

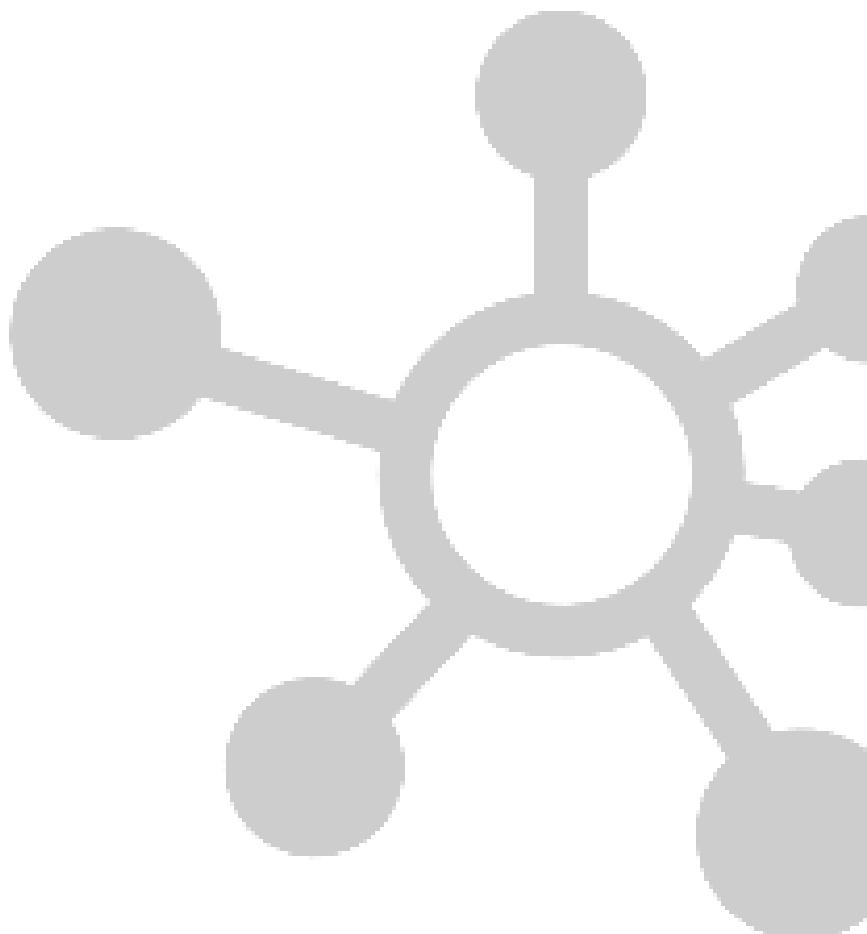
UPM API Quick Start Guide

Ext.prc.002.06 | 03.01.2025

UPM 25.x

ZINFI Confidential & Proprietary

Shared Under NDA



Contents

1. Introduction	3
1.1. Definitions of terms used	3
1.1.1. Fields/Data Field	3
1.1.2. Record	3
1.1.3. Entity	3
1.1.4. Resource.....	3
1.2. Resources	4
1.2. Characteristics and Architecture	4
2. Authentication.....	5
2.1. Implicit Authentication Flow	5
3. Prerequisites	7
4. UPM Entities/Modules and Attributes/Fields	7
4.1. Sample Entities/Modules.....	7
4.1.1. Prospect/Leads	7
4.1.2. Contacts	7
4.1.3. Accounts	7
4.1.4. Opportunities.....	7
4.2 Accessing UPM Entities/Modules (with Entity ID) and Field Details	8
4.3. Sample UPM Entity (with Entity ID) and Attributes.....	11
4.3.1. Prospect.....	11
4.3.2. Contact.....	12
4.3.3. Account	12
4.3.4. Opportunity.....	13
4.4. Application Data Types	14
5.0 API Calls	15
5.1. Get Token for Authentication.....	15
5.2. Read List of Records	15
5.3. Read Details of Specific Record of a Specific Entity/Module	17
5.4. Add and Edit	18
5.6. Response Codes.....	20
5.6.1. Success Response.....	20
5.6.2. Error Response	20
5.7. Response Format	21
5.8. API Call Sequence	21
5.9. API Call Limit.....	21

1. Introduction

UPM API provides a powerful, easy-to-use and simple RESTful web service API for integrating third-party applications with the UPM platform. UPM API is easy to integrate with any HTTP aware applications of various technologies. Integration of the API with your application requires basic familiarity with REST web services, HTTP methods, JSON and ZINFI's UPM platform.

This quick start guide describes:

- Characteristics and architecture of the UPM API. This will help in understanding how the entities within the UPM platform are distributed and/or connected.
- How to set up the development environment.
- How a third party can retrieve/edit/delete UPM resources via UPM API in their own application.
- A step-by-step quick-start example for a typical use case.

1.1. Definitions of terms used

1.1.1. Fields/Data Field

A data field represents an attribute (a characteristic or quality) of some entity (object, person, place, or event). e.g.: a name field, an address field, and a phone number field.

1.1.2. Record

A record is composed of fields, each of which contains one item of information, with each field describing an attribute of the entity. For example, a personnel record “John” might have three fields: a name field, an address field and a phone number field. In relational database management systems, records are called tuples.

1.1.3. Entity

An entity is any object in the system that we want to model and store information about. Entities are usually recognizable concepts, either concrete or abstract, such as persons, places, things or events which have relevance to the database. Some specific examples of entities are Personnel, Student and Lecturer.

1.1.4. Resource

A resource is an endpoint for a specific UPM Platform API.

1.2. Resources

A resource is an abstraction of a piece of information or an action, such as a single data record, a collection of records or a query. Each resource in UPM is identified by a named Uniform Resource Identifier (URI) and is accessed using standard HTTP methods GET and POST.

A third-party application uses a resource to interact with the UPM platform. For example, one can:

- Retrieve summary information about the API availability.
- Obtain detailed information about a record, such as an account, user or record of a custom entity.
- Retrieve records via a query or search filter.
- Update or delete specific records or records.

1.2. Characteristics and Architecture

Stateless

UPM API is stateless—i.e., it does not store any context in the server to share between requests. Each service request from the client must contain all the necessary information required by the service. The relations between entities are managed using interconnected resource URLs. This allows a client to progress between states when manipulating the relational entities of UPM.

Caching behavior

Responses are labeled as cacheable or non-cacheable.

Uniform interface

All records are accessed with a generic interface over HTTP.

Named resources

All resources are named using a base URI. API URIs has been named to convey the UPM resource model so the API is intuitive and easy to use.

Authentication

UPM API is secured with OpenId Connect over OAuth 2.0 security specifications. Users need to retrieve an access token from the ZINFI Identity Server before they can call any protected UPM API services from the UPM Application.

Support for JSON

JavaScript Object Notation (JSON) is the default format for request and response data. JSON is supported with UTF-8 in UPM API.

2. Authentication

ZINFI uses the OpenID Connect authorization framework over the OAuth 2.0 security protocol to protect UPM API. Third-party applications have to retrieve an access token in JWT bearer token format from a ZINFI Identity Server. This JWT bearer token allows the application to connect securely with the UPM API to access data from the UPM platform.

2.1. Implicit Authentication Flow

- The application or client requests authorization to the ZINFI Identity Server. A typical OpenID Connect compliant web application will go through the OAuth/authorize endpoint using the implicit flow. ZINFI Identity Server allows the resource owner to grant credential flow only with an email request.
- The application user lands on a login screen where the user provides their authentication credentials. When these credentials are verified from the UPM platform, users and authorization will be granted.
- Once authorization is granted, the ZINFI Identity Server returns the identity and access token to the application.
- The application uses the access token to access a protected resource (like an API). The identity token is used to verify the identity of the provider.

The following diagram shows how a JWT access token is obtained and used to access APIs or resources:

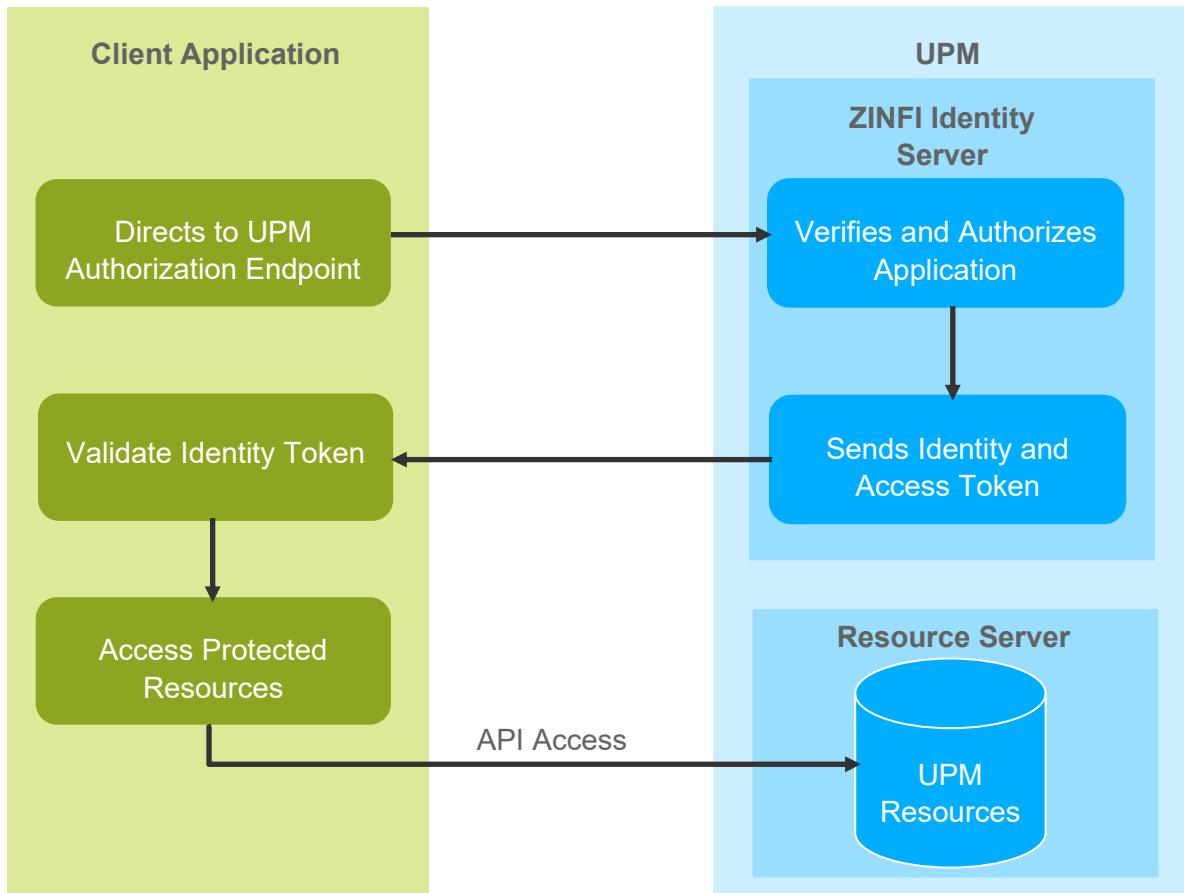


Figure 1: Application Authentication Using Implicit Flow

3. Prerequisites

Before developing a client application, we recommend you test the API services with publicly available API testing tools such as cURL, PostMan or Fiddler. This section will provide you with a list of software tools we have been using to describe our services in this document. Primary knowledge about these tools will help you understand the codes quickly. However, there is no restriction on which tool to use. You are free to choose any alternative available.

4. UPM Entities/Modules and Attributes/Fields

4.1. Sample Entities/Modules

4.1.1. Prospect/Leads

A prospect/lead is a prospective client or a potential opportunity, also called an “unqualified sales opportunity.” Leads can emerge from real-life interactions, such as meeting someone at a conference, or they can come from online interactions, such as when someone fills out a form on your website requesting more information.

When you convert a lead in UPM, the system guides you through an integrated path of lead-nurturing through the creation of new accounts, contacts, opportunities and registered deals, while connecting that data with the information already in its system.

4.1.2. Contacts

Upon qualification, the lead is converted into a contact. A contact object contains information about the person associated with the lead. A contact is a person associated with an account.

Data associated with a contact includes first name, last name, phone, email, qualifications, etc.

4.1.3. Accounts

An account is an organization or person involved with your business such as a customer, competitor or partner.

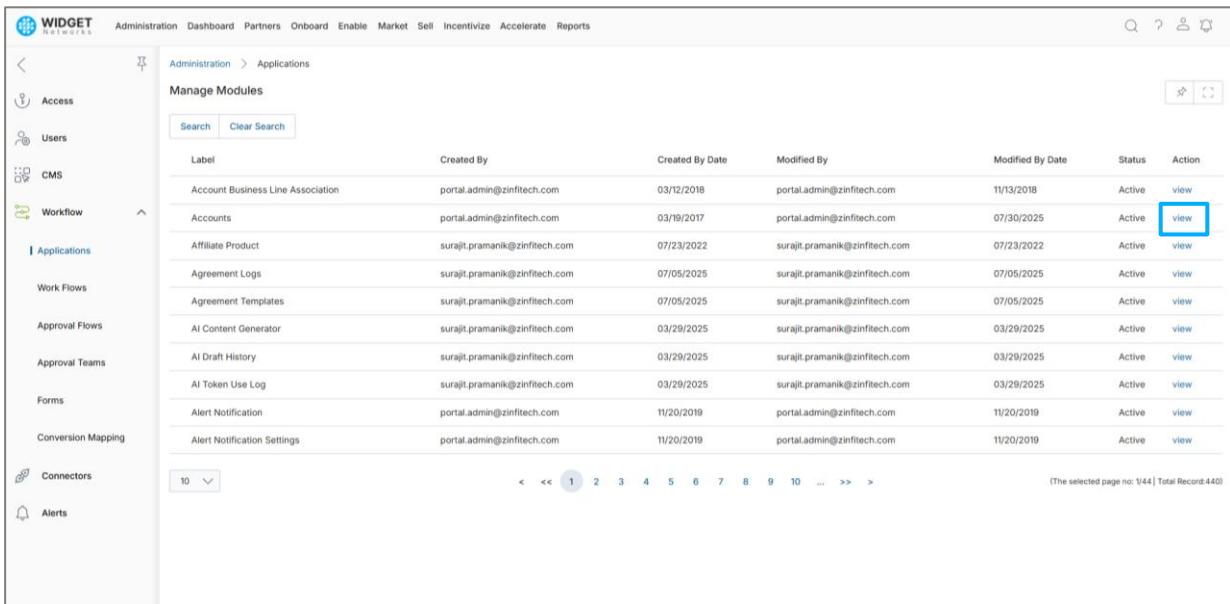
4.1.4. Opportunities

Opportunities include all the information relating to specific sales, campaigns and pending deals. From this information you can easily create lists of various types, such as products, quotes or contacts. Like the account and contact fields, opportunity fields can be customized.

An opportunity normally refers to a sale or pending deal. Multiple opportunities make up the sales pipeline, which contributes to your sales forecast. Opportunities are an integral part of UPM, and it's essential to maintain up-to-date information about all of your opportunities to track progress and make accurate forecasts.

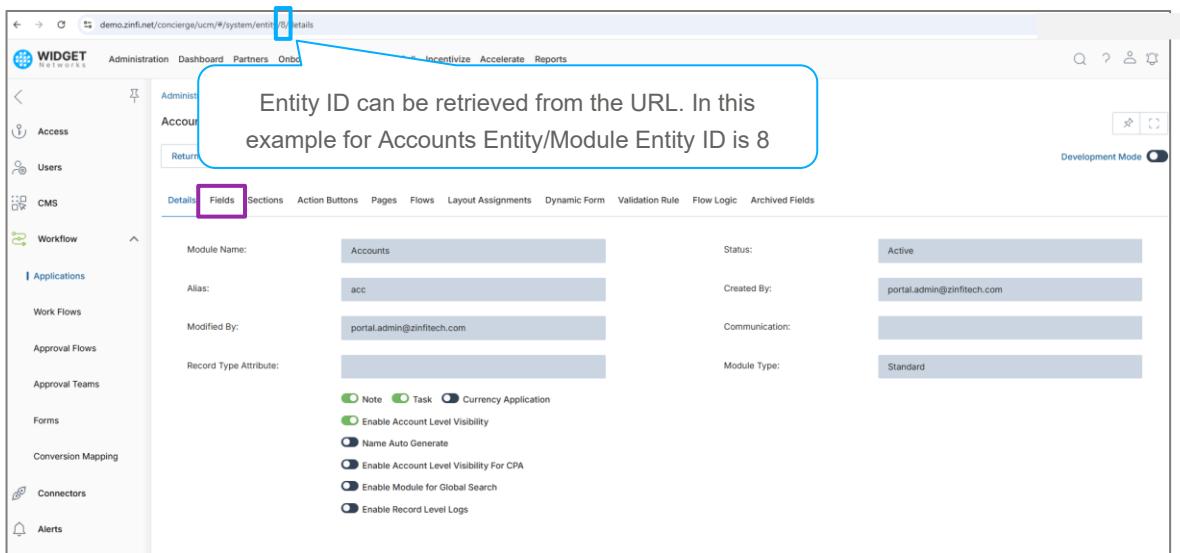
4.2 Accessing UPM Entities/Modules (with Entity ID) and Field Details

1. Login to UPM as Admin and Click on **Administration** on the top menu.
2. Click on **Workflow > Applications** from the left menu bar.
3. List of UPM Entities/Modules are displayed. You can click on Search to Search a specific Entity/Module,
4. For each Entity Record in the list, click **View** to open the details of a specific Module.

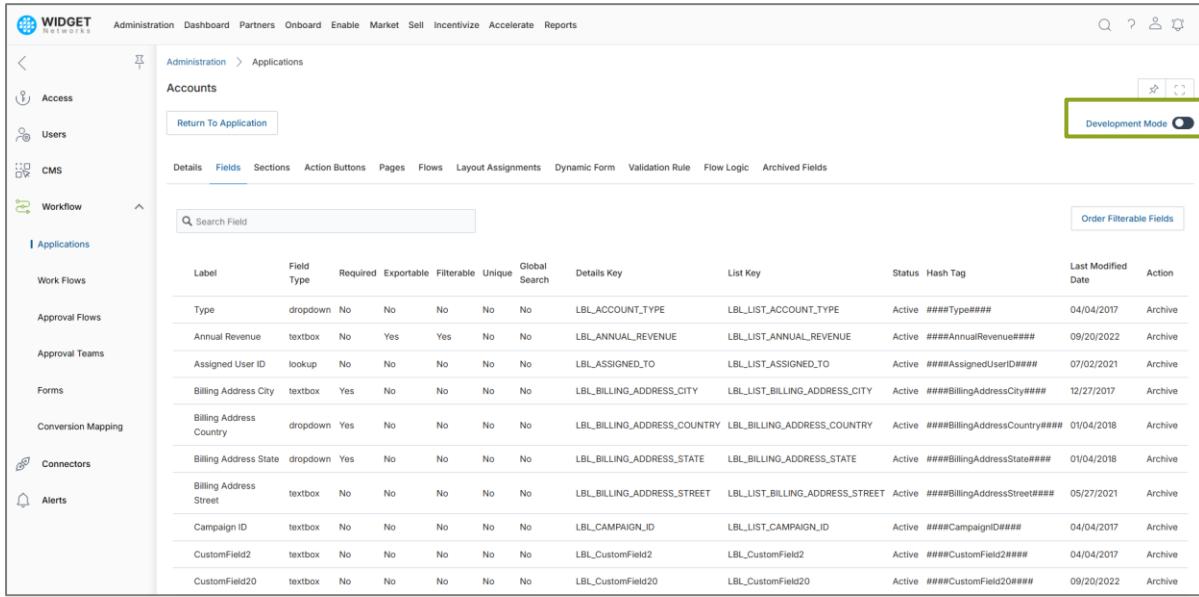


View Module Details

5. The Module Details will assist us to **Get the Module ID and View Field Details**.

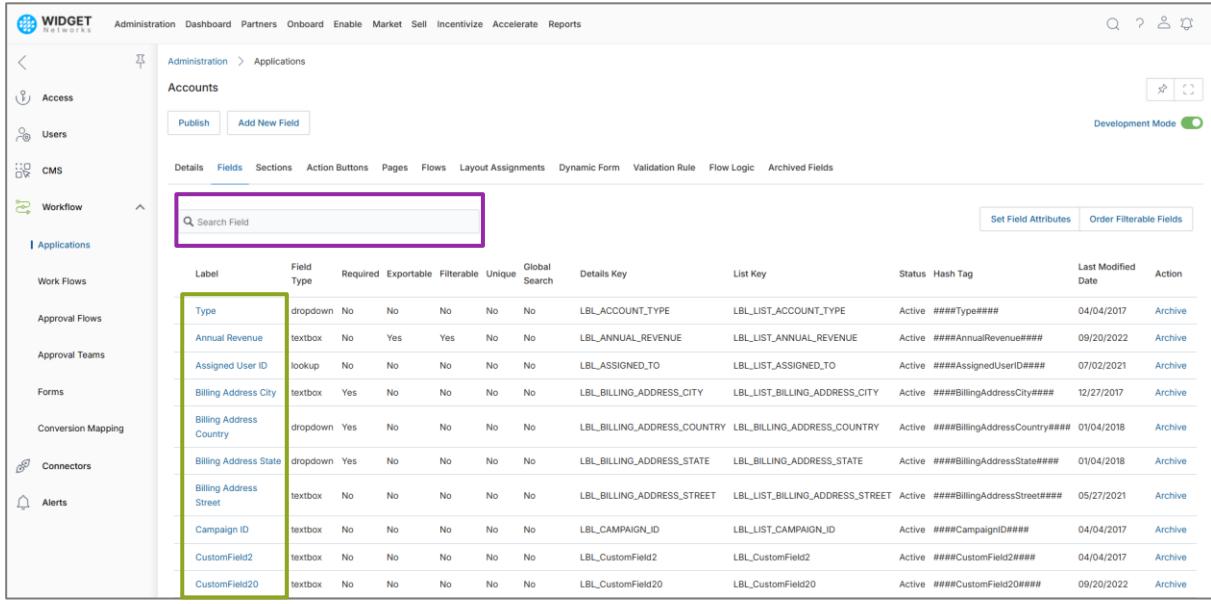


6. Click on Fields to view Entity Fields.



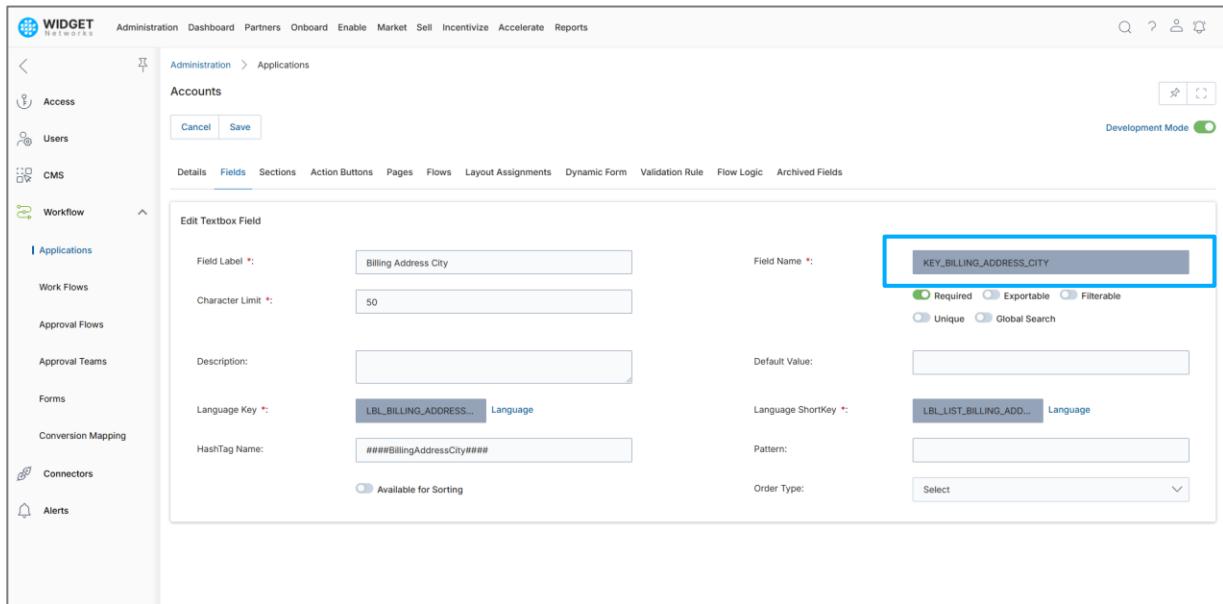
The screenshot shows the Zinfi Platform's administration interface for the 'Accounts' entity. The 'Fields' tab is selected. A green box highlights the 'Development Mode' button in the top right corner. The table lists various fields with their details such as Label, Field Type, Required, Exportable, Filterable, Unique, Global Search, Details Key, List Key, Status, Hash Tag, Last Modified Date, and Action. Fields include Type, Annual Revenue, Assigned User ID, Billing Address City, Billing Address Country, Billing Address State, Billing Address Street, Campaign ID, CustomField2, and CustomField20.

7. To View Field Details of a particular field - **Development Mode** should be enabled. Once the development Mode is enabled, we find the fields contained in the **Label** column become hyperlinks.



The screenshot shows the Zinfi Platform's administration interface for the 'Accounts' entity. The 'Fields' tab is selected. A green box highlights the 'Label' column for the 'Type' field. The table lists various fields with their details. The 'Type' field is shown as a hyperlink, while the other fields are standard text entries.

8. We can speed search for a Field, by keying in the Field Label in the Search Field and Hit Enter.
9. **Clicking the desired hyperlink field takes us to the Field Details Page. The Field Details Page will assist us to access the Field Name.**



Administration > Applications > Accounts

Details Fields Sections Action Buttons Pages Flows Layout Assignments Dynamic Form Validation Rule Flow Logic Archived Fields

Edit Textbox Field

Field Label *: Billing Address City

Character Limit *: 50

Description:

Language Key *: LBL_BILLING_ADDRESS...

HashTag Name: #####BillingAddressCity#####

Field Name *: KEY_BILLING_ADDRESS_CITY

Required Exportable Filterable Unique Global Search

Default Value:

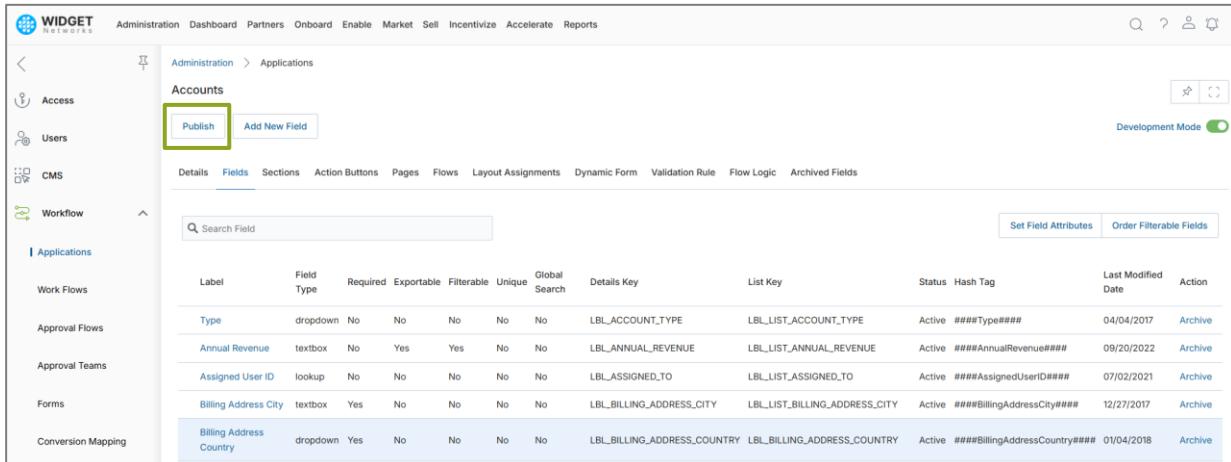
Language ShortKey *: LBL_LIST_BILLING_ADD...

Language:

Pattern:

Available for Sorting Order Type: Select

10. Once the Field Names are retrieved, it is mandatory to Disable Development Mode and Click Publish for the proper functioning of the UPM Module/Entity.



Administration > Applications > Accounts

Details Fields Sections Action Buttons Pages Flows Layout Assignments Dynamic Form Validation Rule Flow Logic Archived Fields

Publish Add New Field

Set Field Attributes Order Filterable Fields

Search Field

Label Field Type Required Exportable Filterable Unique Global Search Details Key List Key Status Hash Tag Last Modified Date Action

Type	dropdown	No	No	No	No	No	LBL_ACCOUNT_TYPE	LBL_LIST_ACCOUNT_TYPE	Active #####Type#####	04/04/2017	Archive
Annual Revenue	textbox	No	Yes	Yes	No	No	LBL_ANNUAL_REVENUE	LBL_LIST_ANNUAL_REVENUE	Active #####AnnualRevenue#####	09/20/2022	Archive
Assigned User ID	lookup	No	No	No	No	No	LBL_ASSIGNED_TO	LBL_LIST_ASSIGNED_TO	Active #####AssignedUserID#####	07/02/2021	Archive
Billing Address City	textbox	Yes	No	No	No	No	LBL_BILLING_ADDRESS_CITY	LBL_LIST_BILLING_ADDRESS_CITY	Active #####BillingAddressCity#####	12/27/2017	Archive
Billing Address Country	dropdown	Yes	No	No	No	No	LBL_BILLING_ADDRESS_COUNTRY	LBL_BILLING_ADDRESS_COUNTRY	Active #####BillingAddressCountry#####	01/04/2018	Archive

Note: Fields which are enabled for viewership in the List Page/Details Page of the Module/Entity can be only retrieved via the specific APIs. To check such Fields enabled in those Pages click Pages and review the Fields listed.

4.3. Sample UPM Entity (with Entity ID) and Attributes

4.3.1. Prospect

Entity ID: 9

Name	Field Type	Required
<code>Id</code>	textbox	
<code>FirstName</code>	textbox	Yes
<code>LastName</code>	textbox	Yes
<code>KEY_SALUTATION</code>	textbox	
<code>COMPANY_NAME</code>	textbox	Yes
<code>KEY_TITLE</code>	textbox	
<code>STATUS</code>	dropdown	
<code>EMAIL1</code>	email	Yes
<code>INDUSTRY_TYPE</code>	textbox	
<code>LEAD_SOURCE</code>	dropdown	
<code>EMPLOYEE_RANGE</code>	textbox	
<code>KEY_COMPANY_WEBSITE</code>	textbox	
<code>KEY_Phone</code>	textbox	
<code>Competitor_Name</code>	textbox	
<code>Competitor_Description</code>	textbox	
<code>Competitor_Strength</code>	textbox	
<code>Competitor_Weakness</code>	textbox	
<code>PHONE_MOBILE</code>	textbox	
<code>KEY_DEPARTMENT</code>	textbox	
<code>KEY_DESCRIPTION</code>	textArea	
<code>KEY_PRIMARY_ADDRESS_STREET</code>	textbox	
<code>KEY_PRIMARY_ADDRESS_STATE</code>	dropdown	
<code>KEY_PRIMARY_ADDRESS_COUNTRY</code>	dropdown	
<code>KEY_PRIMARY_ADDRESS_CITY</code>	textbox	
<code>KEY_ALT_ADDRESS_STREET</code>	textbox	
<code>KEY_ALT_ADDRESS_CITY</code>	textbox	
<code>KEY_ALT_ADDRESS_COUNTRY</code>	dropdown	
<code>KEY_ALT_ADDRESS_POSTALCODE</code>	textbox	
<code>KEY_ALT_ADDRESS_STATE</code>	dropdown	

4.3.2. Contact

Entity ID: 10

Name	Field Type	Required
Id	textbox	
KEY_SALUTATION	textbox	
KEY_FIRST_NAME	textbox	Yes
KEY_LAST_NAME	textbox	Yes
KEY_TITLE	textbox	
KEY_PHONE_HOME	textbox	
KEY_BIRTHDATE	textbox	
KEY_LEAD_SOURCE	dropdown	
KEY_PHONE_MOBILE	textbox	
KEY_DEPARTMENT	textbox	
KEY_DESCRIPTION	textbox	
KEY_EMAIL1	email	
KEY_PRIMARY_ADDRESS_STREET	textbox	
KEY_PRIMARY_ADDRESS_CITY	textbox	
KEY_PRIMARY_ADDRESS_STATE	dropdown	
KEY_PRIMARY_ADDRESS_COUNTRY	dropdown	
KEY_ALT_ADDRESS_STREET	textbox	
KEY_ALT_ADDRESS_CITY	textbox	
KEY_ALT_ADDRESS_POSTALCODE	textbox	
KEY_ALT_ADDRESS_STATE	dropdown	
KEY_ALT_ADDRESS_COUNTRY	dropdown	Yes

4.3.3. Account

Entity ID: 8

Name	Field Type	Required
Id	textbox	
Name	textbox	Yes
KEY_ACCOUNT_TYPE	dropdown	
KEY_ANNUAL_REVENUE	textbox	
KEY_DESCRIPTION	textArea	
KEY_EMPLOYEES	textbox	

KEY_INDUSTRY	dropdown	
KEY_RATING	textbox	
KEY_WEBSITE	textbox	
KEY_BILLING_ADDRESS_STREET	textbox	
KEY_PHONE_OFFICE	textbox	
KEY_BILLING_ADDRESS_STATE	dropdown	Yes
KEY_BILLING_ADDRESS_COUNTRY	dropdown	Yes
KEY_BILLING_ADDRESS_CITY	textbox	Yes
KEY_BILLING_ADDRESS_POSTALCODE	textbox	
KEY_SHIPPING_ADDRESS_STREET	textbox	
KEY_SHIPPING_ADDRESS_CITY	textbox	
KEY_SHIPPING_ADDRESS_POSTALCODE	textbox	
KEY_SHIPPING_ADDRESS_STATE	dropdown	
KEY_SHIPPING_ADDRESS_COUNTRY	dropdown	

4.3.4. Opportunity

Entity ID: 11

Name	Field Type	Required
Id	textbox	
Name	textbox	Yes
KEY_SALES_STAGE	dropdown	Yes
KEY_ACCOUNT_NAME	lookup	Yes
KEY_CONTACT_NAME	lookup	Yes
Quarter	dropdown	
CurrencyISOCode	dropdown	Yes
Price_book	lookup	
KEY_AMOUNT	textbox	Yes
KEY_DATE_CLOSED	date	
KEY_DESCRIPTION	textbox	

4.4. Application Data Types

Standard data types and their expected “Field Type” and “Description” values when creating properties programmatically are listed below. These represent the data type within the backend database and dictate how a field is represented within a form.

Field Type	Description
Text Box	Supports entry of any text on a single line (letters, special characters and numbers)
Multiline Text Box	Supports entry of any text on multiple/separate lines (letters, special characters and numbers)
Drop down List	Supports selection of a value from a pre-defined list
Lookup Relationship	Creates a linking relationship between two objects (EX: clicking on the lookup icon allows selection of a value from a popup list)
Date	Supports entry of a date, or selection from a popup calendar
Email	Supports entry of an email address (that gets auto-validated) - If this field is aligned with a contact or lead, then the email address is selectable when sending a system-generated email
Number	Supports entry of any number(s) - (leading zeros are removed)
Checkbox	Supports a true (checked) or false (unchecked) status for a value
Phone	Supports entry of a phone number (that gets auto-formatted)
File	Supports the selection and upload of an external file to the system
Auto Number	Creates a system-generated number (sequentially added) using a custom pre-defined format
Formula	Supports creation of a formula (user created) to derive a computed value from defined entities
Rich Text Box	Supports entry of any letters, special characters and number with custom formatting options
Progress Bar	Creates a progress bar
Multi Select Drop down List	Supports selection of multiple values from a pre-defined list
Color Picker	Allows users to pick a color from color palette
Currency	Allows users to enter any number. Leading zeros are removed.
Image	Allows users to upload the image that will get previewed before upload.
Declaration	Allows users to add declaration.
Duration	Allows users to add duration.
Link	Supports entry of any text which is a hyperlink
Percentage	percentage

5.0 API Calls

5.1. Get Token for Authentication

```
Url: https://auth.zinfo.net/connect/token
Method: POST
Content-Type: application/x-www-form-urlencoded
Parameters -->
    client_id: UPM-API
    username: <<UserName>>
    password: ****
    grant_type: password
    scope: UPM9xAPI
    client_secret: ****
Response:
{
    "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjNiYTQ5NjIwNzFiNDJmNGFhNmMxM2I1OGY5YzZkMmQxIiwidHlwIjoiSldUIn0.eyJyJuYmYiOjE1MzMwMTU3NTYsImV4cCI6MTUzMzAxNjA1NiwiaXNzIjoiHR0cHM6Ly9zaWduaW4xLnp_YMkbr2gLMLH2ZFxnjs5z_sdJf8KCA",
    "expires_in": 300,
    "token_type": "Bearer"
}
```

5.2. Read List of Records

API Endpoint/URL:

<https://{{your UPM Domain}}/concierge/UPMapi/api/UPM/recordlistobj>

Method: POST

Content Type: application/json

Authorization: Bearer {token}

Body:

```
{
    "EntityId": "77", //The Entity ID can be updated for a specific Module as per Use Case Requirement
    "totalItems": 0,
```

```
"pageNo": 1,  
"numPages": 0,  
"pageSize": 10,  
"maxSize": 10,  
"orderBy": "[Modify.Date]",  
"orderType": "desc",  
"Lang": "en-US",  
"returnType": "list"  
}
```

Sample Response:

```
{  
  " isSuccess": true,  
  " errors": [],  
  " data": null,  
  " objectdata": [  
    {  
      "Id": "F7461B3E-2F66-45DB-AD5F-21D446F05425",  
      "access": "5",  
      "Name": " Ruth Jones",  
      "User_Name": "ruth.jones@zinfo.com",  
      "Email_Address": " ruth.jones@zinfo.com ",  
      "Direct_Phone": "123125",  
      "Status": "Active",  
      ...  
    }  
    {  
      "Id": "C61D18EF-131F-4BFD-A3AD-CD80F4EBFF05",  
      "access": "5",  
      "Name": " Harry Jones",  
      "User_Name": "harry.jones@zinfo.com",  
      "Email_Address": " harry.jones@zinfo.com ",  
      "Direct_Phone": "2345627845",  
    }  
  ]  
}
```

```
        "Status": "Active",  
        ...  
    }  
]  
}
```

5.3. Read Details of Specific Record of a Specific Entity/Module

API Endpoint/URL:

<https://{{your UPM Domian}}/concierge/UPMapi/api/UPM/entity/{{Entity ID}}/recorddetails/{{Record ID}}>

Method: GET

Content Type: application/json

Authorization: Bearer {token}

Response:

```
{  
    "isSuccess": true,  
    "errors": [],  
    "data": null,  
    "objectdata": {  
        "data": [  
            {  
                "Id": "07D63D4A-FB80-4376-8260-2DD43E3E61D5",  
                "Name": "Dave Jones",  
                "access": "0",  
                "EMAIL1": "dave.jones@zinfo.com",  
                "FirstName": "Dave",  
                "LastName": "Jones",  
                "Favorite": false,  
                "COMPANY_NAME": "ZINFO Inc.",  
                "Createby": {  
                    "Name": "John Doe",  
                    "Email": "john.doe@zinfo.com",  
                    "Id": "12345678901234567890123456789012"  
                }  
            }  
        ]  
    }  
}
```

```
        "Name": "mike.corrado@zinfi.com",
        "Date": "2022-12-08T05:24:08.573",
        "Id": "48BB5B22-2AC8-4BDC-909E-14BBCB1524F4"
    },
    "Modify": {
        "Name": "jake.admin@admin.com",
        "Date": "2023-02-27T13:15:44.41",
        "Id": "30F56915-5170-48FC-AA99-D5C4F74C6A54"
    }
}
]
```

5.4. Add and Edit

API Endpoint/URL:

<https://fyour UPM Domian}/concierge/UPMapi/api/UPM/Entity>

Method: POST

Content Type: application/json

Authorization: Bearer {token}

Body:

```
{
    "data": {
        "Id": "07D63D4A-FB80-4376-8260-2DD43E3E61D5",
        "EMAIL1": "Wildey.Oscar@yopmail.com",
        "KEY_TITLE": null,
        "FirstName": "Wildey",
        "LastName": "Oscar",
        "KEY_Phone": null,
        "PHONE_MOBILE": null,
    }
}
```

```
"LEAD_SOURCE": null,  
"STATUS": null,  
"Score": null,  
"Survey_Sent": null,  
"COMPANY_NAME": "ZINFO Inc.",  
"KEY_COMPANY_WEBSITE": null,  
"EMPLOYEE_RANGE": null,  
"INDUSTRY_TYPE": null,  
"KEY_CustomField1": null,  
"PHONE_WORK": null,  
"KEY_PHONE_FAX": null,  
"Competitor_Name": null,  
"Competitor_Description": null,  
"Competitor_Strength": null,  
"Competitor_Weakness": null,  
"KEY_PRIMARY_ADDRESS_STREET": null,  
"KEY_PRIMARY_ADDRESS_STREET2": null,  
"KEY_PRIMARY_ADDRESS_CITY": null,  
"KEY_PRIMARY_ADDRESS_COUNTRY": null,  
"KEY_PRIMARY_ADDRESS_STATE": null,  
"KEY_PRIMARY_ADDRESS_POSTALCODE": null,  
"KEY_ALT_ADDRESS_STREET": null,  
"KEY_ALT_ADDRESS_STREET2": null,  
"KEY_ALT_ADDRESS_CITY": null,  
"KEY_ALT_ADDRESS_COUNTRY": null,  
"KEY_ALT_ADDRESS_STATE": null,  
"KEY_ALT_ADDRESS_POSTALCODE": null,  
"KEY_DESCRIPTION": null  
},  
"enitiy": "9", //The Entity ID can be updated for a specific Module as per Use Case Requirement  
  
"mode": "e", //Mode e is for Edit Mode and Mode n is for Add/New Mode  
}
```

Response:

```
{  
  "isSuccess": true,  
  "errors": [],  
  "data": "[{"ErrorNumber":0,"ErrorSeverity":"Yes","ErrorMessage":"Data Saved\\Updated  
Successfully","Purpose":"SaveData","DataID":"07D63D4A-FB80-4376-8260-  
2DD43E3E61D5"}]",  
  "objectdata": null  
}
```

5.6. Response Codes

5.6.1. Success Response

Code 200 – Valid Request and Response

5.6.2. Error Response

Code	Type	Response
401	Unauthorized Access	No Content
404	Page not Found	No Content
500	Server Error	<pre>{ "errors": { "id": ["The value 'recordlistobj1' is not valid."] }, "type": "https://tools.ietf.org/html/rfc7231#section- 6.5.1", "title": "One or more validation errors occurred.", "status": 500, "traceId": "00-630de49b232e994991b143c1ec8a2406- 3c00a8cd2ef4f948-00" }</pre>
400	Bad Request	<pre>{ "errors": { "id": [] } }</pre>

		<pre> "The value 'recordlistobj1' is not valid."] }, "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1", "title": "One or more validation errors occurred.", "status": 400, "traceId": "00-630de49b232e994991b143c1ec8a2406-3c00a8cd2ef4f948-00" }</pre>
--	--	---

5.7. Response Format

JSON

5.8. API Call Sequence

For each call, your client application typically:

- Prepares the request by defining request parameters, if applicable.
- Invokes the call, which passes the request with its parameters to the UPM Platform web service for processing.
- Receives the response from the API.
- Handles the response, either by processing the returned data (for a successful invocation) or by handling the error (for a failed invocation).

5.9. API Call Limit

The public endpoints are powered by the same underlying technology that powers the core integration process. As a result, engineering teams closely monitor usage of the public APIs to ensure the best quality experience for users of the integrated application.

UPM has the following limits in place for API requests:

- We can do pre-defined calls to the API as per your UPM license.