Description

designing a pretty trivial module of a social network- the connections/friends(two-way links, unlike the followers pattern of twitter) module.

Requirements

Please follow below points for achieving the required functionality:

* Users should be able to sign up (ask for as much/as less information as seems appropriate).
* Users should be able to add connections/friends.
* On signing-up the user should get friend/connection recommendations. Choose any appropriate recommendation criteria.

New Rest service is developed API endpoints exposed through swagger.

POST API will be created one for user signup and Add as a friend

GET API is written to get friend recommendations

Below mentioned details will be posted to API from Sign up:

* Firstname
* Lastname
* Timestamp
* email
* age
* gender

Above mentioned details will be stored in the Json .

Technology Stack

**Languages**: Python   
**Frameworks**: Flask, connexion, Swagger-ui-bundle  
**Platforms**: Ubuntu 16.04,Centos 7.

Design Considerations

1. Functionality
2. Performance
3. Extensibility
4. Adaptability
5. Scalability

Design overview

The design architecture is typical web application architecture, where in there will be request from the end users and from the server response will be written, to implement this use case

Connexion and flask module is used.

Connexion allows you to write these specifications, then maps the endpoints to your Python functions. This is what makes it unique from other tools that generate the specification based on your Python code. You are free to describe your REST API with as much detail as you want and then Connexion guarantees that it will work as you specified

Endpoint Routing to Your Python Views

Connexion uses the operationId from each [Operation Object](https://github.com/swagger-api/swagger-spec/blob/master/versions/2.0.md#operation-object) to identify which Python function should handle each URL

Connexion will automatically identify that your view function expects an argument named *message* and will assign the value of the endpoint parameter *message* to your view function

This view function the business logic is written which will return you the json response if it is a get call if it is Post call it will Add new data to json object

Proposed Workflow:

Rest API

Web application

Store/get data as json

Deploying application

This service can be run in vm or a bare metal server or any cloud ec2 instance

Setup a required package and and run a deployment a script which will start the python application you can access the swagger rest end point.

If it is cloud application to access swagger from browser option one we can use public IP or using classing load balancer create a path based routing