
The h Documentation

Release 0.0.2

Hypothes.is Project and contributors

Jun 27, 2017

Contents

| | |
|----------|-----------------|
| 1 | Contents |
|----------|-----------------|

| |
|----------|
| 3 |
|----------|

h is the web app that serves most of the <https://hypothes.is/> website, including the web annotations API at <https://hypothes.is/api/>. The Hypothesis client is a browser-based annotator that is a client for h's API, see [the client's own documentation site](#) for docs about the client.

This documentation is for:

- Developers working with data stored in h
- Contributors to h

The Hypothesis community

Please be courteous and respectful in your communication on [Slack](#), [IRC \(#hypothes.is on freenode.net\)](#), the mailing list ([subscribe](#), [archive](#)), and [GitHub](#). Humor is appreciated, but remember that some nuance may be lost in the medium and plan accordingly.

If you plan to be an active contributor please join our mailing list to coordinate development effort. This coordination helps us avoid duplicating efforts and raises the level of collaboration. For small fixes, feel free to open a pull request without any prior discussion.

Advice for publishers

If you publish content on the web and want to allow people to annotate your content, the following documents will help you get started.

Generating authorization grant tokens

Warning: This document describes an integration mechanism that is undergoing early-stage testing. The details of the token format may change in the future.

In order to allow your users (i.e. those whose accounts and authentication status you control) to annotate using a copy of Hypothesis embedded on your pages, you can ask us to register your site as a special kind of OAuth client.

We will issue you with a *Client ID* and a *Client secret*. These will allow you generate time-limited “authorization grant tokens” which can be supplied as configuration to the Hypothesis sidebar, and which will allow your users to annotate without needing to log in again. This document describes how to generate those tokens.

[RFC7523](#), “JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants”. Note that we currently only support the HS256 signing algorithm, and not the public-key RS256 signing algorithm mentioned in the RFC.

Examples

This section contains complete example code for generating a JWT in various common programming environments.

Python

We recommend using `PyJWT`:

```
import datetime
import jwt

# IMPORTANT: replace these values with those for your client account!
CLIENT_AUTHORITY = 'customwidgets.com'
CLIENT_ID         = '4a2fa3b4-c160-4436-82d3-148f602c9aa8'
CLIENT_SECRET     = '5SquUVG0Tpg57ywoxUbPPgjtK0OkX1ttipVlfBRRrpo'

def generate_grant_token(username):
    now = datetime.datetime.utcnow()
    userid = 'acct:{username}@{authority}'.format(username=username,
                                                  authority=CLIENT_AUTHORITY)

    payload = {
        'aud': 'hypothes.is',
        'iss': CLIENT_ID,
        'sub': userid,
        'nbf': now,
        'exp': now + datetime.timedelta(minutes=10),
    }
    return jwt.encode(payload, CLIENT_SECRET, algorithm='HS256')
```

Ruby

We recommend using `ruby-jwt`:

```
require 'jwt'

# IMPORTANT: replace these values with those for your client account!
CLIENT_AUTHORITY = 'customwidgets.com'
CLIENT_ID         = '4a2fa3b4-c160-4436-82d3-148f602c9aa8'
CLIENT_SECRET     = '5SquUVG0Tpg57ywoxUbPPgjtK0OkX1ttipVlfBRRrpo'

def generate_grant_token(username)
    now = Time.now.to_i
    userid = "acct:#{username}@#{CLIENT_AUTHORITY}"
    payload = {
        aud: "hypothes.is",
        iss: CLIENT_ID,
        sub: userid,
        nbf: now,
        exp: now + 600
    }
}
```

```
JWT.encode payload, CLIENT_SECRET, 'HS256'  
end
```

The Hypothesis Real Time API

Warning: This document describes an API that is in the early stages of being documented and refined for public use. Details may change, and your systems may break in the future.

In addition to the [HTTP API](#) Hypothesis has a WebSocket-based API that allows developers to receive near real-time notifications of annotation events.

Overview

To use the Real Time API, you should open a WebSocket connection to the following endpoint:

```
wss://hypothes.is/ws
```

Communication with this endpoint consists of JSON-encoded messages sent from client to server and vice versa.

Authorization

Clients that are only interested in receiving notifications about public annotations on a page do not need to authenticate. Clients that want to receive notifications about all updates relevant to a particular user must authenticate.

Server-side clients can authenticate to the Real Time API by providing an access token in an Authorization header:

```
Authorization: Bearer <token>
```

Browser-based clients are not able to set this header due to limitations of the the browser's `WebSocket` API. Instead they can authenticate by setting an `access_token` query parameter in the URL when connecting:

```
var socket = new WebSocket(`wss://hypothes.is/ws?access_token=${token}`)
```

Server messages

Each messages from the server will be either an *event* or a *reply*:

event An event is sent to clients as a result of an action taken elsewhere in the system. For example: if an annotation is made which matches one of the client's subscriptions, the client will receive an event message. All event messages have a `type` field.

reply A reply is sent in response to a message sent by the client. All replies have an `ok` field which indicates whether the server successfully processed the client's message, and a `reply_to` field which indicates which client message the server is responding to.

Clients should ignore events with types that they do not recognise, as this will allow us to add new events in future without breaking your client.

Note: We will add documentation for specific event types as we upgrade the protocol.

Sending messages

All messages sent to the server must have a numeric ID which is unique for the connection. The ID should be sent with the message in the `id` field. In addition, every message sent to the server must have a valid `type` field. See below for the different types of message you can send.

Message types

- `ping`
- `whoami`

ping

To verify that the connection is still open, clients can (and are encouraged to) send a “ping” message:

```
{
  "id": 123,
  "type": "ping"
}
```

The server replies with a pong message:

```
{
  "ok": true,
  "reply_to": 123,
  "type": "pong"
}
```

whoami

Primarily for debugging purposes, you can send the server a “who am I?” message to check whether you have authenticated correctly to the WebSocket.

```
{
  "id": 123,
  "type": "whoami"
}
```

The server will respond with a `whoyouare` message:

```
{
  "ok": true,
  "reply_to": 123,
  "type": "whoyouare",
  "userid": "acct:joe.bloggs@hypothes.is"
}
```

Developing Hypothesis

The following sections document how to setup a development environment for h and how to contribute code or documentation to the project.

Contributor License Agreement

Before submitting significant contributions, we ask that you sign one of our Contributor License Agreements. This practice ensures that the rights of contributors to their contributions are preserved and protects the ongoing availability of the project and a commitment to make it available for anyone to use with as few restrictions as possible.

If contributing as an individual please sign the CLA for individuals:

- [CLA for individuals, HTML](#)
- [CLA for individuals, PDF](#)

If making contributions on behalf of an employer, please sign the CLA for employees:

- [CLA for employers, HTML](#)
- [CLA for employers, PDF](#)

A completed form can either be sent by electronic mail to license@hypothes.is or via conventional mail at the address below. If you have any questions, please contact us.

```
Hypothes.is Project
2261 Market St #632
SF, CA 94114
```

Website dev install

The code for the <https://hypothes.is/> website and API lives in a [Git repo](#) named h. To get this code running in a local development environment the first thing you need to do is install h's system dependencies.

See also:

This page documents how to setup a development install of h. For installing the Hypothesis client for development see <https://github.com/hypothesis/client/>, and for the browser extension see <https://github.com/hypothesis/browser-extension>.

Follow either the *Installing the system dependencies on Ubuntu 14.04* or the *Installing the system dependencies on OS X* section below, depending on which operating system you're using, then move on to *Getting the h source code from GitHub* and the sections that follow it.

Installing the system dependencies on Ubuntu 14.04

This section describes how to install h's system dependencies on Ubuntu 14.04. These steps will also probably work with few or no changes on other versions of Ubuntu, Debian, or other Debian-based GNU/Linux distributions.

Install the following packages:

```
sudo apt-get install -y --no-install-recommends \
    build-essential \
    git \
    libevent-dev \
    libffi-dev \
```

```
libfontconfig \  
libpq-dev \  
libssl-dev \  
python-dev \  
python-pip \  
python-virtualenv
```

Install node by following the [instructions on nodejs.org](#) (the version of the nodejs package in the standard Ubuntu repositories is too old).

Upgrade pip, virtualenv and npm:

```
sudo pip install -U pip virtualenv  
sudo npm install -g npm
```

Installing the system dependencies on OS X

This section describes how to install h's system dependencies on Mac OS X.

The instructions that follow assume you have previously installed [Homebrew](#).

Install the following packages:

```
brew install \  
  libevent \  
  libffi \  
  node \  
  postgresql \  
  python
```

Note: Unfortunately you need to install the `postgresql` package, because Homebrew does not currently provide a standalone `libpq` package.

Getting the h source code from GitHub

Use `git` to download the h source code:

```
git clone https://github.com/hypothesis/h.git
```

This will download the code into an `h` directory in your current working directory.

Change into the `h` directory from the remainder of the installation process:

```
cd h
```

Installing the services

h requires the following external services:

- PostgreSQL 9.4+
- Elasticsearch v1.0+, with the [Elasticsearch ICU Analysis plugin](#)
- RabbitMQ v3.5+

You can install these services however you want, but the easiest way is by using Docker and Docker Compose. This should work on any operating system that Docker can be installed on:

1. Install Docker and Docker Compose by following the instructions on the [Docker website](#).
2. Run Docker Compose:

```
docker-compose up
```

You'll now have three Docker containers running the PostgreSQL, RabbitMQ, and Elasticsearch services. You should be able to see them by running `docker ps`. You should also be able to visit your Elasticsearch service by opening <http://localhost:9200/> in a browser, and connect to your PostgreSQL by running `psql postgresql://postgres@localhost/postgres` (if you have `psql` installed).

Note: If at any point you want to shut the containers down, you can simply interrupt the `docker-compose` command. If you want to run the containers in the background, you can run `docker-compose up -d`.

3. Create the `htest` database in the `postgres` container. This is needed to run the `h` tests:

```
docker-compose exec postgres psql -U postgres -c "CREATE DATABASE htest;"
```

Tip: You can use Docker Compose image to open a `psql` shell in your Dockerized database container without having to install `psql` on your host machine. Do:

```
docker-compose exec postgres psql -U postgres
```

Tip: Use the `docker-compose logs` command to see what's going on inside your Docker containers, for example:

```
docker-compose logs rabbitmq
```

For more on how to use Docker and Docker Compose see the [Docker website](#).

Installing the `gulp` command

Install `gulp-cli` to get the `gulp` command:

```
sudo npm install -g gulp-cli
```

Creating a Python virtual environment

Create a Python virtual environment to install and run the `h` Python code and Python dependencies in:

```
virtualenv .venv
```

Activating your virtual environment

Activate the virtual environment that you've created:

```
source .venv/bin/activate
```

Tip: You'll need to re-activate this virtualenv with the `source .venv/bin/activate` command each time you open a new terminal, before running `h`. See the [Virtual Environments](#) section in the Hitchhiker's guide to Python for an introduction to Python virtual environments.

Running h

Start a development server:

```
make dev
```

The first time you run `make dev` it might take a while to start because it'll need to install the application dependencies and build the client assets.

This will start the server on port 5000 (<http://localhost:5000>), reload the application whenever changes are made to the source code, and restart it should it crash for some reason.

Running h's tests

There are test suites for both the frontend and backend code. To run the complete set of tests, run:

```
make test
```

To run the frontend test suite only, run the appropriate test task with `gulp`. For example:

```
gulp test
```

When working on the front-end code, you can run the Karma test runner in auto-watch mode which will re-run the tests whenever a change is made to the source code. To start the test runner in auto-watch mode, run:

```
gulp test-watch
```

To run only a subset of tests for front-end code, use the `--grep` argument or mocha's `.only()` modifier.

```
gulp test-watch --grep <pattern>
```

Debugging h

The `pyramid_debugtoolbar` package is loaded by default in the development environment. This will provide stack traces for exceptions and allow basic debugging. A more advanced profiler can also be accessed at the `/_debug_toolbar` path.

http://localhost:5000/_debug_toolbar/

Check out the [pyramid_debugtoolbar documentation](#) for information on how to use and configure it.

You can turn on SQL query logging by setting the `DEBUG_QUERY` environment variable (to any value). Set it to the special value `trace` to turn on result set logging as well.

Feature flags

Features flags allow admins to enable or disable features for certain groups of users. You can enable or disable them from the Administration Dashboard.

To access the Administration Dashboard, you will need to first create a user account in your local instance of H and then give that account admin access rights using H's command-line tools.

See the [Accessing the admin interface](#) documentation for information on how to give the initial user admin rights and access the Administration Dashboard.

Troubleshooting

Cannot connect to the Docker daemon

If you get an error that looks like this when trying to run `docker` commands:

```
Cannot connect to the Docker daemon. Is the docker daemon running on this host?  
Error: failed to start containers: postgres
```

it could be because you don't have permission to access the Unix socket that the docker daemon is bound to. On some operating systems (e.g. Linux) you need to either:

- Take additional steps during Docker installation to give your Unix user access to the Docker daemon's port (consult the installation instructions for your operating system on the [Docker website](#)), or
- Prefix all `docker` commands with `sudo`.

Accessing the admin interface

To access the admin interface, a user must be logged in and have admin permissions. To grant admin permissions to a user, run the following command:

```
hypothesis user admin <username>
```

For example, to make the user 'joe' an admin in the development environment:

```
hypothesis --dev user admin joe
```

When this user signs in they can now access the administration panel at `/admin`. The administration panel has options for managing users and optional features.

An introduction to the h codebase

If you're new to the team, or to the Hypothesis project, you probably want to get up to speed as quickly as possible so you can make meaningful improvements to h. This document is intended to serve as a brief "orientation guide" to help you find your way around the codebase.

This document is a living guide, and is at risk of becoming outdated as we continually improve the software. If you spot things that are out of date, please submit a pull request to update this document.

This guide was last updated on 11 Apr 2017.

A lightning guide to Pyramid

The h codebase is principally a [Pyramid](#) web application. Pyramid is more of a library of utilities than a “framework” in the sense of Django or Rails. As such, the structure (or lack thereof) in our application is provided by our own conventions, and not the framework itself.

Important things to know about Pyramid that may differ from other web application frameworks you’ve used:

- Application setup is handled explicitly by a distinct configuration step at boot. You’ll note `includeme` functions in some modules – these are part of that configuration system.
- The `request` object is passed into views explicitly rather than through a `threadlocal` (AKA “global variable”), and is often passed around explicitly to provide request context to other parts of the application. This has a number of advantages but can get a bit messy if not managed appropriately.

You can read more about the distinguishing features of Pyramid in the [excellent Pyramid documentation](#).

Application components

The important parts of the h application can be broken down into:

Models [SQLAlchemy](#) models representing the data objects that live in our database. These live in `h.models`.

Views (and templates) Views are code that is called in response to a particular request. Templates can be used to render the output of a particular view, typically as HTML.

With a few exceptions, views live in `h.views`, and templates live in the `h/templates/` directory.

Services Putting business logic in views can quickly lead to views that are difficult to test. Putting business logic in models can lead to model objects with a large number of responsibilities.

As such, we put most business logic into so-called “services.” These are objects with behaviour and (optionally) state, which can be retrieved from the `request` object.

Services live in `h.services`.

Tasks Tasks are bits of code that run in background workers and which can be easily triggered from within the context of a request.

We use [Celery](#) for background tasks, and task definitions can be found in `h.tasks`.

There are a number of other modules and packages in the h repository. Some of these (e.g. `h.auth`, `h.settings`) do one-off setup for a booting application. Others may be business logic that dates from before we introduced the [services pattern](#), and thus might be more appropriately moved into a service in the future.

Submitting a Pull Request

To submit code or documentation to h you should submit a pull request.

For trivial changes, such as documentation changes or minor errors, PRs may be submitted directly to master. This also applies to changes made through the GitHub editing interface. Authors do not need to sign the CLA for these, or follow fork or branch naming guidelines.

For any non-trivial changes, please create a branch for review. Fork the main repository and create a local branch. Later, when the branch is ready for review, push it to a fork and submit a pull request.

Discussion and review in the pull request is normal and expected. By using a separate branch, it is possible to push new commits to the pull request branch without mixing new commits from other features or mainline development.

Some things to remember when submitting or reviewing a pull request:

- Your pull request should contain one logically separate piece of work, and not any unrelated changes.
- When writing commit messages, please bear the following in mind:
 - <http://tbagery.com/2008/04/19/a-note-about-git-commit-messages.html>
 - <https://github.com/blog/831-issues-2-0-the-next-generation>

Please minimize issue gardening by using the GitHub syntax for closing issues with commit messages.

- We recommend giving your branch a relatively short, descriptive, hyphen-delimited name. `fix-editor-lists` and `tabbed-sidebar` are good examples of this convention.
- Don't merge on feature branches. Feature branches should merge into upstream branches, but never contain merge commits in the other direction. Consider using `--rebase` when pulling if you must keep a long-running branch up to date. It's better to start a new branch and, if applicable, a new pull request when performing this action on branches you have published.
- Code should follow our *coding standards*.
- All pull requests should come with code comments. For Python code these should be in the form of Python *docstrings*. For AngularJS code please use *ngdoc*. Other documentation can be put into the `docs/` subdirectory, but is not required for acceptance.
- All pull requests should come with unit tests. For the time being, functional and integration tests should be considered optional if the project does not have any harness set up yet.

For how to run the tests, see *Running h's tests*.
- Your pull request should add a line to the *changelog* briefly describing the change and giving its GitHub pull request number.

Code style

This section contains some code style guidelines for the different programming languages used in the project.

Python

Follow [PEP 8](#), the linting tools below can find PEP 8 problems for you automatically.

Docstrings

All public modules, functions, classes, and methods should normally have docstrings. See [PEP 257](#) for general advice on how to write docstrings (although we don't write module docstrings that describe every object exported by the module).

The `pep257` tool (which is run by `prospector`, see below) can point out PEP 257 violations for you.

It's good to use Sphinx references in docstrings because they can be syntax highlighted and hyperlinked when the docstrings are extracted by Sphinx into HTML documentation, and because Sphinx can print warnings for references that are no longer correct:

- Use [Sphinx Python cross-references](#) to reference other Python modules, functions etc. from docstrings (there are also Sphinx domains for referencing objects from other programming languages, such as [JavaScript](#)).
- Use [Sphinx info field lists](#) to document parameters, return values and exceptions that might be raised.
- You can also use [reStructuredText](#) to add markup (bold, code samples, lists, etc) to docstrings.

Linting

We recommend running [Flake8](#) and [Prospector](#) over your code to find bugs and style problems, using the configurations provided in this git repo. With our configurations Flake8 is faster and less noisy so is nicer to run more frequently, Prospector is more thorough so it can be run less frequently and may find some problems that Flake8 missed.

Automated code formatting

You can use [YAPF](#) (along with the YAPF configuration in this git repo) to automatically reformat Python code. We don't strictly adhere to YAPF-generated formatting but it can be a useful convenience.

Additional reading

- Although we don't strictly follow all of it, the [Google Python Style Guide](#) contains a lot of good advice.

Front-end Development

See the [Hypothesis Front-end Toolkit](#) repository for documentation on code style and tooling for JavaScript, CSS and HTML.

Testing

This section covers writing tests for the h codebase.

Getting started

Sean Hammond has written up a [guide to getting started](#) running and writing our tests, which covers some of the tools we use (`tox` and `pytest`) and some of the testing techniques they provide (factories and parametrization).

Unit and functional tests

We keep our functional tests separate from our unit tests, in the `tests/functional` directory. Because these are slow to run, we will usually write one or two functional tests to check a new feature works in the common case, and unit tests for all the other cases.

Using mock objects

The `mock` library lets us construct fake versions of our objects to help with testing. While this can make it easier to write fast, isolated tests, it also makes it easier to write tests that don't reflect reality.

In an ideal world, we would always be able to use real objects instead of stubs or mocks, but sometimes this can result in:

- complicated test setup code
- slow tests
- coupling of test assertions to non-interface implementation details

For new code, it's usually a good idea to design the code so that it's easy to test with “real” objects, rather than stubs or mocks. It can help to make extensive use of `value objects` in tested interfaces (using `collections.namedtuple` from the standard library, for example) and apply the `functional core, imperative shell` pattern.

For older code which doesn't make testing so easy, or for code that is part of the “imperative shell” (see link in previous paragraph) it can sometimes be hard to test what you need without resorting to stubs or mock objects, and that's fine.

Writing documentation

To build the documentation issue the `make dirhtml` command from the `docs` directory:

```
cd docs
make dirhtml
```

When the build finishes you can view the documentation by running a static web server in the newly generated `_build/dirhtml` directory. For example:

```
cd _build/dirhtml; python -m SimpleHTTPServer; cd -
```

Serving h over SSL in development

If you want to annotate a site that's served over HTTPS then you'll need to serve h over HTTPS as well, since the browser will refuse to load external scripts (eg. H's bookmarklet) via HTTP on a page served via HTTPS.

To serve your local dev instance of h over HTTPS:

1. Generate a private key and certificate signing request:

```
openssl req -newkey rsa:1024 -nodes -keyout .tlskey.pem -out .tlscsr.pem
```

2. Generate a self-signed certificate:

```
openssl x509 -req -in .tlscsr.pem -signkey .tlskey.pem -out .tlscert.pem
```

3. Run `hypothesis devserver` with the `--https` option:

```
hypothesis devserver --https
```

4. Since the certificate is self-signed, you will need to instruct your browser to trust it explicitly by visiting `https://localhost:5000` and selecting the option to bypass the validation error.

Troubleshooting

Insecure Response errors in the console

The sidebar fails to load and you see `net::ERR_INSECURE_RESPONSE` errors in the console. You need to open `https://localhost:5000` and tell the browser to allow access to the site even though the certificate isn't known.

Server not found, the connection was reset

When you're serving h over SSL in development making non-SSL requests to h won't work.

If you get an error like **Server not found** or **The connection was reset** in your browser (it varies from browser to browser), possibly accompanied by a gunicorn crash with `AttributeError: 'NoneType' object has no attribute 'uri'`, make sure that you're loading <https://localhost:5000> in your browser, not `http://`.

WebSocket closed abnormally, code: 1006

If you see the error message **Error: WebSocket closed abnormally, code: 1006** in your browser, possibly accompanied by another error message like **Firefox can't establish a connection to the server at wss://localhost:5001/ws**, this can be because you need to add a security exception to allow your browser to connect to the websocket. Visit <https://localhost:5001> in a browser tab and add a security exception then try again.

403 response when connecting to WebSocket

If your browser is getting a 403 response when trying to connect to the WebSocket along with error messages like these:

- WebSocket connection to 'wss://localhost:5001/ws' failed: Error during WebSocket handshake: Unexpected response code: 403
- Check that your H service is configured to allow WebSocket connections from <https://127.0.0.1:5000>
- WebSocket closed abnormally, code: 1006
- WebSocket closed abnormally, code: 1001
- Firefox can't establish a connection to the server at wss://localhost:5001/ws

make sure that you're opening <https://localhost:5000> in your browser and *not* <https://127.0.0.1:5000>.

Making changes to model code

Guidelines for writing model code

No length limits on database columns

Don't put any length limits on your database columns (for example `sqlalchemy.Column(sqlalchemy.Unicode(30), ...)`). These can cause painful database migrations.

Always use `sqlalchemy.UnicodeText()` with no length limit as the type for text columns in the database (you can also use `sqlalchemy.Text()` if you're sure the column will never receive non-ASCII characters).

When necessary validate the lengths of strings in Python code instead. This can be done using [SQLAlchemy validators](#) in model code.

View callables for HTML forms should also use Colander schemas to validate user input, in addition to any validation done in the model code, because Colander supports returning per-field errors to the user.

Creating a database migration script

If you've made any changes to the database schema (for example: added or removed a SQLAlchemy ORM class, or added, removed or modified a `sqlalchemy.Column` on an ORM class) then you need to create a database migration script that can be used to upgrade the production database from the previous to your new schema.

We use [Alembic](#) to create and run migration scripts. See the Alembic docs (and look at existing scripts in [h/migrations/versions](#)) for details, but the basic steps to create a new migration script for h are:

1. Create the revision script by running `alembic revision`, for example:

```
alembic -c conf/alembic.ini revision -m "add the foobar table"
```

This will create a new script in `h/migrations/versions/`.

2. Edit the generated script, fill in the `upgrade()` and `downgrade()` methods.

See <https://alembic.readthedocs.io/en/latest/ops.html#ops> for details.

Note: Not every migration should have a `downgrade()` method. For example if the upgrade removes a max length constraint on a text field, so that values longer than the previous max length can now be entered, then a downgrade that adds the constraint back may not work with data created using the updated schema.

3. Stamp your database.

Before running any upgrades or downgrades you need to stamp the database with its current revision, so Alembic knows which migration scripts to run:

```
alembic -c conf/alembic.ini stamp <revision_id>
```

`<revision_id>` should be the revision corresponding to the version of the code that was present when the current database was created. The will usually be the `down_revision` from the migration script that you've just generated.

4. Test your `upgrade()` function by upgrading your database to the most recent revision. This will run all migration scripts newer than the revision that your db is currently stamped with, which usually means just your new revision script:

```
alembic -c conf/alembic.ini upgrade head
```

After running this command inspect your database's schema to check that it's as expected, and run `h` to check that everything is working.

Note: You should make sure that there's some representative data in the relevant columns of the database before testing upgrading and downgrading it. Some migration script crashes will only happen when there's data present.

5. Test your `downgrade()` function:

```
alembic -c conf/alembic.ini downgrade -1
```

After running this command inspect your database's schema to check that it's as expected. You can then upgrade it again:

```
alembic -c conf/alembic.ini upgrade +1
```

Batch deletes and updates in migration scripts

It's important that migration scripts don't lock database tables for too long, so that when the script is run on the production database concurrent database transactions from web requests aren't held up.

An SQL `DELETE` command acquires a `FOR UPDATE` row-level lock on the rows that it selects to delete. An `UPDATE` acquires a `FOR UPDATE` lock on the selected rows *if the update modifies any columns that have a unique index on*

them that can be used in a foreign key. While held this `FOR UPDATE` lock prevents any concurrent transactions from modifying or deleting the selected rows.

So if your migration script is going to `DELETE` or `UPDATE` a large number of rows at once and committing that transaction is going to take a long time (longer than 100ms) then you should instead do multiple `DELETES` or `UPDATES` of smaller numbers of rows, committing each as a separate transaction. This will allow concurrent transactions to be sequenced in-between your migration script's transactions.

For example, here's some Python code that deletes all the rows that match a query in batches of 25:

```
query = <some sqlalchemy query>
query = query.limit(25)
while True:
    if query.count() == 0:
        break
    for row in query:
        session.delete(row)
    session.commit()
```

Separate data and schema migrations

It's easier for deployment if you do *data migrations* (code that creates, updates or deletes rows) and *schema migrations* (code that modifies the database *schema*, for example adding a new column to a table) in separate migration scripts instead of combining them into one script. If you have a single migration that needs to modify some data and then make a schema change, implement it as two consecutive migration scripts instead.

Don't import model classes into migration scripts

Don't import model classes, for example `from h.models import Annotation`, in migration scripts. Instead copy and paste the `Annotation` class into your migration script.

This is because the script needs the schema of the `Annotation` class as it was at a particular point in time, which may be different from the schema in `h.models.Annotation` when the script is run in the future.

The script's copy of the class usually only needs to contain the definitions of the primary key column(s) and any other columns that the script uses, and only needs the name and type attributes of these columns. Other attributes of the columns, columns that the script doesn't use, and methods can usually be left out of the script's copy of the model class.

Troubleshooting migration scripts

(sqlite3.OperationalError) near "ALTER"

SQLite doesn't support `ALTER TABLE`. To get around this, use [Alembic's batch mode](#).

Cannot add a NOT NULL column with default value NULL

If you're adding a column to the model with `nullable=False` then when the database is upgraded it needs to insert values into this column for each of the already existing rows in the table, and it can't just insert `NULL` as it normally would. So you need to tell the database what default value to insert here.

`default=` isn't enough (that's only used when the application is creating data, not when migration scripts are running), you need to add a `server_default=` argument to your `add_column()` call.

See the existing migration scripts for examples.

Environment Variables

This section documents the environment variables supported by h.

CLIENT_URL

The URL at which the Hypothesis client code is hosted. This is the URL to the client endpoint script, by default <https://cdn.hypothes.is/hypothesis>.

Changelog

0.38.0 (2016-08-08)

Features

- Add “proxy” auth policy mode (#3638).
- Implement new form designs for logged out forms under a feature flag (#3640).

Miscellanea

- Add *hypothesis user add* command (#3634) and move admin command under the users subcommand as well (#3632).
- Replace “firstRun” config option with “openSidebar” and “openLoginForm” (#3643).
- Change URL paths for reset password, forgot password and register (#3651).
- Update Hypothesis client to v0.38.0. See <https://github.com/hypothesis/client/releases/tag/v0.38.0>

0.36.0 (2016-07-27)

Miscellanea

- Refactor *h.api.search* interface, it is now a *Search* class which can be customized on a per-query basis, the return value also changed to be a new *SearchResult* (#3622, #3623).
- Update Hypothesis client to v0.36.0. See <https://github.com/hypothesis/client/releases/tag/v0.36.0>

0.35.0 (2016-07-22)

Features

- Add ability to rename users from admin control panel (#3584)
- Improve the copy on reply notification emails (#3602)

Bug fixes

- Fix case sensitivity in user search queries (#3611)

Miscellanea

- Move the URLs for account settings from /profile to /accounts (#3604)
- The Hypothesis browser extension has now been moved to its own repository (#3620)
- Update Hypothesis client to v0.35.0. See <https://github.com/hypothesis/client/releases/tag/v0.35.0>

0.33.0 (2016-07-13)

Bug fixes

- Add a missing unique constraint on the *user_group* join table, and a migration to fix existing databases (#3591).

Miscellanea

- Update Hypothesis client to v0.33.0. See <https://github.com/hypothesis/client/releases/tag/v0.33.0>

0.32.0 (2016-07-08)

Features

- Treat HTTP/HTTPS URLs that are the same except for their protocol as equivalent (#3558).
- Improve performance of real-time updates service (#3574).
- Use consistent terminology for ‘Log in’, ‘Log out’ and ‘Sign up’ actions (#3578).

Bug fixes

- Fix missing escaping of user input in server-rendered forms (#3585).
- Prevent browsers disclosing group URLs to websites linked to from groups pages (#3579).
- Fix a problem where data migrations would update annotations’ last-update timestamp (#3586).

Miscellanea

- Remove assets that relate to the Hypothesis client. These are now part of the hypothesis/client repository (#3577, #3571)
- Update Hypothesis client to v0.32.0. See <https://github.com/hypothesis/client/releases/tag/v0.32.0>

0.31.0 (2016-06-29)

Features

- Show annotations and page notes in separate tabs in the sidebar (feature flagged: `selection_tabs`) (#3504)
- The ‘Share’ action for replies is now a direct link to the parent annotation and its associated conversation in context (#3539).

- Reply notification emails now contain direct links to see the original annotation and its associated conversation in context (#3536).

Bug fixes

- Fix adder not responding to selection changes after making a selection which does not contain any text (#3522, #3526).
- Fix incorrect ‘You do not have permission to see this annotation’ message in the client when direct-linking to an annotation in a PDF (#3529).
- Fix annotation cards containing long URLs overflowing their container (#3546).

0.30.0 (2016-06-23)

Features

- Users now get their own clean welcome page when they finish installing the browser extension (#3505).
- The search API can now accept multiple *uri* parameters and will return annotations made on any of the passed URIs (#3517).
- The annotation client now uses the multiple-*uri*-parameter feature to ensure that all annotations are loaded on PDFs (#3524).

Bug fixes

- Fix an issue where highlights weren’t shown correctly in PDFs (#3531).

Miscellanea

- Our brief experiment with using PDF fingerprint URNs as target *source* properties has ended (#3524).

0.29.0 (2016-06-20)

Features

- Add ‘#annotated’ hashtag to default tweet message to make following annotation activity on social media easier (#3482).
- Unify the ‘About this Version’, ‘Feedback’ and ‘Help’ menu items in the client (#3485).
- Automatically expand the conversation when direct-linking to annotations (#3489).

Bug fixes

- Fix group names wrapping onto multiple lines in sidebar (#3469).
- Fix the permalink pages for annotation replies. These now show the complete conversation thread which the reply appeared in, highlighting the reply matching the URL (#3474, #3483).
- Fix annotations without quotes resulting in entire pages or large sections of pages being highlighted (#3475).

- Fix annotations made on PDFs in browsers other than Safari not appearing when the same PDF was viewed in Safari and vice-versa (#3494).

Miscellanea

- Record message processing times in realtime update service (#3484, #3486).

0.28.1 (2016-06-14)

Bug fixes

- Revert code that loads annotations from database in */api/search* which caused the request times for certain URIs to skyrocket (9af5b65).

0.28.0 (2016-06-13)

Features

- Add option to disable the badge on the extension's toolbar button (#3342).
- Avoid revealing information about the file structure of the user's system to the Hypothesis service by not saving "file://" URLs with annotations made on PDFs (#3441).

Bug fixes

- Fix cursor style for annotation cards (#3463).
- Fix styling of quotes in annotation bodies (#3464).
- Fix annotation adder not being shown if the page already has a selection when Hypothesis is activated (#3453).

0.27.0 (2016-06-09)

Features

- It is now possible to authenticate to the WebSocket using an API token (#3419).
- Improve anchoring of annotations on PDFs by using the PDF fingerprint, when available, instead of the URL (#3404).

Miscellanea

- Load annotations from database in streamer server (#3378)
- Remove the "links" field from the search index (#3413).
- Add CLI command for moving existing annotations to a new URL (#3416).

0.26.0 (2016-06-07)

Features

- Show adder in response to selection changes made via touch or keyboard input (#3347)
- Change timestamps in annotation cards to link to the standalone annotation page rather than the original document (#3395)
- Improve presentation of reply threads (#3376)

Bug fixes

- Fix client on web pages which define a global variable called “global” (#3385)
- Support relative URLs for the “serviceUrl” parameter in the client specifying the Hypothesis service to connect to (#3381)
- Fix spacing around ‘Share’ icon in annotation cards (#3399)
- Fix display of numbered and bullet-point lists in annotation cards (#3396)
- URL encode Twitter/Facebook/Google share links correctly (#3394)

Miscellanea

- Documentation pages now have URLs without a ‘.html’ extension (#3373)
- Remove “direct_linking” feature flag (#3382)

0.25.0 (2016-05-26)

Miscellanea

- Remove code that writes to the legacy storage system (#3357)
- Visual improvements to the adder (#3346, #3370)
- Alleviate the issue where not all replies are loaded by increasing the limit to 200 per request (#3374)

0.24.0 (2016-05-25)

Features

- Improved performance and reduced memory usage of the sidebar when using Hypothesis on heavily annotated pages (#3287, #3360).

0.23.1 (2016-05-24)

Bug fixes

- We have had to temporarily revert the change which allows authenticating to the WebSocket with API tokens, as this had unintended side-effects preventing cookie authentication to same (3ed6ff3).

- Fix a problem with updating page notes when the ‘postgres’ feature flag was switched on (#3355).

0.23.0 (2016-05-23)

Features

- It is now possible to authenticate to the WebSocket using an API token (#3345).

Bug fixes

- The search *limit* parameter is now clamped within a range of acceptable values (#3341).

Miscellanea

- The setting for the WebSocket URL has changed from *H_WEBSOCKET_URL* to *WEBSOCKET_URL* (#3329).
- Added a code of conduct to the project repository (e3fad12).
- Numerous important changes to support the upcoming migration of annotation data to PostgreSQL (#3323, #3349, #3350, 4c25cd8).

0.22.0 (2016-05-12)

Features

- New visual design for annotation cards (#3274).
- Annotation cards now display more precise timestamps for annotations which were last changed more than 24 hours ago. (#3247)

Miscellanea

- Added *hypothesis devserver* command which runs all the services needed for a development instance of Hypothesis and multiplexes their output into the terminal (#3307).
- Added *hypothesis migrate* command to simplify the execution of database migrations (#3319).
- Removed several unused *hypothesis* subcommands (#3303).
- Added *-dev* flag to *hypothesis* command which replaces the need to specify a config file path as an argument to subcommands (#3303).

0.21.0 (2016-05-06)

Features

- Show adder when Hypothesis loads if there is already a text selection on the page (#3257).
- Make annotation adder position itself more intelligently in relation to the selected text and viewport boundaries (#3257).
- Add “indexer” worker queue to handle synchronisation of the search index with the database under the ‘postgres’ feature flag (#3242).

Bug fixes

- Fix a syntax error thrown when injecting the Hypothesis sidebar (#3289).

Miscellanea

- Link to the Chrome extension on the help page (#3265).
- Major work on storage as we continue to work towards a migration of all annotation data to PostgreSQL (#3262, #3250, #3242).
- Preliminary builds of a Firefox WebExtensions add-on (#3263).

0.20.0 (2016-04-25)

Features

- Remove any empty annotations with no tags or text when creating new annotations (#3214).
- Add a ‘Copy to Clipboard’ button in the annotation share dialog (#3197).
- Add ‘About this version’ panel accessible via Help -> About this version (#3216).
- When activating Hypothesis on a page whose canonical URL has changed since it was first annotated, surface all annotations for that page, not just those made since the page had its current canonical URL (#3241). This fixes an issue where annotations disappeared from WordPress blogs.

Bug fixes

- Fix sidebar not scrolling to show Share or Login dialogs when activating them via links in the top bar (#3215).
- Do not treat characters inside math expressions as possible markdown formatting commands (#3208).
- Fix failure to anchor annotations made at the start of PDF pages (#3234).
- Fixed KaTeX font URLs failing to load when rendering math (#3209).
- Handle request failures more gracefully for various network requests made by the sidebar (#3210, #3211).

Miscellanea

- Switched real-time message processing from NSQ to RabbitMQ (#3217).

0.19.1 (2016-04-14)

Bug fixes

- Fix a bug that prevented the “Post” button from correctly updating its enabled/disabled state when typing an annotation body (#3213).

0.19.0 (2016-04-14)

Features

- Background tasks are now processed by Celery workers rather than our own homebrew worker implementation (#3189).
- Improved display for local files in the “recently annotated documents” list on group pages (#3200).

Bug fixes

- Fix the display of documents with multiple possible titles in group pages (#3200).

Miscellanea

- More changes to support the migration of annotation data to PostgreSQL (#3203, #3206).

0.18.0 (2016-04-13)

Features

- Annotations are now available in (beta-quality) Web Annotation compatible JSON-LD format via the API (#3181).

Bug fixes

- Don't report expected/unavoidable errors when injecting the sidebar to Sentry (#3186).
- Fix broken links in the admin panel (#3187).
- Fix an infinite loop occasionally triggered by the truncation of long annotations (#3188).

Miscellanea

- Numerous major changes to annotation storage as part of the work to migrate annotation data to PostgreSQL (#3153, #3184, #3190, #3199).
- Add a skeleton functional test suite for the web application (#3198).

0.17.0 (2016-04-06)

Bug fixes

- Fix a regression in sorting annotations when the sort order is set to ‘Location’ (#3179).
- Clicking on an annotation quote now scrolls the page to that annotation when the quote is collapsed (#3180).

Miscellanea

- Add a live reload facility for development of the Hypothesis front-end (#3038).

0.16.0 (2016-04-04)

Bug fixes

- Fix a bug that prevented “direct links” to annotations on PDFs from working correctly (#3139).

Miscellanea

- Annotation “incontext” links are now generated with the URL of the annotated page appended (#3172).

0.15.0 (2016-03-31)

Features

- Improved iconography for annotation cards (#3116).
- Add “call to action” copy explaining to logged-out users what they’re looking at when viewing a direct-linked annotation (#3124).
- Annotation selections in the sidebar are now preserved across login/logout/account change – this applies to “direct links” in particular (#3133).
- Admin information pages for users now show group memberships (#3138).

Bug fixes

- Fixed a typo in the Chrome extension content security policy that caused extension console warnings (#3140).
- Fix a bug that caused worker processes to crash when encountering unexpected exceptions while handling NSQ messages (#3156).

Miscellanea

- Sentry error reports from the client code are now configured using a distinct DSN (#3132).
- Elasticsearch client timeouts and gunicorn worker timeouts are now configurable using environment variables (1a6ae98, e0ef808).

0.14.0 (2016-03-23)

Miscellanea

- Remove usage of *h.accounts.User.status* bitfield (#3120).
- Show message when a direct-linked annotation is selected but the annotation is not available (#3093).
- Properly hook up Chrome production extension with production bouncer (#3077).

0.13.0 (2016-03-22)

Features

- The ‘adder’ that appears when selecting text has a new, clearer design, and the toolbar ‘note’ button allows the creation of annotations when a selection has been made (#3078).
- The client can now respond to a URL fragment of the form `#annotations:<id>` in order to focus and scroll to a specified annotation on page load (#3085).
- The ‘link’ button on annotation cards can now display links to annotations in context on the page where they were made (feature flagged: `direct_linking`) (#3105).

Bug fixes

- Fixed an issue where the client displayed an inconsistent login state when switching accounts with Hypothesis active in multiple tabs (#2924).
- Fixed an issue where private group annotations were not loaded correctly after switching user accounts (#3083).
- The Chrome extension now correctly indicates when it was not injected into a page because another Hypothesis client is already present (#3097).
- Fix cursor position after using editor toolbar buttons to create a new list or quote, if the selection was previously empty and the cursor was positioned at the start of the line (#3114).

Miscellanea

- The undocumented `window.hypothesisInstall()` function has been removed (#3098).
- Substantial improvements to the feature flagging system, reducing the amount of network chatter between the application and the database (#3110, #3115).

0.12.0 (2016-03-14)

Bug fixes

- Fixed error when rendering invalid LaTeX markup. As a side effect, the fallback to MathJax rendering for markup that is not supported by KaTeX has been removed for the time being (#3042).
- Render annotation’s text property as empty string instead of `null` (#3072).
- Split out DocumentMeta type normalisation to only use it in the Postgres migration (#3090).

Miscellanea

- Render annotation links in API response (#3081) including an `incontext` link behind the direct-linking feature flag (#3087).
- Properly hook up Chrome staging extension with staging bouncer (#3077).
- Add a utility script to aid the transition away from CoffeeScript (#3075).
- Remove `h.claim` package (#3089).
- Clean up old and unused feature flags (#3088).

0.11.0 (2016-03-08)

Bug fixes

- Replies are now correctly displayed after creation (#3057).
- Fix a bug that caused annotations to be rendered incorrectly, both in the API (where they had a *permission* field instead of a *permissions* field) and in the UI (where they had edit controls they shouldn't have had) (#3059).
- Fix a bug that caused page notes to be hidden on all pages (#3063).
- Fix compatibility with IE 10, 11.0 and early versions of Microsoft Edge (#3064).
- Restore two-step confirmation for user deletion in admin panel (#3066).

Miscellanea

- Early support for serving Content-Security-Policy headers from the application (#3024).

0.10.0 (2016-03-07)

Features

- The annotation text input is focused automatically when new annotations are created (#3041).

Miscellanea

- Introduce a presentation layer for the API (#3011, #3047, #3050, #3054).
- Handle deleted annotations in Postgres migration script (#3040).
- Updated the Ubuntu development install instructions (#3046).
- Upload sourcemaps to Sentry (#3055).

0.9.7 (2016-03-01)

Bug fixes

- Fix error in NSQ connection configuration (c25fa95).

0.9.6 (2016-03-01)

Bug fixes

- Fix WebSocket server initialization (#3039).

0.9.5 (2016-03-01)

Bug fixes

- Fix a bug that prevented the release from being built (a7e2189).

0.9.4 (2016-03-01)

Features

- Enable admins to activate users (#3015).
- Preserve selection when applying block formatting (67409c8).

Bug fixes

- Fix a link to our mailing list archive (#3019).
- Fix typo in NIPSA admin template (50abf12).
- Fix tags autocomplete dropdown CSS (#3009).

Miscellanea

- Pyramid upgraded to 1.6 (#3034).
- The undocumented API endpoint /api/annotations has been removed (#3036).

0.9.3 (2016-02-24)

Bug fixes

- Fix the definition of the bouncer URL match pattern in the Chrome extension manifest (6ce2dad).
- Ensure the streamer routes messages correctly when NSQ_NAMESPACE is set (#3008).

0.9.2 (2016-02-24)

Bug fixes

- Fix a blocker bug preventing database transactions from being committed correctly (7ef09c3).

0.9.1 (2016-02-24)

Bug fixes

- Major improvements to the behaviour of the websocket server (AKA the “streamer”) under high concurrency (#2996).

Miscellaneous

- This release contains an all-new build system for the frontend assets, which include the assets for the web service and the browser extension (#2958).
- A second large piece of the work to move annotation storage into PostgreSQL has been merged (#2986).

0.9.0 (2016-02-22)

Features

- Users can now generate long-lived API tokens from their profile page (#2948).
- Correctly detect when the Chrome extension is being installed by an administrative policy and suppress the “welcome page” tab (#2964).

Bug fixes

- Fix a crash that prevented signups if the username blacklist was missing (#2954).
- Groups lists are now scrollable – long lists of groups don’t flow off the end of the page and become unreachable (#2973).
- Fix a bug that prevented users from replying to annotations on stream search pages (#2978).
- The icons in the sidebar now load on GitHub and other sites that restrict loading of inline fonts using Content Security Policy. (#2266)

Miscellaneous

- Attempting to create an annotation in a group of which the logged-in user is not a member will now result in an explanatory error message rather than a 404 (#2981).
- Client upgraded to use Angular 1.5 (#2967).
- An important first tranche of work that will move annotation storage to PostgreSQL has been merged (#2955).
- Source maps are now generated for production builds of the client and front-end assets for easier debugging and better error reporting.
- The download size of the client application has been reduced by 44% from 516KB (minified and gzipped) to 290KB.

0.8.15 (2016-02-12)

Bug fixes

- Ensure a useful error is shown when users with unactivated accounts attempt to sign in (#2952).
- Fix an error that would prevent Hypothesis from loading in IE10/11 (#2953).

0.8.14 (2016-02-12)

Bug fixes

- Fix an issue where a Hypothesis client that couldn’t connect to the web service would busy-loop (#2916).
- Fix a bug where the “sidebar tutorial” wouldn’t correctly dismiss (#2925).
- Ensure that requests to the badge API correctly send a cookie (#2929).
- Numerous small fixes to the Hypothesis client and Chrome extension (#2933, #2941, #2947).

Miscellanea

- The API component of the web application now only accepts API tokens for authentication (#2923).
- Improvements to the handling of unexpected errors in the Chrome extension (#2942).

0.8.13 (2016-02-05)

Features

- Improve messaging for users who click on an activation link more than once, or who visit an activation link when already logged in (#2904).
- Allow administrators to delete user accounts from the admin panel (#2907).

Bug fixes

- Fix a character encoding bug that prevented CSV export of groups from the admin panel (#2989).
- Display the correct number of annotations for all users in the admin panel (#2900).
- Suppress an unsightly flash of banner on the stream pages (#2908).
- Fix a bug that prevented editing of group annotations on stream pages (#2911).

Miscellanea

- Numerous changes to the build process for the Chrome extension. Command-line flags have changed (#2091, #2913).
- Make it possible to build a client/extension that doesn't use a websocket endpoint (#2906).
- Move the websocket service into a standalone Pyramid application. This allows the main web application to use a simpler worker type, thus preventing database connection pool exhaustion in high load environments (#2913).
- Move mail delivery to a worker process. This ensures that failures to deliver mail are a) retried, and b) do not trigger transaction rollbacks in web requests (#2903).

0.8.12 (2016-02-01)

Features

- Searching for “group:<groupid>” in the stream pages will now function correctly (#2882).
- Detect when the Chrome extension is being installed by an administrative policy and suppress the “welcome page” tab (#2891).

Bug fixes

- Don't display the “create an account” banner on the stream or standalone annotation pages (#2879).
- Fix a problem where canceling edits to an annotation would revert the annotation to its first known state rather than the state before the edit was started (#2884).
- Fix a problem where multiple clicks on the save button would create multiple near-identical annotations (#2887).

Miscellanea

- Add support for Sentry client-side error reporting in the Chrome extension and site code (#2850).
- Strip WebTrends “WT.*” query parameters as part of URI normalization (#2862).
- Removed the incomplete Firefox extension code which had fallen a long way behind Chrome, with the intention of replacing it with one based on WebExtensions in the near future (#2877).

0.8.11 (2016-01-14)

Bug fixes

- Fix a bug that prevented the Docker container from correctly starting up (614e6b9).

0.8.10 (2016-01-14)

Bug fixes

- Fix a bug that prevented Hypothesis from working on GitHub (#2847).
- Fix a couple of crashing bugs in the Chrome extension (#2849, #2851).
- Fix a bug that caused the wrong group name to display on annotations in the stream (#2854).
- Fix a bug where threads were being shown uncollapsed instead of collapsed initially, and the collapse/uncollapse buttons didn’t work (#2855).
- Strip the proxy prefixes from URLs we know to be proxied (through our “via” service) when normalising (#2861).

Miscellanea

- Add a clear “call to action” banner in the sidebar for signed-out users (#2843).
- Avoid truncating annotation cards that exceed the collapsed height by only a small amount (feature flagged: truncate_annotations) (#2859).

0.8.9 (2016-01-11)

Bug fixes

- Fix a bug where annotation deletions were not correctly processed by the “real-time” streamer (7daa709)

Features

- Add a tutorial for new users to the sidebar (feature flagged: sidebar_tutorial). (#2824)

0.8.8 (2016-01-11)

Bug fixes

- Fix a bug where embedded videos were not always shown (#2828, #2807).
- Fix numerous usability issues and a potential denial-of-service attack associated with password resets (#2803).
- Fix a bug which prevented Hypothesis from being correctly re-activated on a Chrome tab after it navigated (#2838).

Features

- Embedded videos can now be shown fullscreen (#2814, #2816).
- PDF documents can now be detected regardless of file extension (#2834).

Miscellanea

- We now clear any “selection” of annotations when a new annotation is created, so the editor is always visible (#2817).
- Various improvements to the appearance of truncated annotations (feature flagged: `truncate_annotations`) (#2802).

0.8.7 (2015-12-18)

Bug fixes

- Links containing underscores within annotations are no longer mangled (#2801).
- Fix a crash when invalid data is sent to the login endpoint (#2793).

Features

- New homepage design (feature flagged: `new_homepage`) (#2770).
- Change YouTube and Vimeo links into video embeds (feature flagged: `embed_media`) (#2805).

0.8.6 (2015-12-11)

Bug fixes

- Fix broken standalone annotation pages for replies (#2786).
- Fix a bug where realtime updates weren't delivered to standalone annotation pages for replies if the thread root on that page was also a reply (#2787).
- A number of other small fixes and clean-ups.

0.8.5 (2015-12-08)

Bug fixes

- Fix a blocker bug preventing annotations from being created (3d014fc).

0.8.4 (2015-12-08)

Bug fixes

- Fix a crash when rendering /robots.txt (e1c6021).

0.8.3 (2015-12-08)

Bug fixes

- Fix a crash that meant that reply notification emails would not be sent if the annotation replied to had no text (#2771).
- Fix a bug where realtime annotation updates weren't being sent to logged-out clients (#2776).

Features

- A report on groups is now available in the admin dashboard, which can be downloaded in CSV format (#2764, #2772).

Miscellanea

- Banner added to the homepage (#2758).
- The community guidelines are now linked from the signup form (#2760).
- The frontend code that controls the annotation card widget has been translated to JavaScript, and numerous aspects of it have been refactored or simplified (#2756, #2761, #2762, #2763).
- The site now serves a robots.txt file (#2778).

0.8.2 (2015-11-30)

Bug fixes

- Fix a problem where messages that were supposed to be sent over the WebSocket on reconnect weren't (#2746).
- Fix a bug where an old sort control was shown in the sidebar (#2751).
- Fix a bug where creating an annotation before login could result in invalid data being sent to the server (#2754).

Miscellanea

- The Hypothesis home page is now served by this application (#2740).
- Annotation data is now validated using JSON Schema (#2745).

0.8.1 (2015-11-26)

Bug fixes

- Fix a series of bugs relating to the incorrect handling of replies on the stream pages (#2737).

Features

- Enable highlights by default, everywhere (#2739).

0.8.0 (2015-11-24)

Bug fixes

- Fix a bug where unsaved drafts persisted on logout (#2708).
- Fix a bug where annotation permissions/post intent was not preserved when changing between groups (#2717).
- Don't attempt to update the timestamps of unsaved annotations far too frequently (#2725).
- Fix a bug where a websocket reconnection wouldn't retransmit its search filter state (#2719).
- Ensure that the permissions (shared/public) of new annotations is correctly preserved when switching between groups (#2713).

Features

- Annotations can now be made and shared in private groups (#2729).

Miscellanea

- First steps towards Python 3 compatibility (#2706).
- Old and unused feature flag data is now purged from the database on application startup (#2733).

0.7.13 (2015-11-03)

Bug fixes

- Fix a broken reference within our code (502e8df).

0.7.12 (2015-11-03)

Bug fixes

- Fix a couple of small display issues (#2704, #2710).
- Fix an issue where badge URLs were not correctly encoded before being sent to the server (#2709).

0.7.11 (2015-11-02)

Bug fixes

- Fix a problem where occasionally the set of public annotations would be loaded into the initial view rather than the set of annotations for the focused group (feature flagged: groups) (#2684).
- Fix a bug where creating an annotation when signed out (and subsequently signing in and saving it) could result in invalid permissions fields (#2687).
- Fix a problem in Safari where the search box didn't expand when you clicked on it (#2699).
- Fix improper case-sensitivity for tag searches (e.g. searches for "Tag123" now correctly return annotations tagged with "tag123") (#2690).

Features

- Ignore the "gclid" query parameter (a Google AdWords click-tracking param) when normalising URLs (72f0509).
- Draft annotations are now preserved when switching from group to group in the sidebar (#2689).

Miscellanea

- Improvements to Sentry logging (the current URL, headers, and userid are now recorded with exceptions) (#2697).
- Show the werkzeug debugger on exceptions in development (#2698).

0.7.10 (2015-10-28)

Bug fixes

- Fix a problem where an incorrect search query was sent to our server due to semicolons in the page URL (6513184).

0.7.9 (2015-10-28)

Bug fixes

- Fix a problem where activating the Chrome extension would obliterate a version of Hypothesis embedded on the page (#2657).
- Fix a visual issue causing the "Clear selection" and "Clear search" buttons to be briefly visible when they shouldn't have been (#2668).
- Fix a crash triggered when the set of connected WebSocket clients changed while handling a message (#2647).
- Fix a bug where cancelling leaving a group nonetheless resulted in group focus changing (#2669).

Features

- Improved appearance and behaviour of the sort control for annotations (feature flagged: groups) (#2643).
- Replies now inherit the publication scope of their parents. That is: replies to group annotations will go to the same group (#2650).
- Support HTTP conditional responses (ETag/If-None-Match and Last-Modified/If-Modified-Since) under appropriate conditions (#2664).
- Groups landing pages now show a list of recently annotated pages (feature flagged: groups) (#2667).

Miscellanea

- Upgrade to Angular 1.4.7 (#2629).
- Account settings and profile forms are now rendered by the server (#2636).
- The Chrome extension can now be built in a way that allows distinguishing between development versions of the extension and production ones (#2639).
- No longer perform the URI expansion step when searching for annotations on URLs which have been marked “canonical”. This hopefully reduces the number of false-positive annotations we load on pages with appropriate metadata (#2652).
- Replace group public IDs (hashids) with randomly generated IDs (#2662).

0.7.8 (2015-10-20)

Bug fixes

- Fix a problem where the realtime updates feature would silently stop processing messages on exceptions (#2617).
- Groups in the groups dropdown are always focused, even if their identifier starts with a number (feature flagged: groups) (#2627).

Features

- Improved appearance and behaviour of the controls to clear a selection or a search (#2615).
- Improved appearance and behaviour of the sidebar “top bar” (partially feature flagged: groups) (#2616).
- RSS feed contains authorship information (usernames) (#2621).

Miscellanea

- Search query filters revert to default AND behaviour (#2620).
- Joining a group from a signed out state is now easier (#2625).

0.7.7 (2015-10-14)

Bug fixes

- Clean up annotated document filenames before display (so “my%20doc.pdf” becomes “my doc.pdf”) (#2597).
- Annotators are now able to select the privacy of their annotations before they are able to save them (#2601).
- The component that fetches feature flag data for the frontend will no longer busy-poll the ajax endpoint if it receives an error (#2612).

Features

- Add the ability to leave a group (feature flagged: groups) (#2588).
- Notify the frontend in real time, using the websocket, when groups are joined/left (feature flagged: groups) (#2591).
- Truncate long annotation quotes and bodies (feature flagged: truncate_annotations) (#2451).

Miscellanea

- The Chrome extension is now built using browserify (#2609).
- Accounts forms (login/register/forgot password) are now rendered by the server (#2582).

0.7.6 (2015-10-08)

Bug fixes

- Fix a bug where the URL of the annotated page didn't appear on annotation cards in Safari/IE (#2574).
- Fix the ability to “select” one or more annotations in the sidebar by clicking on a highlight in the page (#2576).

Features

- Introduce a new, clearer “save” button for annotation editing (#2550).
- Add the ability to focus the current view on annotations from a specific group (feature flagged: groups) (#2566).
- Show the filenames of locally annotated files on annotation cards (#2570).
- Improve the appearance of the user flow when joining a group (#2577).

Miscellanea

- Deprecate the use of SQLite in development environments (#2579).

0.7.5 (2015-10-01)

Bug fixes

- Fix a bug where cancelling a change to an annotation did not reset changes to the annotation text (#2562).
- Fix the broken email notification system (#2558).
- Fix a crash caused by submitting an annotation with null document “link” fields (#2520).

Features

- Removed support for old-style Annotator front-end auth (11135fd).
- URLs in annotation text are now automatically converted to links (#2552).

0.7.4 (2015-09-25)

Bug fixes

- Support retrieving comments (“page notes”) through search when using new normalized URI search (feature flagged: groups) (#2549).
- Fix standalone annotation pages failing to display their annotation (92010c1).
- Fix unanchored annotation warnings displaying in the wrong places (feature flagged: show_unanchored_annotations) (#2542).
- Fix a bug where newly created annotations would sometimes disappear from the sidebar for a few moments (#2542).
- Fix display of page titles on some annotation cards (#2533).
- Fix a couple of crashes when annotations were created without expected fields (#2545, #2546).

Features

- Improved group selection menu (feature flagged: groups) (#2514).

0.7.3 (2015-09-22)

Bug fixes

- Fix annotation in IE10.

Features

- New and improved Dockerfile built on Alpine Linux, resulting in a substantially smaller built image (down to 250MB from ~750MB).
- Remove the need to patch the global window object (upgrade to dom-anchor-* v2.0.0).
- Added a stream RSS feed at */stream.rss*.

0.7.2 (2015-08-14)

Bug fixes

- Fixed a regression that prevented infinite scroll from working on the stream.

Features

- Improve scrolling performance by using a fluidly sized body and scrolling the whole document rather than a fixed body.
- Rewrite infinite scroll pagination to use regular HTTP requests instead of WebSocket.

0.7.1 (2015-08-13)

Bug fixes

- Clicking on annotation cards and the navigational bucket indicators should work once more in the PDF.js viewer.
- Fixed an issue with timezone localization that caused unnecessary errors to be thrown and caught. Auditing this resulted in a removal of some significant bloat from unnecessary code.

Features

- When sorting annotations by document location the TextPositionSelector information is now used instead of highlight position information. This causes less shuffling and re-rendering on load and when lazy-rendered pages in the PDF.js viewer appear and disappear at the cost of seeing annotations that have changed position or that target content within fixed position containers sometimes appear to be out of order.
- Removed an unnecessary call, originating in the infinite scrolling code, from the sidebar widget.

0.7.0 (2015-08-10)

Bug fixes

- Only update the stream websocket filter when there is at least one URI to search. See <https://github.com/hypothesis/h/pull/2419>
- Don't give admins permissions globally, but require instead that resources specifically grant privileges to admins. See <https://github.com/hypothesis/h/pull/2424>
- Ensure that API requests always have a valid token if the user is logged in. See <https://github.com/hypothesis/h/pull/2415>

Features

- Enable users to create and share groups. See <https://github.com/hypothesis/h/pull/2402> and <https://github.com/hypothesis/h/pull/2412>
- New, experimental URI normalization, accessible by turning on the 'search_normalized' feature. See <https://github.com/hypothesis/h/pull/2413>

- Add a staff user designation and support feature toggles for staff only. See <https://github.com/hypothesis/h/pull/2416>
- Support feature toggles for admins only. See <https://github.com/hypothesis/h/pull/2435>
- Improve the scrolling experience when clicking on bucket tabs and annotation cards. The view now scrolls so that the annotation is one fifth of the way down the screen, allowing room for navigation bars but leaving the annotation near the top of the screen.
- Support for the PDF.js viewer shipping in Firefox 40.
- Add a NIPSA service worker definition.

0.6.0 (2015-07-29)

Bug fixes

- Support for relative URLs return from document metadata plugins.
- Fix a possible infinite digest cycle in the features client.
- All not found responses now have a 404 status code.

Features

- Support for flagging users as “Not In Public Site Areas” or “NIPSA”. See <https://github.com/hypothesis/h/pull/2300>
- Support for admin users.
- Support for turning features on only for admins.
- A new administration page for admins.

Backwards Incompatibilities

- Support for the h.autologin feature has been removed.

Security

- Session cookies are now marked as HttpOnly to prevent session stealing by cross-site scripting attacks.

0.5.1 (2015-07-21)

Bug fixes

- Fix an error preventing the Atom feed from working.

0.5.0 (2015-07-21)

Features

- Share a link to a page with annotations using the Via proxy service. See <https://github.com/hypothesis/h/pull/2215>
- Make the privacy setting more obvious on new annotations. See <https://github.com/hypothesis/h/pull/2322>
- Use better security practices when making HTTPS requests. See <https://github.com/hypothesis/h/issues/2343>
- Make it possible for administrators to enable and disable features without redeploying. See <https://github.com/hypothesis/h/issues/2354>
- Preliminary support for admin users. See <https://github.com/hypothesis/h/pull/2358>

Bug fixes

- Improve performance, reliability, and responsiveness on complex or dynamic pages, avoiding non-responsive script errors and anchoring failures. See <https://github.com/hypothesis/h/pull/2362>
- Prevent annotating when not signed in to avoid confusing users with data loss. See <https://github.com/hypothesis/h/pull/2361>
- Make it possible to embed a guest frame once again, that participates in annotation with an existing sidebar. See <https://github.com/hypothesis/h/pull/2340>
- Fix formatting issues with the Atom feed. See <https://github.com/hypothesis/h/pull/2341> and <https://github.com/hypothesis/h/pull/2338>
- Fix an issue where badly formatted annotations could break the Atom feed. See <https://github.com/hypothesis/h/pull/2345>
- Speed up searches by avoiding an extra request on the backend. See <https://github.com/hypothesis/h/pull/2346>
- Speed up searches by avoiding extra requests on the frontend. See <https://github.com/hypothesis/h/pull/2348>
- Address several causes of stuck transactions that make migrations difficult and could, in some cases, make the server return errors for many requests. See <https://github.com/hypothesis/h/pull/2381>
- Fix an issue where failed document equivalence searches resulted in annotations on <http://example.com/> being returned. See <https://github.com/hypothesis/h/pull/2334>
- Avoid some problems caused by annotating the application itself, such as by annotating the stream page.
- User experience and usability improvements. See <https://github.com/hypothesis/h/pull/2330> <https://github.com/hypothesis/h/pull/2352> and <https://github.com/hypothesis/h/pull/2349>
- Fix the token command-line tool to generate proper tokens even when the server is running on a port other than the default. See <https://github.com/hypothesis/h/pull/2357>

Documentation

- Expand the search API documentation to better describe the available fields for filtering. See <https://github.com/hypothesis/h/pull/2344>

0.4.2 (2015-06-16)

- Silence SQLAlchemy warnings (#2258)
- Show errors when math parsing fails (#2241)
- Let users change their email address (#2131)
- Fix inappropriate WebSocket error reporting in logs (#2256)
- Support for Python 2.7.9
- Improve extension build documentation (#2265)
- Remove dependency on horus (#2274, #2281, #2284, #2291, #2313, #2312, #2317, #2318)
- Keep CSS for annotator component separate from the site (#2279)
- Prevent environment variables from interfering with tests (#2283)
- Clearly indicate support for using email addresses for login (#2288)
- Improve search code (#2282)
- Improve reporting of form errors (#2290)
- Support anonymous CORS in the API (#2303)
- Remove unnecessary toast messages when editing a user profile (#2310)
- Improve Docker build caching (#2311)
- Upgrade gnsq dependency
- Simplify database session handling (#2320)

0.4.1 (2015-05-21)

- Add NIPSA flag to user table (migration needed!)
- Upgrade to Annotator v1.2.x tip (6536160)
- Hide the widget panel until ready for input (#2207)
- Fix UI z-index to actual maximum (#1909)
- Change annotation card action from ‘share’ to ‘link’
- Add a client-side error when saving an annotation fails
- Snap the sidebar closed as well as open (#2162)
- Put NSQ usage behind feature flag. The API no longer requires NSQ.
- For development, disable WebSocket streaming, email notifications, and NSQ.
- Lots of linting.
- Added support for URL parameters to the Atom feed at `/stream.atom`. For example: `/stream.atom?user=seanh` or `/stream.atom?user=seanh&tags=foo,bar`.
- Users can now change their email addresses using the Account form (#2131)

0.4.0 (2015-05-05)

Highlights

- Add Markdown Editor (#1479)
- Add Math support for annotations (#1558)
- Simpler CSS grid system (#1577)
- Improved Chrome extensions handling of PDF.js viewer (#1563)
- Post-install Welcome page for user onboarding (#1579)
- Switched to Jinja2 for server-side templates (#1628)
- Initial Firefox Addon (#1434)
- Add `./bin/hypothesis reindex` command (#1715)
- Rework Back End Authentication and Authorization (#1791)
- Import Annotator (#1856)
- Depend on upstream Annotator (#1866)
- Enable Sentry logging in production environments (#1906)
- Open Graph protocol metadata added to Annotation view (#1921)
- Refactor auth and separate API from main app (#1951)
- Use key derivation to provide secret keys (#1981)
- Add claim account system (#1941)
- Browserify ALL THE THINGS (#1972)
- Add `./bin/hypothesis token` command to generate OAuth tokens (#2032)
- Refactor UX (#2031)
- Auto-complete tags (#2042)
- Add Atom feed support for `/stream` (#2072)
- Improve packaging, bundling and module boilerplate (#2092)
- Google Analytics support (#2139)
- Mobile support (#2137)
- Protect against double embedding/injecting (#2166)
- Add a blocklist of sites h doesn't work on (#2157)
- Overhaul URI analysis (#2184)

0.3.2 (2014-09-24)

Patch release to upgrade angular.js in light of security vulnerabilities. See <http://avlidienbrunn.se/angular.txt>

0.3.1 (2014-08-25)

Fixes

- The token API returns OAuth errors rather than choking (#1406)

Backwards Incompatibilities

- Support for clients before v0.2 has been dropped

0.3.0 (2014-08-23)

New Features

- Account deactivation and password change support (#632, #1275)
- Heatmap tabs no longer show reply count.
- HTML emails for reply notifications
- Update dom-text libraries to support PDF.js v1.0.277
- Better tokenization of URIs for search (#1308, #1407)
- Markdown previews (#1418)

Fixes

- Improved form validation (#1275, #1388, #1394)
- Source citation information on cards in the stream (#1390, #1423, #1425)
- Searching for a bare username works again (#1391)
- Add iconography for privacy settings
- Replace various SVGs with CSS (#1399)
- Drop jQueryUI
- Make clean properly removes only what it should now
- Improve the copy on reply notification emails
- Restyle tags (#1430, #1435)
- Various other usability and style enhancements (#1354, #1410, #1414)

Known Issues

- Searching for tags with spaces does not work
- Standalone annotation page shows stream updates
- Sphinx documentation is broken

0.2.2 (2014-08-15)

Fixes

- Fix user search (#1391)
- Fix page search

Known issues

- In some circumstances, Firefox can freeze on initial load.

0.2.1 (2014-08-11)

- Revert to using MANIFEST.in so built assets get shipped in the source release.

0.2.0 (2014-08-10)

- Improve usability of the toolbar (#1268, #1316)
- Make the stream cards interactive (#1281, #1290)
- Make the annotation card on a standalone annotation page interactive (#427)
- Fix race conditions with realtime updates (#1306, #1307)
- Exponential backoff on socket failures (#1291)
- Fix infinite scroll regression on stream
- Add a front end test framework (#240, #1309)
- Revalidate forms when autofilled (#374)
- Introduce environment variable overrides for important settings
- Allow bundling assets of a debug build in the extension (#1230)
- Make it possible to override all templates (#1337)
- Simplify the search entry, getting rid of visualsearch.js (#1326)
- Fix infinite scroll in stream (#1373)
- Fix several reports of broken styles on certain sites (#1372)
- Factor out the identity, session, and authentication system into its own package, making it entirely optional (#1357)
- Remove PDF.js from the Chrome extension until it can be made opt-in (#1384)
- Rework the reply notification emails – still disabled by default (#1378)

0.1.3 (2014-07-14)

- Include missing package data

0.1.2 (2014-07-14)

- Include package data in source distributions

0.1.1 (2014-07-14)

- Fix versioneer issue with Python release packages

0.1.0 (2014-07-11)

- Searchable stream (#719)
- Sidebar search (#606)
- Realtime updates (#356)
- Private annotations and highlights (#530)
- Page level comments (#115)
- Support for tags on annotations (#514)
- Support for annotating PDF.js viewers (#74)
- Chrome and Firefox extensions (#43)
- Addition of unit tests for some modules (#240)
- Support for sharing a sidebar between frames (#778)
- Improved URI search (#1243)
- Improved authentication form errors (#1279)
- Pluggable authentication via pyramid_multiauth (#1167)

0.0.6 (2013-01-08)

- Flash messages (#233)
- Static asset build script (#161)
- Finish registration form flow (#159)
- Separate detail and bucket views (#162)
- Slide-over detail view (#150)

0.0.5 (2012-11-27)

- Use AngularJS (#198)
- Confirm discarding of drafts (#188)
- Markdown support (#91)
- Resizable sidebar (#26)

0.0.4 (2012-11-6)

- Refactoring of horus and SCSS
- Reply counts on threads
- Visual improvements

0.0.3 (2012-10-16)

- Up/down tabs are hidden when count is zero
- Long excerpts are truncated and show with less/more links
- New persona dropdown (tinyman)
- Password reset fixed
- Initial sphinx documentation added

0.0.2 (2012-10-09)

- Replace account system
- Threaded replies
- Sidebar iframe
- Release management
- Miscellaneous gardening

0.0.1 (2012-04-16)

- Set up scaffolding, accounts, annotator

C

Client ID, [4](#)
Client secret, [4](#)

D

DEBUG_QUERY, [11](#)

E

environment variable
 CLIENT_URL, [20](#)
 DEBUG_QUERY, [11](#)
event, [6](#)

R

reply, [6](#)