
apostello Documentation

Release 0.1

Dean Montgomery

Oct 18, 2018

Contents

1	Introduction
----------	---------------------

3

free SMS communication software for your church

CHAPTER 1

Introduction

apostello is a web app designed to enable a church to communicate with its congregation via SMS. Messages can be sent, received and automatically replied to.

apostello was written for use at St Columba's Free Church as we couldn't find a cheap SMS solution that fit all our needs and has been running in production since early 2015.

Try out the [demo site](#) or the [getting started](#) help.

Pull requests are encouraged. Please open an issue before working on anything other than minor bug fixes.

Features

- Send messages to ad-hoc or predefined groups.
- Automatically respond to incoming messages that match keywords.
- Track sign ups, questions, etc using keywords.
- Manage access permissions - you can let anyone in you church have read only access.
- Spending safety net - you can set a limit (in \$) for each person. No individual SMS can be sent that will cost more than this.
- Block auto replies to specific contacts.
- Receive daily digest emails of incoming messages.
- Live "wall" - curate and display incoming messages on a big screen. Great for a Q&A.
- Post all messages to a slack channel.
- Import contacts (CSV or [Elvanto](#)).
- Usage overview dashboard
- Sign up and login by email or with a Google account

Keywords

All incoming messages are expected to start with a keyword. This allows for custom replies to be sent back and is useful for tracking things like event sign ups or questions for a Q&A.

Special Keywords

There is a small set of reserved keywords, some by Twilio and some by apostello.

- Twilio's reserved keywords can be found [here](#).
- apostello also reserves the keyword **name**. Any SMS that matches **name** will be parsed for a name and used to update the name associated with that contact. If the parsing fails, then the contact is sent another message asking them to try again.

Custom Keywords

You can create as many keywords as you like, and each keyword comes with the following features:

- Custom response
- Number of response tracked
- Ability to mark each response as “Requires Action” or “Dealt With”
- Archiving of old responses
- Activation time and a too early custom response
- Deactivation time and a too late custom response
- Ability to lock a keyword to certain users (staff users can always see all keywords)
- Daily email digests - send digest of today's responses every night
- CSV export of messages

Keyword Matching

A case insensitive greedy match is performed on the start of every incoming message up to the first space in the message (none alphanumeric characters are also ignored if they begin the message). For example, the messages *connect John Calvin*, *connected John Calvin* and *“connect” John Calvin* would all match the keyword *connect*, but only the second message would match the keyword *connected*.

There is an additional check when creating new keywords - you cannot create a keyword that cause a match collision. For example, if *connect* is a keyword, you will be unable to create *con*, *conn* or *connected*, etc.

Getting Started

There are a number of ways you can deploy apostello:

- **Recommended:** Using the [ansible playbook](#) included in the repo
- One click [Heroku](#) deploy
- One click [Digital Ocean](#) deploy

Prerequisites

- *Required:* A domain name and a server (or you can use Heroku instead).
- *Required:* A [Twilio](#) account with a purchased number.
- *Required:* An SMTP server or [Mailgun](#), etc account for sending email notifications. You can setup apostello without this, but it will not be able to send emails. See [Email Setup](#) for more details.
- *Optional:* A web app registered for authentication with a Google account
- *Optional:* An [Elvanto](#) API Key for importing Elvanto groups.

- *Optional:* A [rollbar](#) account for error logging.

First Run

After you have successfully installed apostello there are a few more steps to finish setup.

- Open your instance of apostello, you will be redirected to the initial setup page
- This page lets you create a new admin user.
- You can check various settings. If something doesn't look right change the setting and reload the page to check it has updated
- If you have issues sending an email or SMS, you will be shown the corresponding error message. If you need help, please get in touch
- Once you have created your account, refresh the page and login with you email and password
- You can now continue to set up apostello: import contacts, start sending messages, publicise your number, etc

Configuration

Once you have successfully logged in, navigate to the Site Configuration page (click the menu button, then Site Configuration). There are various settings on this page, but the first thing to do is to configure Twilio and email.

- Just fill in your credentials and submit the form
- Now you need to [setup Twilio](#) so you can receive messages
- You can also test your setup using the link provided

Other options:

- You can edit the default responses by going to *Menu -> Default Responses*
- If you want to let users sign in with Google, then you need to follow the steps [here](#)
- Any future users will be able to use the normal sign up page. If you do not whitelist any domains, you will need to approve new users manually before they can do anything. Please be extremely careful with the whitelisting setting - if you set it to a domain that you have no control over (e.g `gmail.com`), then anyone will be able to access your instance of apostello
- If you need to approve new users, use the User Permissions page

Twilio Setup

Once you have apostello setup we need to tell Twilio what url to talk to when it receives an SMS:

- Open <https://www.twilio.com/user/account/messaging/phone-numbers>
- Click the number you are using in apostello and a popup should appear
- Click the "Messaging" tab if it is not already selected
- Select "Configure with URL"
- Ensure the HTTP method is set to POST
- In the "Request URL field" add the url to your server, followed by `/sms/`. If you are using Heroku it may look like `https://apostello-demo.herokuapp.com/sms/` or if your site is hosted at `https://sms.example.com`, your URL would be `https://sms.example.com/sms`
- Click save

Now you should be able to test your setup - send a message to your number and you should receive an automated reply. If not, raise an [issue](#).

Email Setup

Emails are sent for a number of reasons by apostello:

- Email verification on sign up
- Daily keyword digests
- Warnings and notifications on some events are sent to the “office email”

apostello needs a mail server to send these emails. We recommend using [Mailgun](#) which allows you to send 10,000 emails for free each month.

Use the *Site Configuration* form to tell apostello about your mail server.

Deployment

Find out more about how you can deploy apostello below, then check out the *getting started* guide.

Deploying to Heroku

Heroku lets you run your application in the cloud. If you click the button below, an instance of apostello will be created for you. It is possible to run apostello on the free tier of Heroku, however Heroku requires your app to sleep for 6 hours a day when you are on the free tier and will sleep your app after inactivity. It can take a few seconds for an app to wake up from its sleeping state. This may mean you miss some incoming messages from Twilio. You can either upgrade to a paid plan or use another deploy method instead. Despite the limitations above, Heroku can be a quick and easy way to evaluate if you want to invest the time into setting up and using apostello at your church.

Upgrading

Unfortunately there is no easy way to keep an app deployed with deploy button updated without using the command line. Fortunately, it is pretty simple to do.

We are going to use [Git](#) to grab a copy of apostello and then use the [Heroku toolbelt](#) to create a new build and push it. If you don't want to use Git, you can skip those steps and just download a zip of the source code from Github instead ([source code](#)).

Before we begin, install the Heroku toolbelt and Git. Then run the command `heroku login` to login.

The first time you update after deploying to Heroku you need to run the following commands:

```
# install the builds plugin (you only need to do this once)
heroku plugins:install heroku-builds
# now, let's grab a copy of apostello
# if you don't want to use git, then download the code manually
git clone https://github.com/monty5811/apostello.git
cd apostello
# view the different releases:
git tag
# grab the latest release (skip this step to use the master branch)
git checkout v<insert-latest-version-here>
# now we want to create a new build:
# substitute your app name into the command e.g.
# heroku builds:create -a apostello-demo
heroku builds:create -a <your-app-name-here>
# you should see a bunch of text scroll by and a successful update
# if you run into any problems, please get in touch

# you can check your build history:
```

```
heroku builds -a <your-app-name-here>
# if a build fails, you can roll back to a previous build in the
# Heroku dashboard
```

Subsequent updates are quicker:

```
# move to the apostello folder, then:
git fetch origin
# view the different releases:
git tag
# Either: grab the latest release:
git checkout v<insert-latest-version-here>
# Or: use the latest code in the master branch:
git pull origin master && git checkout master
# Then push the code to Heroku
heroku builds:create -a <your-app-name-here>
# you may also need to run any migrations:
heroku run -a <your-app-name-here> ./manage.py migrate
```

Do be aware that if you make any other changes in this folder, they will be pushed to Heroku. You could use this to customise the favicons, or change the configuration in `settings/heroku.py`, for example.

Deploying with Ansible

Ansible is a tool to automate deployments. An ansible playbook to deploy apostello is bundled in the git repo.

In order to use the playbook, you need a server to point it towards. Additionally, if you want to use Let's Encrypt to obtain an SSL certificate (enabled by default), you will need a domain that points to your server.

If you do not want to customise the playbook, you should point it at a dedicated server as it may delete or mess up other configured applications on any server you point it at. For example, the playbook may overwrite your nginx config.

Instructions

To run the playbook

```
git clone https://github.com/monty5811/apostello.git
cd apostello
# checkout a specific version:
git checkout |vversion|
cd ansible
python2 -m virtualenv venv --no-site-packages
. venv/bin/activate
pip install ansible==2.5.0.0
cp env_vars/example.yml env_vars/my_site_name.yml
# fill in the credentials in the new yml file
ansible-vault encrypt my_site_name.yml
# replace "env_vars/example.yml" in production.yml
# run the playbook
ansible-playbook --ask-vault-pass -i [IP or Domain name of server], production.yml
```

After the playbook finishes you should have your own apostello server - follow the first run steps on the [Getting Started](#) page to finish.

Let's Encrypt

An SSL certificate can be setup by setting `setup_lets_encrypt` to `yes` in the file `production.yml`. This is disabled by default as it seems to fail on brand new setups. If you want enable this, run the playbook without

lets encrypt, then turn the setting on and run the playbook again.

Deploying with Digital Ocean One-Click

Install

Assuming you have a [Digital Ocean](#) account with an SSH key setup, click the following button to deploy apostello: This will create a droplet, download apostello and run the *ansible playbook* to setup apostello.

Once you have an instance running you will need to configure it.

Configuration

Log in to your server and edit the config file, updating the file according to the *First Run* instructions:

```
nano /home/apostello/apostello-install/ansible/env_vars/example.yml
```

Then rerun the installer:

```
./home/apostello/apostello-install/scripts/ansible_install.sh
```

Upgrading

To upgrade to a new version of apostello, you need to edit the install script:

```
nano /home/apostello/apostello-install/scripts/ansible_install.sh
```

and replace `AP_VER=vx.x.x` with a new version, then rerun the install script:

```
./home/apostello/apostello-install/scripts/ansible_install.sh
```

Switching to the Ansible Deploy Method

Having to log in to the droplet to update can be cumbersome. Since the one-click installer uses the ansible playbook method under the hood, it is easy to switch by setting up ansible on your own machine.

Set up:

```
# clone apostello:
git clone https://github.com/monty5811/apostello.git
cd apostello
# checkout a specific version:
git checkout |vversion|
# create a virtualenv (you need python2 already installed)
cd ansible
python2 -m virtualenv venv --no-site-packages
. venv/bin/activate
# install ansible
pip install ansible==2.5.0.0
```

Now you need to copy your secrets into `env_vars/example.yml` (or copy the file from your existing server).

Run the upgrade:

```
ansible-playbook --ask-vault-pass -i [IP or Domain name of server], production.yml
```

Subsequent upgrades:

```
cd <path-to-apostello-folder>
git fetch
git checkout <version-tag>
cd ansible
. venv/bin/activate
ansible-playbook --ask-vault-pass -i [IP or Domain name of server], production.yml
```

Frequently Asked Questions

What is the character limit for an SMS?

By default, each SMS is limited to 160 characters. Twilio can send messages up to 1600 characters long, but it will charge you multiple times, e.g. a 1600 character message will be charged the same as 10 messages.

The character limit can be increased on the Site Configuration page (Menu --> Site Configuration).

How do I “mail merge” my messages?

Any occurrence of `%name%` in outgoing messages or keyword replies will be replaced by the first name of the contact. For example, `Hi %name%! becomes Hi John!`

How do I cancel scheduled messages?

Go to Menu --> Scheduled Messages, and click the cancel button for any scheduled messages you wish to cancel.

How do I setup the slack integration?

Create a new Slack incoming [webhook](#). Then open the Site Configuration page (Menu --> Site Configuration) and paste the hook URL you were given by Slack. Click save and all your incoming messages should show up in the Slack channel you picked when creating the webhook.

How do I prepopulate the send SMS form?

The send SMS form will read url parameters. So an URL `apostello-demo.herokuapp.com/send/adhoc/?content=test&recipients=[1,2]` will prepopulate the SMS form with `test` in the content box and the recipients 1 and 2 (you can get these numbers from the api or the url when editing a contact) in the send field.

The URLs in my emails are incorrect

You may need to let apostello know what your domain is. You can do this by opening `<your domain>/admin/sites/site/`, click on the first entry and update the domain name field, then click save.

How can I get rid of old messages?

Open the Site Configuration page (Menu --> Site Configuration), scroll down to find the SMS Expiration section. You can set a hard cut off date: any messages before this date will be purged from the system. Or you can choose the number of days to hold on to messages: any messages older than this number of days will be purged. The purge is run daily. You can use this to hide old messages you no longer care about or to make sure you stay within the limits of the Heroku hobby database.

Demo Site

There is a demo site available at <https://apostello-demo.herokuapp.com>

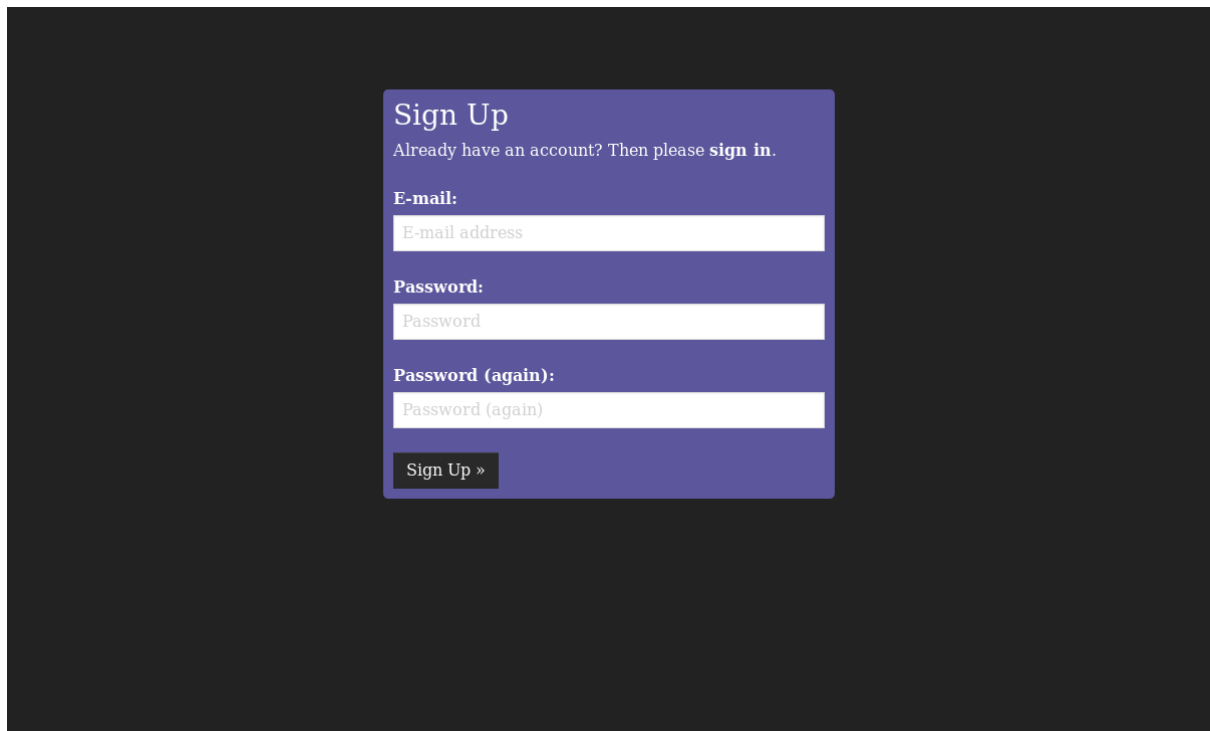
You can login with the email `test@example.com` and the password `apostello`.

Note that:

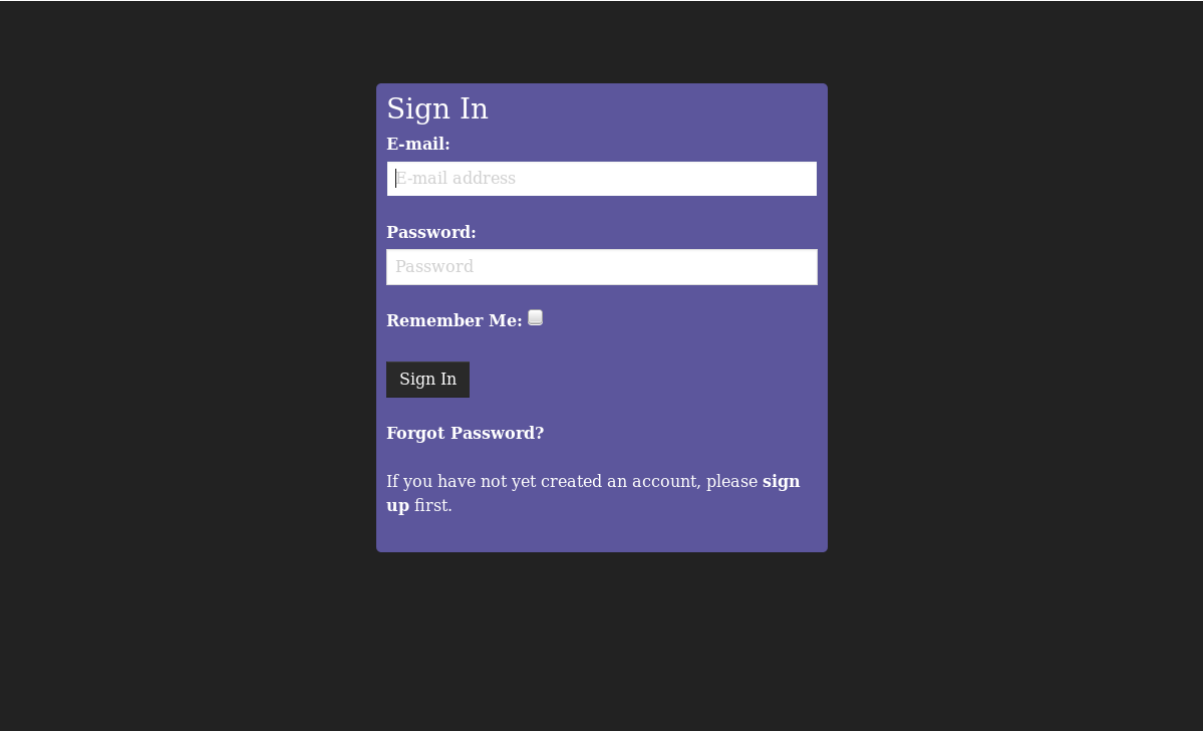
- All the data is reset every hour
- You cannot send or receive messages as there are no Twilio credentials
- No emails can be sent either (so signing up as new user will not work)
- The demo site is running on the free tier of Heroku, so it may be unavailable as it needs to sleep 6 hours everyday

Screenshots

Signup

A screenshot of a web application's sign-up form. The form is a light purple rectangle centered on a dark grey background. It has a title 'Sign Up' in white, followed by a link 'Already have an account? Then please **sign in**.' Below this are three input fields: 'E-mail:' with placeholder 'E-mail address', 'Password:' with placeholder 'Password', and 'Password (again):' with placeholder 'Password (again)'. At the bottom is a dark button with the text 'Sign Up »' in white.

Login



Home

apostello

HomeMenu

New Keyword

New Contact

New Group

Welcome to apostello.

Please bear in mind that each message you send costs **money**, so please do not send frivolous messages.

Additionally, try not to send too many messages - do not abuse this system, people do not like being bombarded with messages all day.

FAQs/Help

In

Out

1

0.9

0.8

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

31 Jul

01 Aug

02 Aug

03 Aug

04 Aug

05 Aug

06 Aug

07 Aug

08 Aug

09 Aug

10 Aug

11 Aug

12 Aug

13 Aug

14 Aug

15 Aug

16 Aug

17 Aug

18 Aug

19 Aug

20 Aug

21 Aug

22 Aug

23 Aug

24 Aug

25 Aug

26 Aug

27 Aug

28 Aug

29 Aug

30 Aug

Send to Individuals

apostello

HomeMenu

New Keyword

New Contact

New Group

Send

Send a message to a single person or to an ad-hoc group of people:

Content *

Recipients *

Filter...

John Calvin

John Owen

Scheduled Time

Send to Group

apostello

HomeMenu

New Keyword

New Contact

New Group

Send

Send a message to a single person or to an ad-hoc group of people:

Content *

Recipient_group *

Filter...

(E) All - Imported from Elvanto(\$0.08)

(E) England - Imported from Elvanto(\$0.04)

(E) Geneva - Imported from Elvanto(\$0.04)

Note that empty groups are not shown here.

Scheduled Time

12

Chapter 1. Introduction

Recipients

apostello

Home Menu

New Contact

Archived

Contacts

Contacts

Filter...

Name	Last Message	Received
John Calvin		Archive
John Owen		Archive

Recipient Edit

apostello

Home Menu

Remove

Edit Contact

Name

First Name *

John

Last Name *

Calvin

Number *

+447902537509

Cannot be our number, or we get an SMS loop.

☐ Do not reply

Tick this box to disable automated replies for this person.

Update

Filter...

From	Keyword	Message	Time
John Calvin	No Match	this doesn't match?	20:46 - 23 Dec

Keywords

apostello

HomeMenu

New Keyword

Archived Keywords

Keywords

Filter...

	Matches	Description	Auto Reply	Status		
connect	1	Connect sign up	Thanks for signing up :-)	Active	Edit	Archive
test	2	Test keyword	Thanks for testing!	Active	Edit	Archive

Keyword Edit

apostello

HomeMenu

Replies (2)

Archived Replies (0)

Remove

Edit Keyword

Keyword *
test

Keyword Description *
Test keyword

☐ **Disable all replies**
If checked, then we will never reply to this keyword.Note that users may still be asked for their name if they are new.

Replies

Auto response
Thanks for testing!
This text will be sent back as a reply when any incoming message matches this keyword. If empty, the site wide response will be used.

Deactivated response
You are far too late for testing
Use this if you want a custom response after deactivation. e.g. 'You are too late for this event, sorry!'

Not yet activated response

Keyword Responses

apostello

Home Menu

Edit

Export responses

Replies (2)

Archived Replies (0)

test reponses

Filter...

From	Time Received	Message	Requires Action?
John Calvin	21:44 - 22 Dec	testing a silly match	✓ Dealt With
John Calvin	20:42 - 22 Dec	test Hi there!	! Requires Action

☐ Tick to archive all responses

Archive all!

Incoming Log

apostello

Home Menu

Live Updates

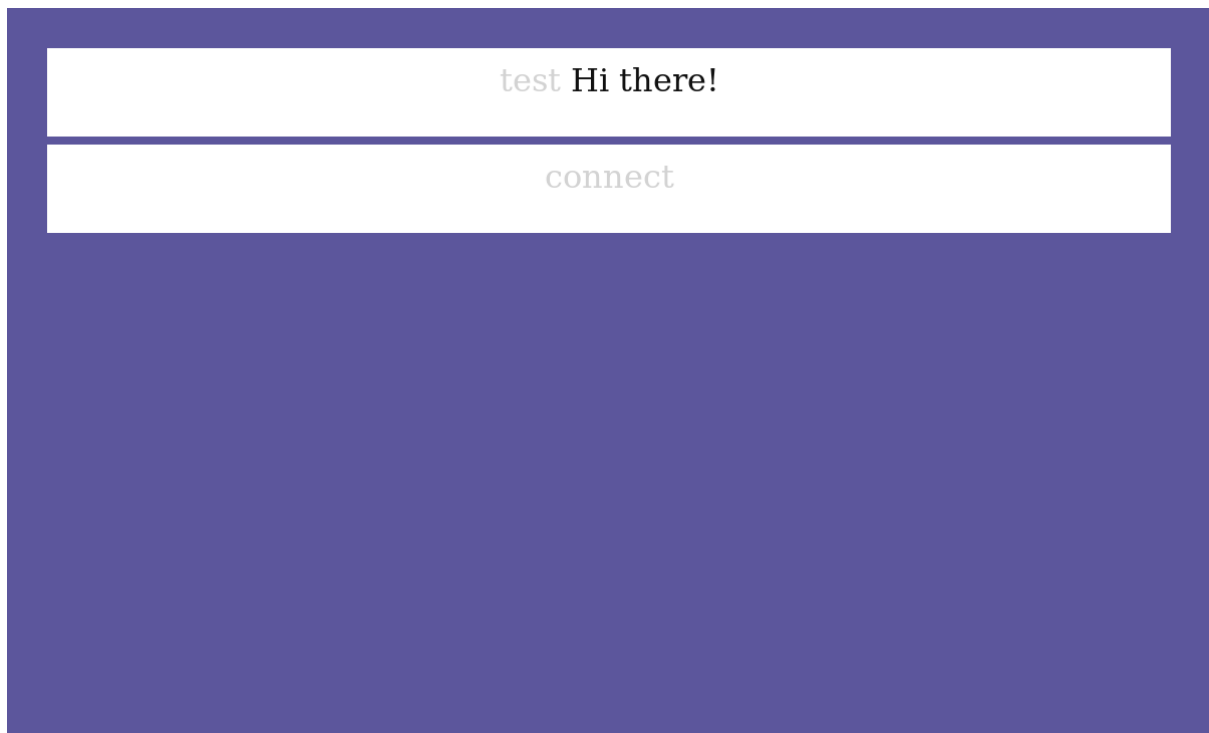
Live Curator

Incoming

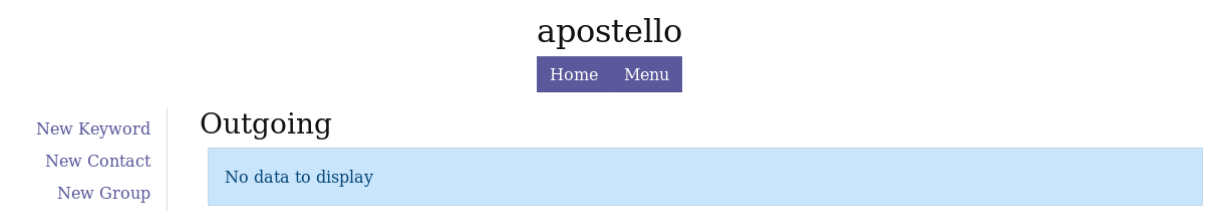
Filter...

From	Keyword	Message	Time
John Calvin	No Match	this doesn't match?	20:46 - 23 Dec
John Calvin	test	testing a silly match	21:44 - 22 Dec
John Calvin	test	test Hi there!	20:42 - 22 Dec
John Owen	connect	connect	13:00 - 24 Aug

Incoming Wall



Outgoing Log



Elvanto Sync

apostello

[Home](#)[Menu](#)

New Keyword

New Contact

New Group

Elvanto Sync

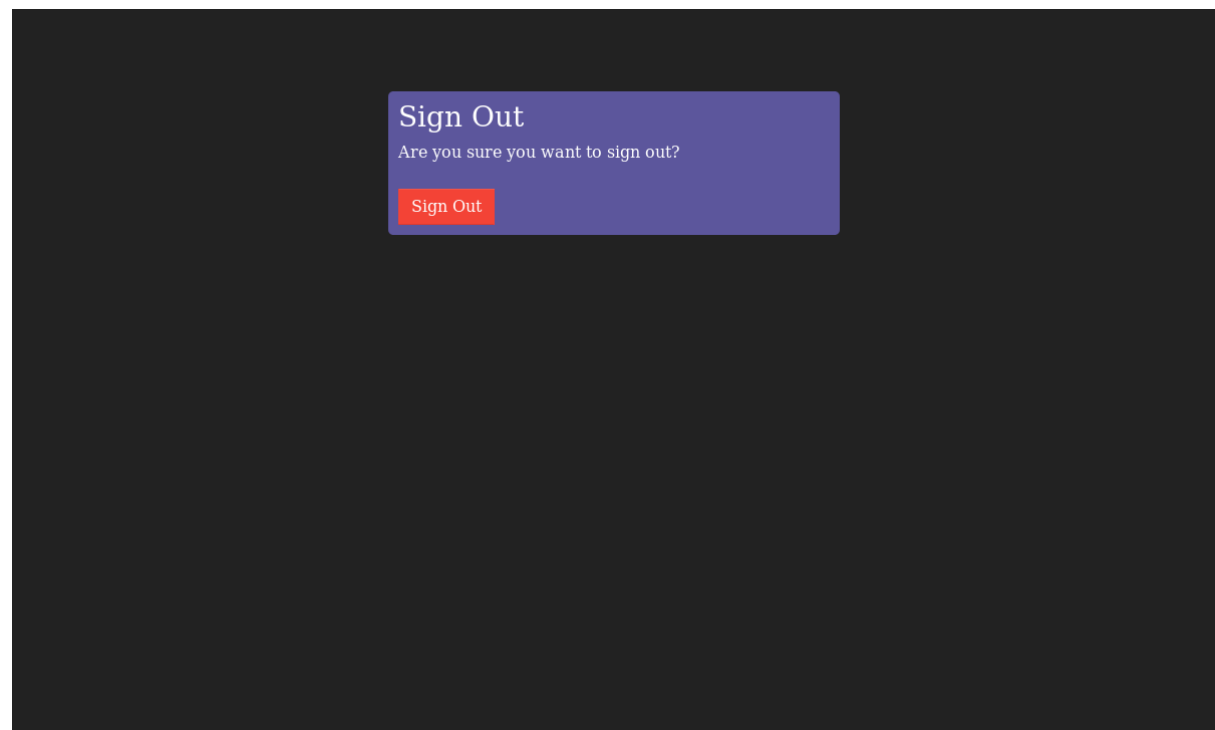
Fetch Groups

Pull Groups

Filter...

	Last Synced	Sync?
All	14:39 - 23 Dec	<div>Syncing</div>
Empty	14:39 - 23 Dec	<div>Disabled</div>
England	14:39 - 23 Dec	<div>Syncing</div>
Geneva	14:39 - 23 Dec	<div>Syncing</div>
Scotland	14:39 - 23 Dec	<div>Syncing</div>

Logout



API

An API is available for programmatic access to your instance of apostello. Currently the API is read only and experimental.

Setup

In order to access the API, you must generate an API key. Open your instance of apostello and navigate to /api-setup/, or click Menu -> API Setup, then generate an API key. Only staff users can access this page.

You can also regenerate or remove your API keys on this page.

You can change another user's API key in the admin panel.

Documentation

Documentation of the API endpoints can be found at <your-apostello-url>/api-docs/.

The API key must be included in the Authorization HTTP header (Authorization: Token <API-KEY>).

An example request might look like:

```
import requests

r = requests.get(
    url,
    headers={'Authorization': 'Token {0}'.format(API_KEY)}
)
print(r.json())
```

Contributing

Thank you for considering contributing to apostello, please read these guidelines so we can best manage things.

There are many ways for you to contribute: writing tutorials, improving the documentation, submitting bug reports or feature requests and writing code.

Code Contributions

Guidelines for code contributions:

- Unless it is a small or trivial fix, please open an issue before starting
- Any new features must include tests - if a feature requires interaction through the browser, please add selenium tests
- Please do not hit the network in tests - see how vcrpy is used in the test suite for help with this
- Please run `./scripts/run_yapf.py` before committing to maintain code style
- Please add only a single feature per pull request

Development Environment

Prerequisites

- Python 3
- Git
- Redis
- Optional: Node for frontend development
- Optional: xvfb, Geckodriver and Firefox (>47) to run browser based tests

Get Started

Fork apostello on Github ([how-to](#)), then clone the repo:

```
git clone <url-to-your-fork-here>
cd apostello
# create a branch for your feature/fix:
git checkout -b <branch-name>
```

Create a python 3.6 virtualenv and install dependencies:

```
python3.6 -m venv venv
pip install -r requirements/test.txt
```

Create a development database (this uses sqlite, if you need to reset the database, just delete db.sqlite3 and run this command again):

```
./manage.py migrate
```

Create a super user:

```
./manage.py createsuperuser # follow the prompts
```

Start the development server:

```
./manage.py runserver
```

Open your browser and go to 127.0.0.1:8000/admin and login.

Running Tests

```
pip install tox
tox # you need xvfb, firefox and geckodriver installed
tox -- -m \"not slow\" # runs only quick tests
```

Frontend

Most of the frontend is written in Elm.

Setup:

```
cd assets/
yarn install # this may take a while the first time
```

Changes must then be compiled:

```
npm run format # format elm and js code to maintain style
npm run build # regenerate all the assets
npm run watchjs # watch js and elm code for changes
npm run prodjs # build the js and elm for production
npm run elm-test # run elm tests
```

Tips

Charity/Non-profit Discounts

If you are a registered charity you can:

- Get \$500 Twilio credits and 25% rate reduction (eligibility)
- Contact [Digital Ocean](#) and apply for a discount