# Address book REST API

## Login API

**URL**: /api/auth/login

**Method**: POST

**Input JSON:**

{

"user\_name": String,

"password": String

}

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Value |
| Header | Content-Type | Define the http content type | application/json |
| BODY | user\_name | Username of the user |  |
| BODY | password | Password of the user |  |

**Response**

JSON

{

"access\_token": String,

"token\_type": "Bearer"

}

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success |  |
| 401 | Unauthorized |  |
| 500 | Internal server error |  |
|  |  |  |

## Metadata API

To Fetch list of metadata using a RefSet key like Address Type, Email address type, phone number type, country etc.,

**URL**: api/meta-data/ref-set/{key}

**Method**: GET

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Path | key | RefSet key to get the associated values | 1. ADDRESS\_TYPE 2. PHONE\_NUMBER\_TYPE 3. EMAIL\_ADDRESS\_TYPE 4. COUNTRY |

**Response**

JSON

[

{

"id": String,

"key": String,

"description": String,

}

]

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | List of metadata linked to the refset key |
| 401 | Unauthorized |  |
| 404 | Not Found |  |
| 500 | Internal server error |  |

## Create Address Book API

Create address book with first name, last name and their communication details.

**URL**: /api/address-book

**Method**: POST

**Input JSON:**

{

"first\_name":"Balasubramanian",

"last\_name" : "V",

"emails":[

{

"email\_address":"vbalaas@gmail.com",

"type":{

"key":"PERSONAL"

}

}

],

"addresses":[

{

"line1" : "12131",

"line2" : "street",

"city" : "chennai",

"zipcode" : "69991",

"state\_name" : "Tamilnadu",

"type":{

"key":"HOME"

},

"country":{

"key":"INDIA"

}

}

],

"phones":[

{

"phone\_number" :"9872981231",

"type":{

"key":"PERSONAL"

}

}

]

}

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| BODY | first\_name | First Name of the User |  |
| BODY | last\_name | Last Name of the user |  |
| BODY | emails | List of email addresses objects |  |
|  | email\_address | Email address of the user |  |
|  | "type":{"key":"WORK"} | Email address type | 1. PERSONAL 2. WORK |
| BODY | phones | List of phone number objects |  |
|  | phone\_number | Phone Number of the user |  |
|  | "type":{ "key":"WORK"} | Phone Number type | 1. PERSONAL 2. WORK 3. ALTERNATE |
| BODY | address | List of addresses objects |  |
|  | "line1" : "12131", "line2" : "street",  "city" : "chennai", "zipcode" : "69991",  "stateName" : Tamilnadu", | Address of the user |  |
|  | "type":{  "key":"PERSONAL"  } | Address type | 1. PERSONAL 2. WORK |
|  | "country":{  "key":"INDIA"  } | Country of the user |  |

**Response**

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 201 | Created | Id of the address book |
| 401 | Unauthorized |  |
| 409 | Conflict | Error messages like email address already exists and invalid value for a field |
| 404 | Not found | Error when the metadata is not found in the database. |
| 500 | Internal server error |  |

## 4. Get All Address Book API

To get all the address book stored in the database.

**URL**: /api/adress-book

**Method**: GET

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Query Param | row-size | No of address books to be returned | Integer value like 10, 20 .. |
| Query Param | start-index | Cursor position to fetch the address book | Integer value like 1, 2, 3 .. |

**Response**

JSON

List of the address book with name and communication details.

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | List of address book |
| 401 | Unauthorized |  |
| 404 | Not Found |  |
| 500 | Internal server error |  |

## 5. Get Address Book Count API

To get the total count of all address book stored in the database.

**URL**: /api/adress-book/count

**Method**: GET

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |

**Response**

JSON

{

“count”: Integer

}

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | Count of address book |
| 401 | Unauthorized |  |
| 500 | Internal server error |  |

1. Get Address Book API

To Fetch the details of a specific address book stored in the database.

**URL**: /api/adress-book/{address-book-id}

**Method**: GET

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Path Param |  | ID of the address book to get the details | UUID - Unique id of the address book account. |

**Response**

JSON

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | Address book details |
| 401 | Unauthorized |  |
| 404 | Not Found |  |
| 500 | Internal server error |  |

## 

1. Update Address Book API

To update the existing address book details like first name, last name and their communication details.

**URL**: /api/address-book/{adress-book-id}

**Method**: PUT

**Input JSON:**

{

“id”: UUID

"first\_name":"Balasubramanian",

"last\_name" : "V",

"emails":[

{

"email\_address":"vbalaas@gmail.com",

"type":{

"key":"PERSONAL"

}

}

],

"addresses":[

{

"line1" : "12131",

"line2" : "street",

"city" : "chennai",

"zipcode" : "69991",

"state\_name" : "Tamilnadu",

"type":{

"key":"HOME"

},

"country":{

"key":"INDIA"

}

}

],

"phones":[

{

"phone\_number" :"9872981231",

"type":{

"key":"PERSONAL"

}

}

]

}

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Path Param | id | ID of the address book to update the details | UUID - Unique id of the address book account |
| BODY | id | ID of the address book to update the details | UUID - Unique id of the address book account |
| BODY | first\_name | First Name of the User |  |
| BODY | last\_name | Last Name of the user |  |
| BODY | emails | List of email addresses objects |  |
|  | email\_address | Email address of the user |  |
|  | "type":{"key":"WORK" } | Email address type | 1. PERSONAL 2. WORK |
| BODY | phones | List of phone number objects |  |
|  | Phone\_number | Phone Number of the user |  |
|  | "type":{"key":"WORK" } | Phone Number type | 1. PERSONAL 2. WORK 3. ALTERNATE |
| BODY | address | List of addresses objects |  |
|  | "line1" : "12131", "line2" : "street",  "city" : "chennai", "zipcode" : "69991",  "state\_name" : Tamilnadu", | Address of the user |  |
|  | "type":{"key":"WORK" } | Address type | 1. PERSONAL 2. WORK |
|  | "country":{"key":"INDIA" } | Country of the user |  |

**Response**

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success |  |
| 401 | Unauthorized |  |
| 409 | Conflict | Error messages like email address already exists and invalid value for a field |
| 404 | Not found | Error when the metadata is not found in the database. |
| 500 | Internal server error |  |

1. Delete Address Book API

To delete an address book details stored in the database.

**URL**: /api/address-book/{id}

**Method**: DELETE

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
| Header | Content-Type | Define the http content type | application/json |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Path Param | id | ID of the address book to get the details | UUID - Unique id of the address book account. |

**Response**

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | Address book deleted successfully |
| 401 | Unauthorized |  |
| 404 | Not Found |  |
| 500 | Internal server error |  |

## 

## File Upload API

To upload the image or any file and then map it to an account.

**URL**: /api/asset

**Method**: POST

**Input binary:**

{

“file\_name”: String,

“file” : “String”

}

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Value |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| BODY | file | File object |  |
| Query Param | address-book-id | Id of the address book | Guid |

**Response**

JSON

{

"id": "b983cea4-1eeb-4bee-8b09-3d2377171c38"

}

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | JSON with id, will be used to map with account. |
| 401 | Unauthorized |  |
| 500 | Internal server error |  |

## 10. Download File API

To download the file

**URL**: /api/asset/{adress-book-id}

**Method**: GET

**Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Attribute | Description | Possible Value |
|  |  |  |  |
| Header | Authorization | Send the authorization JWT Bearer token to access the API | “Bearer : <token\_value>” |
| Path Param | id | ID of the asset | UUID - Unique id of the file stored in db |

**Response**

JSON

{

"id": "b983cea4-1eeb-4bee-8b09-3d2377171c38",

“file\_name” : String,

“file\_type” : String,

“file\_content: : String

}

|  |  |  |
| --- | --- | --- |
| Status Code | Reason | Response Message |
| 200 | Success | File object |
| 401 | Unauthorized |  |
| 404 | Not Found |  |
| 500 | Internal server error |  |

## 