Sponsor Humanity API version 1

23 september 2021 - DRAFT
27 October 2021 - updated some API details & added person API
14 December 2021 - added JSON responses for all API's
31 December 2021 - added API for adding, updating, and retrieving person records. Change password.

Introduction

A new need has arisen to provide an API to the SH firebase database. The SH mobile application will use these API's to access the three firebase collections: Settings, Offers, and Requests. The API will be stateless and each API request is autonomous, therefore the HTTP protocol will be used.

This document is the first version of the API (1).

Goals

1. Develop API for the SH mobile using HTTP protocol.
2. Ensure a high-level of security between the mobile app (front end) and the API server (back end).
3. Initially provide the ability to
    a. insert offers and request records collections.
    b. Retrieve records from the settings collection.

Required Use Cases

1. SH sends a request to insert (PUT) an OFFER using an unencrypted authentication method. API authorizes the request, validates the arguments sent from the app, and returns an appropriate completion code.
2. SH sends a request to insert (PUT) a REQUEST using an unencrypted authentication method. API authorizes the request, validates the arguments sent from the app, and returns an appropriate completion code.
3. SH sends a request to retrieve (GET) a SETTINGS using an unencrypted encrypted authentication method. API authorizes the request, then returns the SETTINGS in JSON format with an appropriate completion code.

4. SH sends a request to update (PUT) a PERSON using an unencrypted method. API authorizes the request, then returns the PERSON in JSON format with an appropriate completion code.
5. SH sends a request to add (PUT) a PERSON using an unencrypted method. API authorizes the request, then returns the PERSON in JSON format with an appropriate completion code.
6. SH sends a request to retrieve (GET) a PERSON using an unencrypted method. API authorizes the request, then returns the PERSON in JSON format with an appropriate completion code.
7. SH sends a request to update (PUT) the password in the Authentication collection using an unencrypted method. API authorizes the request, then returns the status in JSON format with an appropriate completion code.

Additional Use Cases

In order to minimize exposure to the SH Database, we will add additional use cases as needed. This will include GET, PUT, UPDATE, and DELETE API requests to all collections.

Security

By always using SSL, the authentication credentials can be simplified to a randomly generated access token. The token is delivered in the username field of HTTP Basic Authentication

Other Considerations (potential future)

1. Although basic auth is good enough for most of the APIs and if implemented correctly, it’s secure as well – yet we may want to consider OAuth as well in the future. The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its behalf.
2. Using HTTP endpoints that are in the sponsorhumanity.org domain is the preferred option, however, a discussion about the domain might be warranted. Please see https://flask.palletsprojects.com/en/2.0.x/deploying/.

   For the purposes of this document, the sponsorhumanity.org domain will be used.

API definition

1. Responses and completion codes

Responses from the API will be of the format listed below.

```
{
        "code": "<string>",
        "reason": "<string>",
        "timestamp": "<number>"
}
```

The SH API code will return status using this format, however, there are a number of responses that are not under the control of the API code and the SH mobile application must be able to handle these conditions as well. These responses could come from the server or from the flask framework and aren't under the control of the SH API such as 500 Internal Server Error or 401 Unauthorized.

The top 10 completion codes will be supported as appropriate and defined in https://www.restapitutorial.com/httpstatuscodes.html.

"Top 10" HTTP Status Code. More REST service-specific information is contained in the entry

2. SETTING (GET)

Request URL	See example below
REQUIRED PARAMETERS	
none	
OPTIONAL PARAMETERS	
none	

Sample results (JSON format)

```
{
        "max_amount_per_person": "100",
        "max_household_size": 4
}
```

Localhost:5000\setting

https://lit-oasis-95955.herokuapp.com/setting

3. OFFERS (PUT)

Request URL	See example below
REQUIRED PARAMETERS	
sponsor-humanity-offer-amount	Amount offered > 10
sponsor-humanity-uid	Uid (existence validation)

http://127.0.0.1:5000/offers?sponsor-humanity-offer-amount=100&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2

https://lit-oasis-95955.herokuapp.com/setting/offers?sponsor-humanity-offer-amount=100&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2

https://lit-oasis-95955.herokuapp.com/offers?sponsor-humanity-offer-amount=100&sponsor-humanity-uid=3EXO9QmqqdZ8S65kyGSHxC01NwU2

4. REQUESTS (PUT)

Request URL	See example below
REQUIRED PARAMETERS	
sponsor-humanity-requested-amount	Amount requested > 25 and less than Max_amount_per_person in setting collection.
sponsor-humanity-requestor-email	Requestor’s valid email address. Must exist in the person collection.
sponsor-humanity-phone-number	Requestor’s valid telephone number.

http://127.0.0.1:5000/requests?sponsor-humanity-requested-amount=80&sponsor-humanity-requestor-email=jobo132017@gmail.com&sponsor-humanity-phone-number=9784603896

https://lit-oasis-95955.herokuapp.com/requests?sponsor-humanity-requested-amount=80&sponsor-humanity-requestor-email=jobo132017@gmail.com&sponsor-humanity-phone-number=9784603896

5. ADD PERSON (PUT)

Request URL	See example below
REQUIRED PARAMETERS	
sponsor-humanity-email	Valid email address. Must exist in the authorization collection. No duplicates!
sponsor-humanity-first-name	Person’s first name.
sponsor-humanity-last-name	Person’s last name.
sponsor-humanity-display-name	How a person will be displayed.
sponsor-humanity-send-email-on-match	True=send emails, False = do not send email.
sponsor-humanity-house-size	Valid house size (>=1 and <= settings.max_household_size)
sponsor-humanity-vendor-preference	“Walmart”, “Kroger”, “Publix”, “Whole Foods” (* required for requestor, not required for donor)
sponsor-humanity-phone-number	Person valid telephone number.
sponsor-humanity-uid	User UID from SH Authentication collection
sponsor-humanity-url	Used to store images
sponsor-humanity-geopoint	Location of person’s phone

sponsor-humanity-email-verified	Bool: Has email been verified,

sponsor-humanity-is-anonymous	Bool

sponsor-humanity-status	NO ARGUMENT. “ACTIVE” by default

http://127.0.0.1:5000/person?sponsor-humanity-email=jobo132017@gmail.com&sponsor-humanity-first-name=Fred&sponsor-humanity-last-name=Flinstone&sponsor-humanity-display-name="Mr. Clay Bogusky"&sponsor-humanity-send-email-on-match=True&sponsor-humanity-house-size=3&sponsor-humanity-vendor-preference="WholeFoods"&sponsor-humanity-phone-number=908-722-7146&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2&sponsor-humanity-url=”www.clay.url”&sponsor-humanity-email-verified=FALE&sponsor-humanity-geopoint=”NJ”&sponsor-humanity-is_anomymous=FALSE

http://127.0.0.1:5000/person?sponsor-humanity-email=jobo132017@gmail.com&sponsor-humanity-first-name=Fred&sponsor-humanity-last-name=Flinstone&sponsor-humanity-display-name="Mr. Clay Bogusky"&sponsor-humanity-send-email-on-match=True&sponsor-humanity-house-size=3&sponsor-humanity-vendor-preference=Walmart&sponsor-humanity-phone-number=908-722-7146&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2&sponsor-humanity-url=”www.clay.url”&sponsor-humanity-email-verified=true&sponsor-humanity-geopoint=NJ&sponsor-humanity-is_anomymous=FALSE

https://lit-oasis-95955.herokuapp.com/person?sponsor-humanity-email=jobo132017@gmail.com3&sponsor-humanity-first-name=Fred&sponsor-humanity-last-name=Flinstone&sponsor-humanity-display-name="Mr. Clay Bogusky"&sponsor-humanity-send-email-on-match=True&sponsor-humanity-house-size=3&sponsor-humanity-vendor-preference=Walmart&sponsor-humanity-phone-number=908-722-7146&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2&sponsor-humanity-url=”www.clay.url”&sponsor-humanity-email-verified=true&sponsor-humanity-geopoint=NJ&sponsor-humanity-is_anomymous=FALSE

6. UPDATE PERSON (PUT)

Request URL	See example below

PARAMETERS	
sponsor-humanity-uid	UID of person in person table. ***(REQUIRED)***
sponsor-humanity-first-name	Person’s first name. (Can’t be blank)
sponsor-humanity-last-name	Person’s last name. (Can’t be blank)
sponsor-humanity-send-email-on-match	True=send emails, False = do not send email. (Can’t be blank)
sponsor-humanity-house-size	Valid house size (>=1 and <= settings.max_household_size) (Can’t be blank)

http://127.0.0.1:5000/update_person?sponsor-humanity-first-name=Donald&sponsor-humanity-last-name=Trump&sponsor-humanity-send-email-on-match=True&sponsor-humanity-house-size=3&sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2

7. UPDATE PASSWORD (PUT)

Updates Firebase password given a valid Firebase user ID.

Request URL	See example below
PARAMETERS	
sponsor-humanity-uid	UID of person in person table. ***(REQUIRED)***
sponsor-humanity-password	Person’s new password. ***(REQUIRED and*** Can’t be blank) Minimum of 6 characters.

http://127.0.0.1:5000/update_pwd?sponsor-humanity-uid=3EXO9QmggdZ8S65kyGSHxC01NwU2&sponsor-humanity-pwd=foobar@123!

8. GET PERSON (GET)

Request URL	See example below
PARAMETERS	
sponsor-humanity-uid	UID of person in person table. ***(REQUIRED)***

http://127.0.0.1:5000/get_person?sponsor-humanity-uid=bzfYeQQXHmeRzRpATuPWyg2ZDig2

9. Default route

A default route is necessary and is of a simple form using the flash framework.

```
@app.route('/', defaults={'path': ''})
@app.route('/<path:path>')
def catch_all(path):
   return 'Welcome to the Sponsor Humanity API.'
```