Lupl Open API

Introduction

The **Lupl Open API** provides user-level access via an API to the backend functionality of Lupl. With it, users can script complex or bulk operations that are not easily performed through the UI, and are also able to automate tasks so that human intervention is not needed.

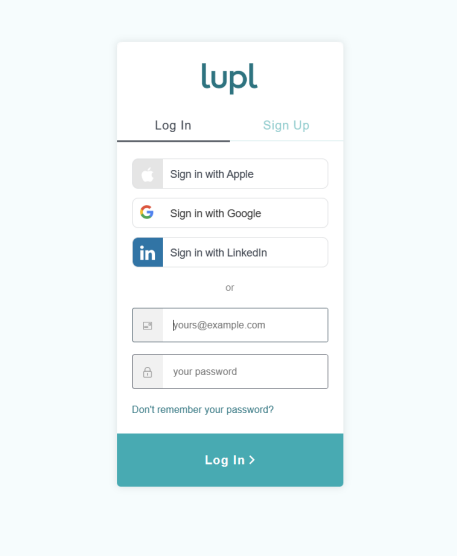
Beyond providing direct access to existing functionality, the **Lupl Open API** goes a step further by providing access to functionality that isn't available elsewhere. For instance, vendors who want to provide an integration that appears within Lupl (e.g. a practice management system) are able to register, create, configure, and test that integration via the **Lupl Open API**.

Initial Setup

Using the **Lupl Open API** can be as easy as grabbing your auth token, opening Postman, hitting an endpoint, and receiving responses. Some parts of the API have additional limitations, however, which we'll describe below. For now, let's walk through each part in turn.

Getting an Auth Token

The **Lupl Open API** utilizes OAuth2 protocol, and is leveraging Auth0 as an identity provider. To get a bearer and refresh token you will need to request a Client ID and Client Secret from the Lupl integrations team, which can be used with an Auth0 SDK of your preference - [Auth0 Libraries](#).



Note that tokens exist separately for each environment, so if you are targeting the Lupl Production environment (app.lupl.com) you should procure your token from that environment, whereas if you are readying an integration in the Stage environment (stage.lupl.com) you will need to use a Stage token.

Postman Collection

A Postman collection covering the endpoints for PMS providers is available [here](#).

PMS Providers

The Lupl Open API supports Practice Management System (PMS) providers who wish to enable an integration within Lupl to our mutual customers. Providers who want to enable such an integration need to register with Lupl and should be aware of some key concepts.

Key Concepts

A **PMS Provider Configuration**, sometimes referred to as simply a **provider config**, is a description provided to Lupl by you, a third-party Practice Management System provider, to configure Lupl with the details necessary to contact your service. In it, you can specify details such as your service's name, details for how the Lupl service can contact your service to retrieve the desired data for the integration, and list additional details that a mutual customer would need to provide in order to use the integration. For instance, you as a PMS provider may supply a configuration that says your name is ACME Inc., you provide endpoints from which we can retrieve data related to hours and billing, and a mutual customer would need to specify which region they belong to before they would be able to log in to your system to retrieve their data.

A **PMS Organization Configuration**, sometimes referred to in shorthand as an **org config**, is a set of configuration values for a particular organization that satisfy the details specified in the provider config. For instance, in the ACME provider config example above we stated mutual customers need to specify their region to use ACME. Thus, an organization using ACME as their PMS provider would need to provide an org config that specified which region they belong to.

More details and examples are provided in the sections below.

Registering as a PMS Provider

To be able to use the endpoints related to PMS Providers, you must first register with Lupl as a PMS Provider. Lupl will ask you to provide some basic information about your service, as well as the email address you use to login to Lupl. Once registered, you will be provided with a PMS ID. This ID is used on all related endpoints to identify the `PMSProviderConfiguration` on which you are operating. *These endpoints will only be accessible to you in the Lupl Stage environment (openapi-stage.stage.lupl.com).*

Endpoints

The full JSON schema of the Open API can be found here.

Examples of the JSON returned by success responses can be found in the full schema. In the case of a `403 Forbidden` response, an empty JSON object is typically returned in the body. In the case of a `400 Bad Request` response, a JSON object containing *at least* the following properties is returned:

```
Unset
```

```
{
  "errors": {
    "pmsId": [
      "The value 'asdf' is not valid."
    ]
  }
}
```

Get PMS Provider Configuration

`GET /api/1.0/pms/providerconfig`

- **Summary:** Retrieves the current configuration for a PMS provider.
- **Parameters (query)**
  - `pmsId` - ID of the PMS provider
- **Example Request**
  - `GET api/1.0/pms/providerconfig?pmsId=4efafdc0-2783-4af6-9cac-d2b2be27a080`
- **Responses**
  - `200 OK`
    - The body contains the JSON of the current PMS provider configuration. The configuration may be null if the provider is newly registered and has not uploaded a configuration yet.
  - `403 Forbidden`
    - Occurs if you have attempted to access a PMS provider that you are not authorized to ***or*** if the PMS provider does not exist.
  - `400 Bad Request`
    - Occurs if the request is not valid, and responds with some information about why.

Set PMS Provider Configuration

`POST /api/1.0/pms/providerconfig`

- **Summary:** Sets the current configuration for a PMS provider.
- **Parameters (body)**
  - `pmsId` - ID of the PMS provider
  - `config` - The configuration for the PMS provider
- **Example Request**
- **Responses**
  - `200 OK`
    - Occurs on successful execution, returning an empty JSON object.
  - `403 Forbidden`
    - Occurs if you have attempted to access a PMS provider that you are not authorized to ***or*** if the PMS provider does not exist.
  - `400 Bad Request`
    - Occurs if the request is not valid, and responds with some information about why.

Get PMS Organization Configuration

`GET /api/1.0/pms/organizationconfig`

- **Summary:** Gets an organization's current **PMS Org Config**.
- **Parameters (query)**
  - `organizationId` - ID of the organization
- **Example Request**
  - `GET api/1.0/organizationconfig?organizationId=3fa85f64-5717-4562-b3fc-2c963f66afa6`
- **Responses**
  - `200 OK`
  - `403 Forbidden`
    - Occurs if you have attempted to access an organization that you are not authorized to manage ***or*** if the organization does not exist.
  - `400 Bad Request`
    - Occurs if the request is not valid, and responds with some information about why.

Set PMS Organization Configuration

`POST /api/1.0/pms/organizationconfig`

- **Summary:** Sets the **PMS Org Config** for an organization.
- **Parameters (body)**
  - `organizationId` - ID of the organization
  - `pmsId` - ID of the PMS to set for the organization
  - `fields` - Configuration values for the PMS
- **Example Request**
- **Responses**
  - `200 OK`
    - Occurs on successful execution, returning an empty JSON object.
  - `403 Forbidden`
    - Occurs if you have attempted to access an organization that you are not authorized to manage ***or*** if the organization does not exist.
  - `400 Bad Request`
    - Occurs if the request is not valid, and responds with some information about why.

Usage Considerations

Environmental Scope

As previously described, some actions require operating out of the Lupl Stage environment, rather than the Lupl Production environment. Work done in the Lupl Stage environment is not automatically carried into the Lupl Production environment and vice versa. Once a Lupl representative is made aware of an integration—or other work—that needs to be published to the Lupl Production environment, Lupl will review the work and publish it accordingly to the Lupl Production environment. Please be sure to check which environment you are operating from.

Throttling (Rate Limiting)

Systems that make use of the **Lupl Open API** should exercise care in how often they make requests. To the degree that doing so is reasonable, caching should be used to prevent excessive calls. The API currently enforces a throttling policy (subject to change) of 1000 API calls per minute.

Support

Contact

For additional information or assistance, please contact Lupl via the following address:
it@lupl.com