “Is Wetland”, “Is Forest”, “Is Religious Type related”, “khatian Number List” - (array type data) “Khatian number wise Area amount” - (array type data) }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}

805	{"error":" Division Code Missing."}
806	{"error":" District Code Missing."}
807	{"error":" Upazila Code Missing."}
808	{"error":" Mouja Code Missing."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
866	{"error":" Invalid Upazila Code."}
867	{"error":" Invalid Mouja Code."}
921	{"error":" Dag Number Missing."}
971	{"error":" Invalid Dag Number."}
852	{"error":" Invalid Area Code Type."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
922	{"error":" Dag Origin Type Missing."}
972	{"error":" Invalid Dag Origin Type."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

This information is provided to the recipient as-is.
For survey type data, LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

For Mutated type data, LISF may ensure medium-lived tokens usually have a lifetime of about 30 days.

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Specific Service those need Dag Area information such as Mutation, Land lease or any related service

13) Dag/Plot Information in Khatian

Method	URL
GET	{LISF BASE URL}/api/khatiandaginfo/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Required	
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Dag Number List", "Dag Number Land Type List", "Total Area amount List", "Khatian Area Amount List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
955	{"error":"Invalid/Missing Khatian Number."}
964	{"error":" Invalid Khatian Type."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

14)Land Owner Information in Khatian

Method	URL
GET	{LISF BASE URL}/api/khatianownerinfo/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required****	**If code type BBS then it required otherwise optional**
mouja_code	String	**Required**	**Respective Area code based on code type**
khatian_number	String	**Required**	
khatian_type	String	**Required**	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	**Required**	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ "Land Owner Name List", "Land Owner Father/Husband Name List", "Land Owner Address List", "Land Owner Percentage List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
955	{"error":" Invalid/Missing Khatian Number."}
964	{"error":" Invalid Khatian Type."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

15) Land Owner List in Mouja

Method	URL
GET	{LISF BASE URL}/api/landownerlist/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Required**	If code type BBS then it required otherwise optional
mouja_code	String	Required	Respective Area code based on code type
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	Required***	If Survey Type Choose BRS/1 or CITY/4 then A) 1 use for Survey B) 2 use for Mutated Otherwise Default value 1, stands for Survey
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Land Owner Number”, “Total Khatian Number”, “Land Owner List (khatian Number)”- (array type data), “Land Owner Address List (khatian Number)”- (array type data)}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
921	{"error":" Dag Number Missing."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

16)Search Land Owner Name in Mouja

Method	URL
GET	{LISF BASE URL}/api/landownerinmouja/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LIS
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS G) 7 use for PETI
survey_sub_type	String	Required**	If Survey Type Choose BRS/1 or CITY/4 then A) 1 use for Survey B) 2 use for Mutated Otherwise Default value 1, stands for Survey
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
land_owner_name	String	Required	
owner_father_name	String	Optional	
owner_husband_name	String	Optional	
search_type	String	Required	A) 1 use for Strict - Every searching criteria strictly followed B) 2 use for Similar – similar word or phrase matchin
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Area of Owner (sum of all khatian)”, “Actual Area of Owner (sum of all valid khatian)”, “All Khatian Number List with Owner percentage & Area Amount”, “Valid Khatian Number List with Owner percentage & Area Amount” “Dag Number list with area amount of valid khatians” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}

612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
805	{"error":" Division Code Missing."}
806	{"error":" District Code Missing."}
807	{"error":" Upazila Code Missing."}
808	{"error":" Mouja Code Missing."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
866	{"error":" Invalid Upazila Code."}
867	{"error":" Invalid Mouja Code."}
931	{"error":" Owner Name Missing."}
981	{"error":" Owner Name Not Found."}
982	{"error":" Owner Name and Father/Husband Name Not Matched."}
852	{"error":" Invalid Area Code Type."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
922	{"error":" Khatian Type Missing."}
972	{"error":" Invalid Khatian Type."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

This information is provided to the recipient as-is.
For survey type data, LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

For Mutated type data, LISF may ensure medium-lived tokens usually have a lifetime of about 30 days.
For survey type data, LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Specific Service those need Dag Area information such as Mutation, Land lease or any related service

17)Khas/Government Owned Land Information in Mouja

Method	URL
GET	{LISF BASE URL}/api/khaslandmouja/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Required**	If code type BBS then it required otherwise optional
mouja_code	String	Required	Respective Area code based on code type
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Land Area Amount”, “Khatian Number List”, “Dag Number List (khatian Number)”, “Dag Number Area Amount (khatian Number)” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Specific Service those need Khas Area information such as Mutation, Land lease or any related service

18)Khatian Number Count in Mouja

Method	URL
GET	{LISF BASE URL}/api/khatiancountmouja/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Required**	If code type BBS then it required otherwise optional
mouja_code	String	Required	Respective Area code based on code type
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Total Khatian Number", "Khatian Number List (array)" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Specific Service those need to generate report or dashboard based on khatian

19)Khatian Number Count in Upazila

Method	URL
GET	{LISF BASE URL}/api/khatiancountupazila/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
survey_type	String	**Required**	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	**Required**	A) 1 use for Survey B) 2 use for Mutated
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required**	
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ “Total Khatian Number”, “Mouja List with Khatian Number” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Specific Service those need to generate report or dashboard based on khatian

20)Reference Khatian Number

Method	URL
GET	{LISF BASE URL}/api/referencekhatian/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Khatian Number”, “Khatian Number List with Type (array)” }
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
965	{"error":" Invalid/Missing Khatian Number."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure medium-lived tokens usually have a lifetime of about 20 days.

Service List

Specific Service those need to find reference khatian such Mutation Service.

21) Reference Dag Number

Method	URL
GET	{LISF BASE URL}/api/referencedag/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
dag_number	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Khatian Number List with Khatian Type & Details (array)” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
971	{"error":" Invalid Dag Number."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure medium-lived tokens usually have a lifetime of about 20 days.

Service List

Specific Service those need to find reference khatian such Mutation Service.

22)Khatian Tax Information

Method	URL
GET	{LISF BASE URL}/api/khatiantaxinfo/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "tax amount" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
961	{"error":" Invalid Khatian Number."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure medium-lived tokens usually have a lifetime of about 20 days.

Service List

Specific Service those need to find taxes of khatian such Mutation Service.

C) AREA & MAP RELATED (View Only)

23)Generate Division List

Method	URL
GET	{LISF BASE URL}/api/divisionlist/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF LISF area code with documentation can be found in Land portal.
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Division Code List”, “Division Name List (English)”, “Division Name List (Bangla)”}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

24)Generate District list

Method	URL
GET	{LISF BASE URL}/api/districtlist/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Required	If '-1' use as value of Division Code then it generates Whole countrywide district list.
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "District Code List", "District Name List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
805	{"error":" Division Code Missing."}
864	{"error":" Invalid Division Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

25)Generate Upazila List

Method	URL
GET	{LISF BASE URL}/api/upazilalist/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Upazila Code List", "Upazila Name List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
806	{"error":" District Code Missing."}
865	{"error":" Invalid District Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

26)Generate Mouja List

Method	URL
GET	{LISF BASE URL}/api/moujalist/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required**	
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ “Mouja Code List”, “Mouja Name List” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
866	{"error":" Invalid Upazila Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

27)Generate Union List

Method	URL
GET	{LISF BASE URL}/api/unionlist/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required**	
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ “Union Code List”, “Union Name List” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
866	{"error":" Invalid Upazila Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

28)Generate Pouroshova List

Method	URL
GET	{LISF BASE URL}/api/Pouroshova-list/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required**	
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ "Pouroshova Code List", "Pouroshova Name List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
866	{"error":" Invalid Upazila Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

29)Generate Pouroshova Ward List

Method	URL
GET	{LISF BASE URL}/api/Pouroshova-ward-list/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Required**	If code type BBS then it required otherwise optional
pouroshova_code	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Pouroshova Ward Code List", "Pouroshova Ward Name List" }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
866	{"error":" Invalid Upazila Code."}
812	{"error":" Pouroshova Code Missing."}
873	{"error":" Invalid Pouroshova Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those need to use area related operation may use this API

30)Generate City Corporation List

Method	URL
GET	{LISF BASE URL}/api/city-corporation-list/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “City Corporation Code List”, “City Corporation Name List” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
866	{"error":" Invalid Upazila Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

31) Generate City Corporation Ward List

Method	URL
GET	{LISF BASE URL}/api/city-corporation-ward-list/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Optional	
city_corporation_code	String	Required	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Pouroshova Code List”, “Pouroshova Name List” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
865	{"error":" Invalid District Code."}
807	{"error":" Upazila Code Missing."}
814	{"error":" City Corporation Code Missing."}
871	{"error":" Invalid City Corporation Code."}
866	{"error":" Invalid Upazila Code."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those need to use area related operation may use this API

32)Specific Area Information

Method	URL
GET	{LISF BASE URL}/api/areainfo/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_type	String	Required	A) 01 use for whole country B) 02 use for Division C) 03 use for Zone D) 04 use for District E) 05 use for Upazila F) 06 use for Mouja G) 07 use for Thana H) 08 use for Union I) 09 use for City Corporation J) 10 use for City Corporation Ward K) 12 use for Pouroshova Ward L) 22 Use for Police Station M) 13 use for Union Ward N) 14 use for Subah O) 15 use for Pargana P) 16 use for Taluk Q) 17 use for Touji R) 18 use for Mohokuma S) 19 use for Subdivision T) 20 use for Province U) 21 use for Jamindari V) 23 use for GET-office W) 24 use for Village X) 25 use for Restricted Area
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF LISF area code with documentation can be found in Land portal.
division_code	String	Required**	If area type division then it's required
zone_code	String	Required**	If area type Zone otherwise optional/No Use
district_code	String	Required**	If area type District OR area code type BBS & below order require then this field mandatory otherwise optional/No Use
thana_code	String	Required**	If area type Thana OR area code type BBS & below order required then this field mandatory otherwise optional/No Use.
upazila_code	String	Required**	If area type Upazila OR area code type BBS & below order required then this field mandatory otherwise optional/No Use.
mouja_code	String	Required**	If area type Mouja then it's required otherwise optional/No Use
union_code	String	Required**	If area type union then it's required otherwise optional/No Use
city_corporation_code	String	Required**	If area type city corporation or City Corporation ward then it's required otherwise optional/No Use
pouroshova_code	String	Required**	If area Pouroshova or Pouroshova ward then it's required otherwise optional/No Use
area_code	String	Required**	If Not specified area type listed above, then this field will be used, e.g.- Post-Office, Police Station, Jamindari etc.
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional	

Return Response

Code	String
200	[{ “area code”, “area Name Bangla”, “area Name English”, “change History exist”, “last Update Timestamp” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
801	{"error":" Area Type Missing."}
851	{"error":" Invalid Area Type."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
845	{"error":" Invalid Zone Code."}
865	{"error":" Invalid District Code."}
866	{"error":" Invalid Upazila Code."}
867	{"error":" Invalid Mouja Code."}
869	{"error":" Invalid Thana Code."}
868	{"error":" Invalid Union Code."}
871	{"error":" Invalid City Corporation Code."}
872	{"error":" Invalid City Corporation Ward Code."}
873	{"error":" Invalid Pouroshova Code."}
874	{"error":" Invalid Pouroshova Ward Code."}
875	{"error":" Invalid Touji Code."}
876	{"error":" Invalid Pargana Code."}
803	{"error":" Area Code Missing."}
853	{"error":" Invalid Area Code."}
609	{"error":" Missing at least One Required Parameter."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Payment

N/A

Service List

Services those need to use area related operation may use this API

33) Specific Area Change Information History

Method	URL
GET	{LISF BASE URL}/api/areachangeinfo/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_type	String	**Required**	A) 01 use for whole country B) 02 use for Division C) 03 use for Zone D) 04 use for District E) 05 use for Upazila F) 06 use for Mouja G) 07 use for Thana H) 08 use for Union I) 09 use for City Corporation J) 10 use for City Corporation Ward K) 12 use for Pouroshova Ward L) 22 Use for Police Station M) 13 use for Union Ward N) 14 use for Subah O) 15 use for Pargana P) 16 use for Taluk Q) 17 use for Touji R) 18 use for Mohokuma S) 19 use for Subdivision T) 20 use for Province U) 21 use for Jamindari V) 23 use for Post-office W) 24 use for Village X) 25 use for Restricted Area
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF LISF area code with documentation can be found in Land portal.
division_code	String	**Required****	If area type division then it's required
zone_code	String	**Required****	If area type Zone otherwise optional/No Use
district_code	String	**Required****	If area type District OR area code type BBS & below order require then this field mandatory otherwise optional/No Use
thana_code	String	**Required****	If area type Thana OR area code type BBS & below order required then this field mandatory otherwise optional/No Use.
upazila_code	String	**Required****	If area type Upazila OR area code type BBS & below order required then this field mandatory otherwise optional/No Use.
mouja_code	String	**Required****	If area type Mouja then it's required otherwise optional/No Use
union_code	String	**Required****	If area type union then it's required otherwise optional/No Use
city_corporation_code	String	**Required****	If area type city corporation or City Corporation ward then it's required otherwise optional/No Use
pouroshova_code	String	**Required****	If area Pouroshova or Pouroshova ward then it's required otherwise optional/No Use
area_code	String	**Required****	If Not specified area type listed above, then this field will be used, e.g.- Post-Office, Police Station, Jamindari etc.
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional**	

Return Response

Code	String
200	[{ “Previous Area Type”, “Previous Area Code”, “Previous Area Name Bangla”, “Previous Area Name English”, “Current Area Type”, “Current Area Code”, “Current area Name Bangla”, “Current Area Name English”, “Change Date Time”, “Affected Number of Area Element” }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
801	{"error":" Area Type Missing."}
851	{"error":" Invalid Area Type."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
864	{"error":" Invalid Division Code."}
845	{"error":" Invalid Zone Code."}
865	{"error":" Invalid District Code."}
866	{"error":" Invalid Upazila Code."}
867	{"error":" Invalid Mouja Code."}
869	{"error":" Invalid Thana Code."}
868	{"error":" Invalid Union Code."}
871	{"error":" Invalid City Corporation Code."}
872	{"error":" Invalid City Corporation Ward Code."}
873	{"error":" Invalid Pouroshova Code."}
874	{"error":" Invalid Pouroshova Ward Code."}
875	{"error":" Invalid Touji Code."}
876	{"error":" Invalid Pargana Code."}
803	{"error":" Area Code Missing."}
853	{"error":" Invalid Area Code."}
609	{"error":" Missing at least One Required Parameter."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those need to use area related operation may use this API

34)Plot Draw in MAP (2D)

Method	URL
GET	{LISF BASE URL}/api/plotdraw2dmap/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
level_of_detail	String	**Optional**	Value between 0 and 3, where 0 specifies the coarsest level of boundary detail and 3 specifies the best.
getAllPolygons	Boolean	**Optional**	A) 1 use for All Polygon B) 0 use for only main outline
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required****	**If code type BBS then it required otherwise optional**
mouja_code	String	**Required**	**Respective Area code based on code type**
khatian_number	String	**Required**	
khatian_type	String	**Required**	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	**Required**	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional**	
center_postion	Integer	**Required**	{latitude: 24.886, longitude: -70.268}
boundary_type	String	**Required**	A) 1 use for Rectangle B) 2 use for Polygon C) 3 use for Circle D) 4 use for Line
boundary_coords	String	**Required**	{ latitude: 25.774, longitude: -80.190 }, { latitude: 18.466, longitude: -66.118 }, { latitude: 32.321, longitude: -64.757 }, { latitude: 32.321, longitude: -64.757 }
zoom_level	Integer	**Required**	Default value 5
show_marker	Integer	**Optional**	

Return Response

Code	String
200	**[{ “Map URL Link” }]**
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
955	{"error":" Invalid/Missing Survey Type."}

912	{"error":" Khatian Type Missing."}
964	{"error":" Invalid Khatian Type."}
854	{"error":" Invalid/Missing Division Code."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
965	{"error":" Invalid/Missing Khatian Number."}
1201	{"error":" 2D Map Center Position Coordinates Missing."}
1202	{"error":" 2D Map Area Boundary Type Missing."}
1203	{"error":" 2D Map Boundary Coordinates Missing."}
1221	{"error":" Invalid Center Position Coordinates (2D MAP)."}
1222	{"error":" Center Position is outside of Khatian Area (2D MAP)."}
1223	{"error":" Boundary Coordinates are outside of Khatian Area."}
1224	{"error":" Boundary Type and Number of Boundary Coordinates mismatch."}
606	{"error":" Return Type Missing."}

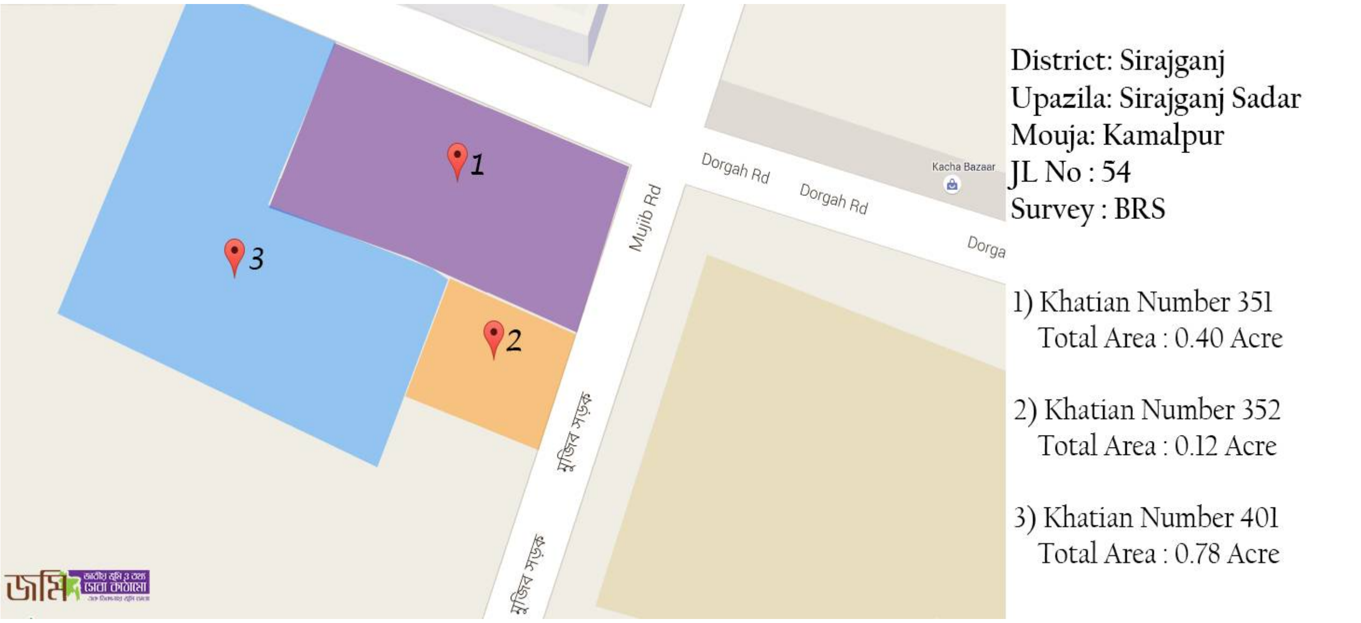
Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those need to use MAP Plotting Khatian such Mutation etc.

Sample Output of MAP



District: Sirajganj
Upazila: Sirajganj Sadar
Mouja: Kamalpur
JL No : 54
Survey : BRS

1) Khatian Number 351
Total Area : 0.40 Acre

2) Khatian Number 352
Total Area : 0.12 Acre

3) Khatian Number 401
Total Area : 0.78 Acre

D) Deeds Related (View Only)

35) Deed Information Based on Khatian Number

Method	URL
GET	{LISF BASE URL}/api/deedinfobykhatian/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS then it required otherwise optional
upazila_code	String	Required**	If code type BBS then it required otherwise optional
mouja_code	String	Required	Respective Area code based on code type
khatian_number	String	Required	
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Deed No”, “Deed Type”, “Reference Deed No”, “Balam Book No”, “Tafsil Information”, “Owner Information”, “Seller Information”, “Deed Date”, “Office Information”, “Scanned Deed Link” }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
850	{"error":" Invalid/Missing Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
955	{"error":" Invalid/Missing Survey Type."}
966	{"error":" Invalid/Missing Khatian Type."}
965	{"error":" Invalid/Missing Khatian Number."}
973	{"error":" Restricted Information, Can't deliverable."}
1150	{"error":" Deed's Not found or Stored in LISF."}
606	{"error":" Return Type Missing."}

36)Deed Information Based on Dag/Plot Number

Method	URL
POST	{LISF BASE URL}/api/deedinfobydag/

Input Parameter

Name	Type	Description	
service_id	String	**Required**	You received from LISF when registered your service.
access_code	String	**Required**	Access code defined for each API binding with service.
area_code_type	String	**Required**	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	**Optional**	
district_code	String	**Required****	**If code type BBS then it required otherwise optional**
upazila_code	String	**Required****	**If code type BBS then it required otherwise optional**
mouja_code	String	**Required**	**Respective Area code based on code type**
dag_number	String	**Required**	
dag_origin_type	String	**Required**	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	**Required**	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
return_type	Integer	**Required**	A) 1 use for JSON B) 2 use for XML
ip_address	String	**Optional.**	

Return Response

Code	String
200	[{ "Deed No", "Deed Type", "Reference Deed No", "Balam Book No", "Tafsil Information", "Owner Information", "Seller Information", "Deed_date", "Office Information", "Scanned Deed Link" }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
850	{"error":" Invalid/Missing Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
955	{"error":" Invalid/Missing Survey Type."}
966	{"error":" Invalid/Missing Dag Origin Type."}
975	{"error":" Invalid/Missing Dag Number."}
973	{"error":" Restricted Information, Can't deliverable."}
1150	{"error":" Deed's Not found or Stored in LISF."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

37)Deed Information Based on Owner Name

Method	URL
GET	{LISF BASE URL}/api/deedinfobyowner/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
deed_type	String	Optional	
owner_name	String	Required	
owner_mother_name	String	Optional	
owner_father_name	String	Optional	
date_of_birth	Date	Optional	
profession	String	Optional	
religion	String	Optional	
national_id	Integer	Optional	
address	String	Optional	
deed_date	Date	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Deed No”, “Deed Type”, “Reference Deed No”, “Balam Book No”, “Tafsil Information”, “Owner Information”, “Seller Information”, “Deed Date”, “Office Information”, “Scanned Deed Link” }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
931	{"error":" Name Missing."}
981	{"error":" Provided Name Not Found."}
982	{"error":" Provided Name does not match with Father/Husband Name."}
983	{"error":" Provided Name does not match with Mother Name."}
984	{"error":" Provided Name does not match with National ID."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

38) Deed information Based on Seller Name

Method	URL
GET	{LISF BASE URL}/api/deedinfobyseller/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
deed_type	String	Optional	
seller_name	String	Required	
seller_mother_name	String	Optional	
seller_father_name	String	Optional	
date_of_birth	Date	Optional	
profession	String	Optional	
religion	String	Optional	
national_id	Integer	Optional	
address	String	Optional	
deed_date	Date	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Deed No”, “Deed Type”, “Reference Deed No”, “Balam Book No”, “Tafsil Information”, “Owner Information”, “Seller Information”, “Deed Date”, “Office Information”, “Scanned Deed Link” }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
931	{"error":" Name Missing."}
981	{"error":" Provided Name Not Found."}
982	{"error":" Provided Name does not match with Father/Husband Name."}
983	{"error":" Provided Name does not match with Mother Name."}
984	{"error":" Provided Name does not match with National ID."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

39) Deed Information Base on Deed Id

Method	URL
GET	{LISF BASE URL}/api/deedinfobydeedid/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
deed_id	String	Required	
owner_ name	String	Optional	
owner_father_name	String	Optional	
owner_date_of_birth	Date	Optional	
owner_profession	String	Optional	
owner_religion	String	Optional	
owner_national_id	Integer	Optional	
owner_address	String	Optional	
deed_date	Date	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Deed No", "Deed Type", "Reference Deed No", "Balam Book No", "Tafsil Information", "Owner Information", "Seller Information", "witness Information", "Stamp Fee", "Deed Date", "Office Information", "Scanned Deed Link"}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
1101	{"error":" Deed ID Missing."}
1111	{"error":" Invalid Deed Id."}
1121	{"error":" Owner Name and Deed ID Not Matched."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

40)Deeds Information (data list) Base on Area

Method	URL
GET	{LISF BASE URL}/api/deedinfobyarea/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	If code type BBS / Searching level District base then it required otherwise optional
upazila_code	String	Required**	If code type BBS / Searching level Upazila base then it required otherwise optional
mouja_code	String	Required**	If code type BBS / Searching level Mouja base then it required otherwise optional
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Deed No”, “Deed Type”, “Reference Deed No”, “Balam Book No”, “Tafsil Information”, “Owner Information”, “Seller Information”, “Deed_date”, “Office Information”, “Scanned Deed Link” }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
850	{"error":" Invalid/Missing Area Code Type."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
973	{"error":" Restricted Information, Can't deliverable."}
1150	{"error":" Deed's Not found or Stored in LISF."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those are using in Sub-Registry Office may use this.

41) Deeds Information (data list) Base on Office

Method	URL
GET	{LISF BASE URL}/api/deedinfobyoffice/

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
office_name	String	Optional	
office_address	String	Optional	
start_date	Date	Optional	
end_date	Date	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Deed No”, “Deed Type”, “Reference Deed No”, “Balam Book No”, “Tafsil Information”, “Owner Information”, “Seller Information”, “Deed_date”, “Office Information”, “Scanned Deed Link” }, ...]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code"}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
973	{"error":" Restricted Information, Can't deliverable."}
1150	{"error":" Deed's Not found or Stored in LISF."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

Service List

Services those are using in Sub-Registry Office may use this.

E) Lease Type Records Related (View Only)

42)Generate List of lease property (AP Type)

Method	URL
GET	{LISF BASE URL}/api/ list-ap-land-lease /

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_type	String	Required	A) 04 for District B) 05 for Upazila C) 06 for Mouja
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
year	String	Required	Use 4 digit numeric year, such ‘2014’
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Total Lease Owner Number”, “Total Leased Area”, “Lease Owner’s ID & Name List (dag Number)”- (array type data), “Lease Owner’s Address List (dag Number)” - (array type data), “Lease Owner’s Area amount” – (array type data), “Leased Area Value (taka)” - (array type data)}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
1320	{"error":" Lease Year field Missing."}
1321	{"error":" Invalid Lease Year."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

43)Generate List of Lease Property (VP Type)

Method	URL
GET	{LISF BASE URL}/api/ list-vp-land-lease /

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_type	String	Required	A) 04 for District B) 05 for Upazila C) 06 for Mouja
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
year	String	Required	Use 4 digit numeric year, such '2014'
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ "Total Lease Owner Number", "Total Leased Area", "Lease Owner's ID & Name List (dag Number)"- (array type data), "Lease Owner's Address List (dag Number)" - (array type data), "Lease Owner's Area amount" – (array type data), "Leased Area Value (taka)" - (array type data)}]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
1320	{"error":" Lease Year field Missing."}
1321	{"error":" Invalid Lease Year."}
973	{"error":" Restricted Information, Can't deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

44)Generate List of Expired Lease Owners

Method	URL
GET	{LISF BASE URL}/api/ list-expired-land-lease /

Input Parameter

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_type	String	Required	A) 04 for District B) 05 for Upazila C) 06 for Mouja
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
division_code	String	Optional	
district_code	String	Required**	**If code type BBS then it required otherwise optional**
upazila_code	String	Required**	**If code type BBS then it required otherwise optional**
mouja_code	String	Required	**Respective Area code based on code type**
time_limit	String	Required	Use understandable syntax – 7 days/ 2 month/ 80 days
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML
ip_address	String	Optional.	

Return Response

Code	String
200	[{ “Lease Owner Number”, “Leased Area”, “Lease Owner’s ID & Name List (dag Number)”- (array type data), “Lease Owner’s Address List (dag Number)” - (array type data), “Lease Owner’s Area amount” – (array type data), “Leased Area Value (taka)” - (array type data) }]
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
802	{"error":" Area Code Type Missing."}
804	{"error":" Division/District/Upazila/Mouja Code Missing."}
870	{"error":" Invalid Division/District/Upazila/Mouja Code."}
1322	{"error":" Time Limit field Missing."}
1323	{"error":" Invalid Time Limit."}
973	{"error":" Restricted Information, Can’t deliverable."}
606	{"error":" Return Type Missing."}

Data Validity Duration

LISF may ensure long-lived tokens usually have a lifetime of about 60 days.

F) RECORDS/Khatian/Porcha Related (Write Request)

45) Write New Khatian

Method	URL
POST	{LISF BASE URL}/api/write-new-khatian/

Input Parameter – Phase 1

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
district_code	String	Required	**Respective Area code based on code type**
upazila_code	String	Required	**Respective Area code based on code type**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Required	
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
office_id	String	Required	
khatian_owner_type	String	Required	A) 1 use for Government B) 2 use for Private
owner_number	Integer	Required	Total Number of Owner in That Khatian/Porcha
owner_name	String	Required	If multiple owner, use comma as separator for each data
owner_address	String	Required	If multiple owner, use comma as separator for each data
owner_percentage	Double	Required	If multiple owner, use comma as separator for each data
khatian_tax	Double	Required	
total_dag_number	String	Required	Total Number of Dag/Plot Number in That Khatian/Porcha
dag_number	String	Required	If multiple dag/plot, use comma as separator for each data
dag_area_total	String	Required	If multiple dag/plot, use comma as separator for each data
dag_area_khotian	String	Required	If multiple dag/plot, use comma as separator for each data
dag_land_type	String	Required	If multiple dag/plot, use comma as separator for each data
dag_remarks	String	Required	If multiple dag/plot, use comma as separator for each data
reference_dag	String	Required	If no reference available, use 0 as default
ref_khatian_number	String	Required	If no reference available, use 0 as default
entry_user_info	String	Required	User Id + Full Name + Designation + Office ID *
approval_user_info	String	Required	User Id + Full Name + Designation + Office ID **
khatian_date_time	Date	Required	Entry + Approval
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML

Input Parameter – Phase 2

received_access_token	String	Required	Received After Phase-1 submission of data

confirmation_instruction	Integer	Required	User Final Response, 1 use for True & 0 use for False
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.

Return Response – Phase 1

Code	**String**
201	**{“random short durational access token”, “allowed port number”}**
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
804	{"error":" Any Area Code is missing."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
912	{"error":" Khatian Type Missing."}
964	{"error":" Invalid Khatian Type."}
911	{"error":" Khatian Number Missing."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
930	{"error":" Khatian Owner Type Missing."}
967	{"error":" Same Khatian number exists for same mouja & khatian type."}
941	{"error":" This dag {dag_number} area can't bigger than total area."}
942	{"error":" This dag {dag_number} can't be zero or null."}
943	{"error":" This dag {dag_number} doesn’t enough area to use here."}
944	{"error":" This dag {dag_number} not permitted to use here."}
945	{"error":" This dag {dag_number} must have valid land type."}
946	{"error":" Total Dag Number & supplied dag data mismatch."}
947	{"error":" Khatian owner name can’t be blank or null."}
948	{"error":" Khatian owner name must have an address."}
949	{"error":" Total owner number & supplied owner name mismatch."}
935	{"error":" Invalid/Missing Khatian Entry/approve Date-time."}
936	{"error":" Invalid/Missing Khatian Entry/approve User Info."}
937	{"error":" Khatian Approval User must be LISF Register User."}
606	{"error":" Return Type Missing."}

Return Response – Phase 2

202	**{“successful transaction number return”, “date time” }**
1404	{"error":" Issued Token & supplied token doesn't match."}

1410	{"error":" Transactional Time Limit Expire."}
1420	{"error":" Invalid Port Use for confirmation communication."}
1430	{"error":" Previous service id, access code doesn't match current info."}

Extra Information

Whenever any user information uses in creating khatian, entry user* may not be LISF System user or his/her information may not save in LISF core database.

But Approval user must have to store in LISF core database and only this Request will be initiated by LISF core user.**

This type of API Method Call must use HTTPS or secure connection, otherwise No connection established or even not any error code provide by LISF.

In phase 1 return LISF restricted Dynamic Port number for further communication, In that case second time communication must use that allowed PORT number.

In Phase 2 return any ‘error code’, all communication must start from Beginning, not after phase 1.

Data Validity Duration

N/A
Time limit used for write confirmation.

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Mutation & Joma-Vag,
Porcha/Records Correction instructed by Court,
Miss case Management,
Correction of khatian clerical mistake and several other
Land Related Services may use this API method call to update LISF Core Database.

46)Update Khatian

Method	URL
POST	{LISF BASE URL}/api/update-khatian/

Input Parameter – Phase 1

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
district_code	String	Required	**Respective Area code based on code type**
upazila_code	String	Required	**Respective Area code based on code type**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Required	
khatian_type	String	Required	A) 1 use for Survey B) 2 use for Mutated
survey_type	String	Required	A) 1 use for BRS B) 2 use for CS C) 3 use for SA D) 4 use for CITY E) 5 use for DIARA F) 6 use for RS
office_id	String	Required	
khatian_owner_type	String	Required	A) 1 use for Government B) 2 use for Private
owner_number	Integer	Required	Total Number of Owner in That Khatian/Porcha
owner_name	String	Required	If multiple owner, use comma as separator for each data
owner_address	String	Required	If multiple owner, use comma as separator for each data
owner_percentage	Double	Required	If multiple owner, use comma as separator for each data
khatian_tax	Double	Required	
total_dag_number	String	Required	Total Number of Dag/Plot Number in That Khatian/Porcha
dag_number	String	Required	If multiple dag/plot, use comma as separator for each data
dag_area_total	String	Required	If multiple dag/plot, use comma as separator for each data
dag_area_khotian	String	Required	If multiple dag/plot, use comma as separator for each data
dag_land_type	String	Required	If multiple dag/plot, use comma as separator for each data
dag_remarks	String	Required	If multiple dag/plot, use comma as separator for each data
reference_dag	String	Required	If no reference available, use 0 as default
ref_khatian_number	String	Required	If no reference available, use 0 as default
update_user_info	String	Required	User Id + Full Name + Designation + Office ID *
approval_user_info	String	Required	User Id + Full Name + Designation + Office ID **
khatian_date_time	Date	Required	Update + Approval
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML

Input Parameter – Phase 2

received_access_token	String	Required	Received After Phase-1 submission of data
confirmation_instruction	Integer	Required	User Final Response, 1 use for True & 0 use for False
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.

Return Response – Phase 1

Code	String
201	{“random short durational access token”, “allowed port number”}
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}
605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
901	{"error":" Survey Type Missing."}
951	{"error":" Invalid Survey Type."}
912	{"error":" Khatian Type Missing."}
964	{"error":" Invalid Khatian Type."}
911	{"error":" Khatian Number Missing."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
930	{"error":" Khatian Owner Type Missing."}
967	{"error":" Khatian number doesn't exist for same mouja & khatian type."}
941	{"error":" This dag {dag_number} area can't bigger than total area."}
942	{"error":" This dag {dag_number} can't be zero or null."}
943	{"error":" This dag {dag_number} doesn’t enough area to use here."}
944	{"error":" This dag {dag_number} not permitted to use here."}
945	{"error":" This dag {dag_number} must have valid land type."}
946	{"error":" Total Dag Number & supplied dag data mismatch."}
947	{"error":" Khatian owner name can’t be blank or null."}
948	{"error":" Khatian owner name must have an address."}
949	{"error":" Total owner number & supplied owner name mismatch."}
950	{"error":" Total owner percentage can't cross 100 %."}
938	{"error":" Invalid/Missing Khatian Update/Approve Date-time."}
939	{"error":" Invalid/Missing Khatian Update/Approve User Info."}
937	{"error":" Khatian Approval User must be LISF Register User."}
606	{"error":" Return Type Missing."}

Return Response – Phase 2

202	{“successful transaction number return”, “date time” }
1404	{"error":" Issued Token & supplied token doesn't match."}
1410	{"error":" Transactional Time Limit Expire."}
1420	{"error":" Invalid Port Use for confirmation communication."}
1430	{"error":" Previous service id, access code doesn't match current info."}

```
1450     {"error":" Exist Khatian & Submitted Data matched 100% - No update."}
```

Extra Information

Whenever any user information uses in updating khatian or correcting khatian record information, update user* may not be in LISF System user or His/her information may not save in LISF core database.

But Approval user must have to store in LISF core database and only this Request will be initiated by LISF core user.**

This type of API Method Call must use HTTPS or secure connection, otherwise No connection established or even not any error code provide by LISF.

In phase 1 return LISF restricted Dynamic Port number for further communication, In that case second time communication must use that allowed PORT number.

In Phase 2 return any ‘error code’, all communication must start from Beginning, not after phase 1.

Data Validity Duration

N/A
Time limit used for update confirmation.

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Mutation & Joma-Vag,
Porcha/Records Correction instructed by Court,
Miss case Management,
Correction of khatian clerical mistake and several other
Land Related Services may use this API method call to update LISF Core Database.

47) New Lease Owner Entry

Method	URL
POST	{LISF BASE URL}/api/new-lease-owner/

Input Parameter – Phase 1

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
district_code	String	Optional	Respective Area code based on code type
upazila_code	String	Optional	Respective Area code based on code type
mouja_code	String	Required	Respective Area code based on code type
khatian_number	String	Optional	
dag_number	String	Required	If multiple dag, use comma as separator for each data
lease_area	Double	Required	If multiple dag, use comma as separator for each data
lease_value	Double	Required	
lease_type	String	Required	A) 1 use for AP B) 2 use for VP C) 3 use for Chandina-Viti
lease_owner_name	String	Required	If multiple owner, use comma as separator for each data
lease_owner_address	String	Required	If multiple owner, use comma as separator for each data
lease_owner_percent	Double	Optional	If multiple owner, use comma as separator for each data
start_date	Date	Required	
end_date	Date	Required	
lease_remarks	String	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML

Input Parameter – Phase 2

received_access_token	String	Required	Received After Phase-1 submission of data
confirmation_instruction	Integer	Required	User Final Response, 1 use for True & 0 use for False
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.

Return Response – Phase 1

Code	String
201	{"random short durational access token", "allowed port number"}
601	{"error":" Service id is missing."}
611	{"error":" Invalid Service Id."}
602	{"error":" Access Code is missing."}
612	{"error":" Invalid Access Code."}

605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
975	{"error":" Invalid/Missing Dag Number."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
1316	{"error":" Invalid/Missing Lease Type."}
1301	{"error":" Leased dag {dag_number} area can't bigger than total area."}
1302	{"error":" Leased dag {dag_number} can't be zero or null."}
943	{"error":" This dag {dag_number} doesn’t have enough area to use here."}
944	{"error":" This dag {dag_number} not permitted to use here."}
1311	{"error":" Lease owner name can’t be blank or null."}
1312	{"error":" Lease owner name must have an address."}
1315	{"error":" Invalid Lease start/end date."}
606	{"error":" Return Type Missing."}

Return Response – Phase 2

200	**{“successful transaction number return”, “date time” }**
1404	{"error":" Issued Token & supplied token doesn't match."}
1410	{"error":" Transactional Time Limit Expire."}
1420	{"error":" Invalid Port Use for confirmation communication."}
1430	{"error":" Previous service id, access code doesn't match current info."}

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Chandina Viti,
Acquired Property Lease management,
Vested Property Lease management
And Other lease Related Services may use this API method call to update LISF Core Database.

48) Update Lease Owner Entry

Method	URL
POST	{LISF BASE URL}/api/update-lease-owner/

Input Parameter – Phase 1

Name	Type	Description	
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.
office_id	String	Required	
area_code_type	String	Required	A) 1 use for BBS B) 2 use for DLRS C) 3 use for LISF
district_code	String	Required	**Respective Area code based on code type**
upazila_code	String	Required	**Respective Area code based on code type**
mouja_code	String	Required	**Respective Area code based on code type**
khatian_number	String	Optional	
dag_number	String	Required	If multiple dag, use comma as separator for each data
lease_area	Double	Required	If multiple dag, use comma as separator for each data
lease_value	Double	Required	
lease_type	String	Required	A) 1 use for AP B) 2 use for VP C) 3 use for Chandina-Viti
lease_owner_name	String	Required	If multiple owner, use comma as separator for each data
lease_owner_address	String	Required	If multiple owner, use comma as separator for each data
lease_owner_percent	Double	Required	If multiple owner, use comma as separator for each data
start_date	Date	Required	
end_date	Date	Required	
lease_remarks	String	Optional	
return_type	Integer	Required	A) 1 use for JSON B) 2 use for XML

Input Parameter – Phase 2

received_access_token	String	Required	Received After Phase-1 submission of data
confirmation_instruction	Integer	Required	User Final Response, 1 use for True & 0 use for False
service_id	String	Required	You received from LISF when registered your service.
access_code	String	Required	Access code defined for each API binding with service.

Return Response – Phase 1

Code	String
201	`{“random short durational access token”, “allowed port number”}`
601	`{"error":" Service id is missing."}`
611	`{"error":" Invalid Service Id."}`
602	`{"error":" Access Code is missing."}`
612	`{"error":" Invalid Access Code."}`

605	{"error":" Please provide office id."}
701	{"error":" Wrong Office ID."}
710	{"error":" Blocked Office."}
802	{"error":" Area Code Type Missing."}
852	{"error":" Invalid Area Code Type."}
975	{"error":" Invalid/Missing Dag Number."}
855	{"error":" Invalid/Missing District Code."}
856	{"error":" Invalid/Missing Upazila Code."}
857	{"error":" Invalid/Missing Mouja Code."}
1316	{"error":" Invalid/Missing Lease Type."}
1301	{"error":" Leased dag {dag_number} area can't bigger than total area."}
1302	{"error":" Leased dag {dag_number} can't be zero or null."}
943	{"error":" This dag {dag_number} doesn’t have enough area to use here."}
944	{"error":" This dag {dag_number} not permitted to use here."}
1311	{"error":" Lease owner name can’t be blank or null."}
1312	{"error":" Lease owner name must have an address."}
1315	{"error":" Invalid Lease start/end date."}
606	{"error":" Return Type Missing."}

Return Response – Phase 2

200	**{“successful transaction number return”, “date time” }**
1404	{"error":" Issued Token & supplied token doesn't match."}
1410	{"error":" Transactional Time Limit Expire."}
1420	{"error":" Invalid Port Use for confirmation communication."}
1430	{"error":" Previous service id, access code doesn't match current info."}
1350	{"error":" Exist Entry & Submitted Data matched 100% - No update.."}

Payment

Can Be implemented in Next phase with introducing various payment methods just initializing by transaction pin/code with same request.

Service List

Chandina Viti,
Acquired Property Lease management,
Vested Property Lease management
And Other lease Related Services may use this API method call to update LISF Core Database.

Usage Example

Every service in LISF requires a service ID and access code for the data transmitted in queries. Service ID and access code should be sent as HTTP headers "service_id" and "access_code".

Access Code is a HMAC-SHA256 encoded message. The HMAC-SHA256 code must be generated using a secret key that was generated with your Service Code. This code must be converted to its hexadecimal representation and uppercase characters.

CURL

```
curl  -w "\n%%{http_code}"\
      -H "service_id: gJx7Wa7qXkPtmTAaK3ADCtr6m5rCYYMy"\
      -H "access_code: EA4AB0289512C3A65092321F9BF25058B841"\
      --data "username=shawon&password=123456&return_type=1"\
      https://api.land.gov.bd/live/usercheck
```

PHP

PHP POST request

```
<?php
$url = "https://api.land.gov.bd/live/newleaseowner";
$service_id = "gJx7Wa7qXkPtmTAaK3ADCtr6m5rCYYMy";
$access_code = "EA4AB0289512C3A65092321F9BF25058B841";
$params = array(
    'office_id'=> '4532423432',
    'area_code_type'=> '3',
    'mouja_code'=> '123456',
    'dag_number'=> '22',
    'lease_area'=> '0.080',
    'lease_value'=> '25000.00',
    'lease_type'=> '1',
    'lease_owner_name'=> 'Abul Monem Chowdhury',
    'lease_owner_address'=> '25/1 New village, district road',
    'return_type'=>1
);
ksort($params);
$postFields = http_build_query($params, '', '&');
$signature = strtoupper(hash_hmac('sha256', $postFields, $secretKey
));
$headers = array(
    "service_id: $service_id",
    "access_code: $access_code"  );
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, 'POST');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postFields);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$response = curl_exec($ch);
$statusCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
if ($statusCode ==201) {
    $params_second = array(
             'new_access_token'=> '5dsfs532423432',
             'confirmation_message'=> '1'
    );
    ksort($params);
    $postFields2nd = http_build_query($params, '', '&');
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_POST, 'POST');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $postFields2nd);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    $response = curl_exec($ch);
    $statusCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    if ($statusCode !=200) {
        echo "Status code: $statusCode, response: $response";
    }
    var_dump(json_decode($response));
}
var_dump(json_decode($response));
```

PHP GET request

```
<?php
$url = "https://api.land.gov.bd/live/usercheck";
$service_id = "gJx7Wa7qXkPtmTAaK3ADCtr6m5rCYYMy";
$access_code = "EA4AB0289512C3A65092321F9BF25058B841";
$params = array(
    'username'=> 'shawon',
    'password'=> '123456',
    'return_type'=>1
);
ksort($params);
$fields = http_build_query($params, '', '&');
$signature = strtoupper(hash_hmac('sha256', $fields, $secretKey));
$headers = array(
    "service_id: $service_id",
    "access_code: $access_code"
);
```

```
$ch = curl_init($url."?".http_build_query($fields, '', '&'));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
$response = curl_exec($ch);
$statusCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

if ($statusCode!=200) {
    echo "Status code: $statusCode, response: $response";
}
var_dump(json_decode($response));
```

JAVA

POST request

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.TreeMap;
import java.util.Map;

public class ExtAPIExample {
    public static final java.lang.String HMAC_SHA256_ALGORITHM = "HmacSHA256";
    public static final java.lang.String UNICODE_CODE = "UTF-8";
    public static final String URL = "https://api.land.gov.bd/live/newleaseowner";
    // your Service ID
    public static final String serviceId = "YYYYYYYYYYYYYYYYYYYYYYY
YYYYYY";
    // your Access Code
    public static final String accessCode = "XXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX";

    public static void main(String[] args) {
        try {
            Map<String, String> postData = new TreeMap<>();
            postData.put ("office_id", "4532423432");
            postData.put ("area_code_type", "3");
            postData.put ("mouja_code", "123456");
            postData.put ("dag_number", "22");
            postData.put ("lease_area", "0.080");
            postData.put ("lease_value", "25000.00");
            postData.put ("lease_type", "1");
            postData.put ("lease_owner_name", "Abul Md Chowdhury");
            postData.put ("lease_owner_address", "address");
            postData.put ("return_type", "1");
         String queryString = buildQueryString(postData);
         String signature = createSignature(queryString,serviceId);

            URL queryUrl = new URL(URL);
            HttpURLConnection connection = (HttpURLConnection)query
Url.openConnection();
            connection.setDoOutput(true);
            connection.setRequestProperty("service_id", serviceId);
            connection.setRequestProperty("Sign", signature);
            connection.getOutputStream().write(queryString.getBytes());

            BufferedReader rd = new BufferedReader(new InputStreamReader(connection.getInputStream()));
            StringBuilder sb = new StringBuilder();
            String line;
            While ((line = rd.readLine ()) != null) {
                sb.append(line + '\n');
            }
            System.out.println(sb.toString());
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

private static String buildQueryString(Map<String, String> args) {
        StringBuilder result = new StringBuilder();
        for (String hashKey : args.keySet()) {
            if (result.length() > 0) result.append('&');
            try {
                result.append(URLEncoder.encode(hashKey, "UTF-8"))
                    .append("=").append(URLEncoder.encode(args.get(hashKey), "UTF-8"));
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
        return result.toString();
    }


    private static String createSignature(String paramData, String plainSecretKey) {
        try {
            SecretKeySpec secretKey = new SecretKeySpec(plainSecretKey.getBytes(UNICODE_CODE), HMAC_SHA256_ALGORITHM);
            Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
            mac.init(secretKey);
            byte[] hmacData = mac.doFinal(paramData.getBytes(UNICODE_CODE));
            return byteArrayToHexString(hmacData).toUpperCase();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }


    private static String byteArrayToHexString(byte[] bytes) {
        final char[] hexArray = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
        char[] hexChars = new char[bytes.length * 2];
        int v;
        for (int j = 0; j < bytes.length; j++) {
            v = bytes[j] & 0xFF;
            hexChars[j * 2] = hexArray[v >>> 4];
            hexChars[j * 2 + 1] = hexArray[v & 0x0F];
        }
        return new String(hexChars);
    }
}
```

GET request

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
public class ExtAPIExample {
    public static void main(String[] args) {
        try {
            URL queryUrl = new URL("https://api.land.gov.bd/live/newleaseowner?office_id=12312&area_code_type=3&mouja_code=1231312&dag_number=322&lease_area=0.080&lease_value=25000&lease_type=1&lease_owner_name=abul md chowdhur&lease_address=adress&return_type=1 ");
            HttpURLConnection connection = (HttpURLConnection)queryUrl.openConnection();
            connection.setDoOutput(true);
            BufferedReader rd = new BufferedReader(new InputStreamReader(connection.getInputStream()));
            StringBuilder sb = new StringBuilder();
            String line;
            while ((line = rd.readLine()) != null) {
                sb.append(line + '\n');
            }
            System.out.println(sb.toString());

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

C++

GET request

```
#include "stdafx.h"
#using <mscorlib.dll>
#using <System.dll>
#using <System.Web.dll>
using namespace System;
using namespace System::Text;
using namespace System::Web;
using namespace System::Net;
using namespace System::Security::Cryptography;
using namespace System::IO;

int _tmain(int argc, _TCHAR* argv[])
{
      String^ param = "username=joy&password=dfsf25";
      String^ uri = "https://api.land.gov.bd/live/usercheck";
      String^ ResponseFromServer = "";
      HttpStatusCode StatusCode;
      HttpWebRequest ^request = safe_cast<HttpWebRequest^>(WebRequest::Create(uri + param));
      request->Method = "GET";
      request->Timeout = 12000;
      request->ContentType = "application/x-www-form-urlencoded";
      Stream ^dataStream;
```

```
        try
        {
                WebResponse ^WebResponse = request->GetResponse();
                dataStream = WebResponse->GetResponseStream();
                StreamReader ^StreamReader = gcnew System::IO::StreamReader(dataStream);
                ResponseFromServer = StreamReader->ReadToEnd();
                dataStream->Close();
                WebResponse->Close();
                StatusCode = HttpStatusCode::OK;
        }catch (WebException ^ex){
                if (ex->Response != nullptr)
                {
                        dataStream = ex->Response->GetResponseStream();
                        StreamReader ^StreamReader = gcnew System::IO::StreamReader(dataStream);
                        StatusCode = (safe_cast<HttpWebResponse^>(ex->Response))->StatusCode;
                        ResponseFromServer = ex->Message;
                }else{
                        StatusCode = HttpStatusCode::ExpectationFailed;
                        ResponseFromServer = " Not familiar Error ";
                }
        }
        catch (Exception ^ex){
                StatusCode = HttpStatusCode::ExpectationFailed;
                ResponseFromServer = "Not familiar Error";
        }
Console::WriteLine("Response Code: " + StatusCode.ToString());
```

```
Console::WriteLine("Response String: " + ResponseFromServer);
}
```

C#

GET request

```
using System;
using System.Text;
using System.Web;
using System.Net;
using System.Security.Cryptography;
using System.IO;
namespace apice
{
    class Program{
        static void Main(string[] args)
        {
            string param = "username=Imran&password=234234sdsf";
            string uri = "https://api.land.gov.bd/live/usercheck";
            string ResponseFromServer = "";
            HttpStatusCode StatusCode;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(uri + param);
            request.Method = "GET";
            request.ContentType = "application/x-www-form-urlencoded";
            Stream dataStream;
            try
            {
                WebResponse WebResponse = request.GetResponse();
                dataStream = WebResponse.GetResponseStream();
                StreamReader StreamReader = new StreamReader(dataSt
ream);
                ResponseFromServer = StreamReader.ReadToEnd();
                dataStream.Close();
                WebResponse.Close();
                StatusCode = HttpStatusCode.OK;
            }
            catch (WebException ex)
            {
                if (ex.Response != null)
                {
                    dataStream = ex.Response.GetResponseStream();
                    StreamReader StreamReader = new StreamReader(da
taStream);
                    StatusCode = ((HttpWebResponse)ex.Response).Sta
tusCode;
                    ResponseFromServer = ex.Message;
                }else{
                    StatusCode = HttpStatusCode.ExpectationFailed;
                    ResponseFromServer = "Unknown Error";
                }
            }catch (Exception ex){
                StatusCode = HttpStatusCode.ExpectationFailed;
                ResponseFromServer = "Unknown Error";
            }
            Console.WriteLine("Response Code: " + StatusCode);
            Console.WriteLine("Response String: " + ResponseFromSer
ver);
```

```
        }
    }
}
```

RUBY

GET request

```
require "net/http"
require "uri"
require 'openssl'
require "base64"
require "rubygems"
require "json"
uri = URI::parse("https://api.land.gov.bd/live/usercheck")
service_id = "gJx7Wa7qXkPtmTAaK3ADCtr6m5rCYYMy"
access_code = "8eLps29wsXszNyEhOl9w8dxsOsM2lTzg"
uri.query = URI.encode_www_form({
'username'=> 'perso_nuser' , 'password'=> 'password',
'return_type'=> '1'
})
sha256 = OpenSSL::Digest::SHA256.new
signature = OpenSSL::HMAC.hexdigest(sha256, service_id, uri.query).upcase
request = Net::HTTP::Get.new(uri)
request.add_field("service_id", service_id)
request.add_field("access_code", access_code)
response = Net::HTTP.start(uri.hostname, uri.port, :use_ssl =>uri.scheme == 'https') {|http|
http.request(request)
}
```

```
response_data = JSON.parse(response.body())
 print response_data
```

PYTHON 2.7

GET request

```
import httplib
import urllib
import json
import hashlib
import hmac
from collections import OrderedDict
server = "api.land.gov.bd"
method = /live/usercheck "
service_id = "gJx7Wa7qXkPtmTAaK3ADCtr6m5rCYYMy"
access_code = "8eLps29wsXszNyEhOl9w8dxsOsM2lTzg"

data = OrderedDict([('username', 'shawon'),('password', '2342sdf'),
                     ('return_type', '1')])
encoded_data = urllib.urlencode(data)
sign = hmac.new(access_code, msg=encoded_data, digestmod=hashlib.sha256).hexdigest().upper()
headers = {"service_id": service_id, "access_code": access_code}
conn = httplib.HTTPSConnection(server)
conn.request("GET", method + '?' + encoded_data, '', headers)
response = conn.getresponse()
data = json.load(response)
conn.close()
print data
```

Response Codes

Code	Brief Definition
200	Success
201	Request has been fulfilled and new resource being created
202	Request has been accepted for processing
204	No Content Found
205	Reset Request
206	Partial GET request for the resource
301	Requested resource has been assigned another URL/Moved
302	Re-directed to new URL
305	The requested resource MUST be accessed through the proxy
400	Bad request
401	Unauthorized Access
402	Payment Required – reserved for future use
403	Forbidden Request
404	The requested resource/Method Not Found
409	Request couldn't be completed due to conflict with other request/resource
415	Unsupported Media type/Request Format not supported by LISF
429	Too many request
451	Unavailable for legal reason
500	Internal Server Error
501	Requested Resource/Method's functionality still not implemented
504	Bad Gateway
503	Service unavailable
504	Gateway Timeout
505	HTTP version not supported
600	HTTPS/Secure Connection required
601	Service id is missing
602	Access Code is missing
603	Please provide username
604	Please provide user password
605	Please provide office id
606	Return Type Missing
609	Missing at least One Required Parameter
611	Invalid Service Id
612	Invalid Access Code

625	Incorrect username or password
641	Blocked User ID
701	Wrong Office ID
702	Inactive Office
710	Blocked Office
801	Area Type Missing
802	Area Code Type Missing
803	Area Code Missing
804	Any Area Code Missing
804	Division/District/Upazila/Mouja Code Missing
805	Division Code Missing
806	District Code Missing
807	Upazila Code Missing
808	Mouja Code Missing
809	Union Code Missing
810	Thana Code Missing
811	Zone Code Missing
812	Pouroshova Code Missing
813	Pouroshova Ward Code Missing
814	City Corporation Code Missing
815	City Corporation Ward Code Missing
845	Invalid Zone Code
850	Invalid/Missing Area Code Type
851	Invalid Area Type
852	Invalid Area Code Type
853	Invalid Area Code
854	Invalid/Missing Division Code
855	Invalid/Missing District Code
856	Invalid/Missing Upazila Code
857	Invalid/Missing Mouja Code
864	Invalid Division Code
865	Invalid District Code
866	Invalid Upazila Code
867	Invalid Mouja Code
868	Invalid Union Code
869	Invalid Thana Code

870	Invalid Any Area Code
870	Invalid Division/District/Upazila/Mouja Code
871	Invalid City Corporation Code
872	Invalid City Corporation Ward Code
873	Invalid Pouroshova Code
874	Invalid Pouroshova Ward Code
875	Invalid Touji Code
876	Invalid Pargana Code
901	Survey Type Missing
911	Khatian Number Missing
912	Khatian Type Missing
921	Dag Number Missing
922	Dag Origin Type Missing
922	Survey Sub Type Missing
930	Khatian Owner Type Missing
931	Owner Name Missing
931	Name Missing
935	Invalid/Missing Khatian Entry/Approve Date-time
936	Invalid/Missing Khatian Entry/Approve User Info
937	Khatian Approval User must be LISF Register User
938	Invalid/Missing Khatian Update/Approve Date-time
939	Invalid/Missing Khatian Update/Approve User Info
941	This dag {dag_number} area can't bigger than total area
942	This dag {dag_number} can't be zero or null
943	This dag {dag_number} don't enough area to use here
944	This dag {dag_number} not permitted to use here
945	This dag {dag_number} must have valid land type
946	Total Dag Number & supplied dag data mismatch
947	Khatian owner name can't be blank or null
948	Khatian owner name must have an address
949	Total owner number & supplied owner name mismatch
950	Total owner percentage can't cross 100%
951	Invalid Survey Type
955	Invalid/Missing Survey Type
961	Invalid Khatian Number
962	Khatian Not Found/stored in System

963	Restricted Khatian, Can't deliverable
964	Invalid Khatian Type
965	Invalid/Missing Khatian Number
966	Invalid/Missing Khatian Type
966	Invalid/Missing Dag Origin Type
967	Same Khatian number exist for same mouja & khatian type
968	Khatian number doesn't exist for same mouja & khatian type
971	Invalid Dag Number
972	Invalid Dag Origin Type
973	Restricted Information, Can't deliverable
975	Invalid/Missing Dag Number
981	Owner Name Not Found
981	Provided Name Not Found
982	Owner Name does not match with Father/Husband Name
982	Provided Name does not match with Father/Husband Name
983	Owner Name does not match with Mother Name
983	Provided Name does not match with Mother Name
984	Owner Name does not match with National ID
984	Provided Name does not match with National ID
1101	Deed ID Missing
1111	Invalid Deed Id
1121	Owner Name and Deed ID Not Matched
1150	Deed's Not found or Stored in LISF
1201	2D Map Center Position Coordinates Missing
1202	2D Map Area Boundary Type Missing
1203	2D Map Boundary Coordinates Missing
1204	2D Map Zoom Level Missing
1221	Invalid Center Position Coordinates (2D MAP)
1222	Center Position is outside of Khatian Area (2D MAP)
1223	Boundary Coordinates are outside of Khatian Area
1224	Boundary Type and Number of Boundary Coordinates mismatch
1231	Invalid Zoom Level
1301	Leased dag {dag_number} area can't bigger than total area
1302	Leased dag {dag_number} can't be zero or null
1311	Lease owner name can’t be blank or null
1312	Lease owner name must have an address

1315	Invalid Lease start/end date
1316	Invalid/Missing Lease Type
1320	Lease Year field Missing
1321	Invalid Lease Year
1350	Exist Entry & Submitted Data matched 100% - No update..
1404	Issued Token & supplied token doesn't match
1410	Transactional Time Limit Expire
1420	Invalid Port Use for confirmation communication
1430	Previous service id, access code doesn't match current info
1450	Exist Khatian & Submitted Data matched 100% - No update

LISF Data Integration Glossary

Term	Brief Definition
Core Data Definition (CDD)	Universally agreed data structure to facilitate the sharing of data.
Origin Data Attribute	A field of data of which many can be combined to for a core data definition.
Core Data Dictionary	Catalogue of all core data definitions.
Core Data Domain	Grouping of core data definitions within an area of activity within the organization.
Origin Data	Data identified in LISF systems that populate core data definitions.
Recognized Data	Data identified in one or more systems that populate core data definitions but have yet to be implemented.
Local Data	Data in a system that has not been identified as for use in a Core Data Definition.
Replicated Core Data	A populated core data definition stored in a location different to its source and maintains core data definition structure.
Transformed Core Data	A populated core data definition stored in a location different to its source and maps CD definition to destination system data structure (custom client view of CDD).
Core Data Cache (CDC)	Stored copy of every populated core data definition, independent of any application.
Data Custodian	A person or division which is accountable for a collection of source core data attributes.
Core Data Catalogue	Summary list of all core data definitions (business view).
Origin/LISF System	Core Framework.
Application/Service	Other Application/System which integrate into Origin/LISF system.
LISF Information Hub (LISFIH)	Collection of core data definitions, origin data, CDC and collection of services to access and manipulate core data.

References

Fayad, Mohamed. *"Object-Oriented Application Frameworks"*.

Henning, Michi. *"API: Design Matters"*

Benslimane, Djamal; Schahram Dustdar; Amit Sheth (2008). "Services Mashups: The New Generation of Web Applications

Richardson, Leonard; Mike Amundsen (2013), RESTful web APIs, O'Reilly Media,*ISBN* 978-1-449-35806-8,

Fielding, Roy Thomas (2000). "Chapter 2: Network-based Application Architectures"

M. Hostetter, D. Kranz, C. Seed, C. Terman, S. Ward. *"Curl: A Gentle Slope Language for the Web"*

A Guidance Framework for Designing a Great Web API, by Eric Knipp and Gary Olliffe , Gartner

Technical assistance by:

Soft BD Ltd.

Flat No:B5, Panthavila,
58/11 Box Culvert, Panthapath
Dhaka 1205 (Opposite of Bashundhara City).
Mobile: +880-1710334756, +880-1917778999
Email: info@soft-bd.com
Website: SoftBD Ltd.