**Friend Management Api**

This is an application with a need to build its own social network, "Friends Management" is a common requirement which usually starts off simple but can grow in complexity depending on the application's use case. Usually, the application comprised of features like "Friend", "Unfriend", "Block", "Receive Updates" etc.

**Technology Choice**

**Spring Boot**

1. Spring Boot allows easy setup of standalone Spring-based applications.
2. Ideal for spinning up microservices and easy to deploy.
3. Database – H2 (In memory DB)

**Swagger**

1. Swagger is a framework for describing API using a common language that everyone can understand.
2. The Swagger spec standardizes API practices, how to define parameters, paths, responses, models, etc.

**PCF**

1. Cloud Foundry is an open-source platform as a service (PaaS), which provides platform to deploy microservices.
2. **Cloud Foundry** is designed to be configured, **deployed**, managed the services.

**Deployment to PCF**

The Application is deployed on Pivotal cloud and can be accessed via the below URL and the path for all the ape is /api <https://sps-demo-new.cfapps.io/friendmgt/api/>

For example: To access /create endpoint, the URL should be:

### <https://sps-demo-new.cfapps.io/friendmgt/api/create>

Swagger UI is configured for the app and it is available [https://sps-demo-new.cfapps.io/friendmgt/swagger-ui.html#!/friend-mgmt-api-controller](https://sps-demo-new.cfapps.io/friendmgt/swagger-ui.html%23!/friend-mgmt-api-controller)

**List of REST Endpoints and Explanation**

1. Create friend list of a person
   * Path: /create
   * Input:

{

"requestor":"lisa@example.com",

"target":"lina@example.com"

}

Sample Output:

{

"status": "Success",

"description": "Successfully connected"

}

1. Returns a list of friends of a person.
   * Path: / friendlist
   * Input:

{

"email":"lisa@example.com"

} Sample Output:

{

"status": "Success",

"count": 1,

"friends": [

"lina@example.com"

]

}

* + Defined Errors:
    - 400: Occurs when invalid email provided in the request.
    - 503: Occurs when the email address in the request is not valid (Not matched with the Regex)

1. Returns list of common friends of two persons
   1. Path: /friends
   2. Input:

{

"friends”: [

"john@example.com",

"lina@example.com"

]

}

Output:

{

"status": "Success",

"count": 1,

"friends": [

"lisa@example.com"

]

}

Defined Errors:

400: Bad Request, [if email not available].

1. **Subscribe to updates from an email address** 
   1. Path: /subscribe
   2. Input:

{

"requestor":"lisa@example.com",

"target":"lina@example.com"

}

Output:

{

"status": "Success",

"description": "Subscribed successfully"

}

Defined Errors:

* + 1. 400: Invalid request

1. **API to block updates from an email address**
   1. Path: / unsubscribe
   2. Input:

{

"requestor":"lisa@example.com",

"target":[john@example.com](mailto:john@example.com)

}

Output:

{

"status": "Success",

"description": "Unsubscribed successfully"

}

Defined Errors:

* + 1. 400: Invalid request

1. **API to retrieve all email addresses that can receive updates from an email address** Path

Path: / updatelist

Input:

{

"sender":"lisa@example.com",

"text": " Hi! @abc@example.com"

}

Output:

{

"status": "Success",

"friends": [

"john@example.com",

"andy@example.com",

"lina@example.com",

"abc@example.com"

]

}

Defined Errors:

* + 1. 400: Invalid request

**Database**

The Database is pre-populated with 4 persons for testing purpose, also the data can be found from the SQL script file which is placed inside the code repository. I have used in memory database (H2).

Below are table snap shots for this development task:

Table Name: FRIENDMANAGEMENT

|  |  |  |
| --- | --- | --- |
| id | int | PK |
| EMAIL | Varchar | Null |
| FRIEND\_LIST | VARCHAR | Null |
| SUBSCRIBER | VARCHAR | Null |
| SUBSCRIBEDBY | VARCHAR | Null |
| UPDATED | VARCHAR | Null |
| UPDATED\_TIMESTAMP | DATE | Not Null |

Table Name: UNSUBSCRIBE

|  |  |  |
| --- | --- | --- |
| ID | INT | PK |
| REQUESTOR\_EMAIL | VARCHAR | Null |
| TARGET\_EMAIL | VARCHAR | Null |
| SUBSCRIPTION\_STATUS | VARCHAR | Null |