Ethical Ad Server

Release 0.40.0

Oct 21, 2021

Installation Guide

The Ethical Ad Server is an advertising server without all the tracking. It was created by Read the Docs to serve the ads on Read the Docs.

CHAPTER 1

Features

- Supports image, image+text, and text-only ads
- Extensive and extensible ad fraud prevention
- Reports based on campaign, flight, or individual ad
- DoNotTrack ready
- GDPR ready
- Supports many targeting parameters options including:
  - geographical targeting by country and state/province
  - mobile, tablet, or desktop targeting
  - extensible custom targeting options

The Ethical Ad Server uses GeoLite2 data created by MaxMind or IP Geolocation by DB-IP.

CHAPTER 2

Use our existing ad network

This code is what powers our EthicalAds network. If you are looking for a way to serve ethical ads to developers, we highly recommend that you give our existing network a try. If you are interested in handling your own ads sales while serving ads ethically, please get in contact with us as we are currently building a solution for you!

CHAPTER 3

Installation Guide

If you are looking to install your own instance of the Ethical Ad Server, these docs are for you.

3.1 Installation

The Ethical Ad Server is intended to be run on any host that can run a Dockerized application, such as Heroku, AWS Elastic Beanstalk, Azure App Service, or on your own infrastructure. It can also be run directly on virtual machines. It is intended to be run with:

- PostgreSQL
- Redis

3.1.1 Configuring the ad server

The ad server is configured by environment variables. See the *document outlining them*.

3.1.2 Set the ad server URL

After configuring the ad server and setting up the database, you'll need to set the URL for the ad server. This URL is used to create links to the ad server when clicking or viewing ads.

- Login to the *administration interface*.
- Under *Sites*, click on the first and only Site
- Set the domain to be a domain without scheme (eg. `server.ethicalads.io` not `http://...`)

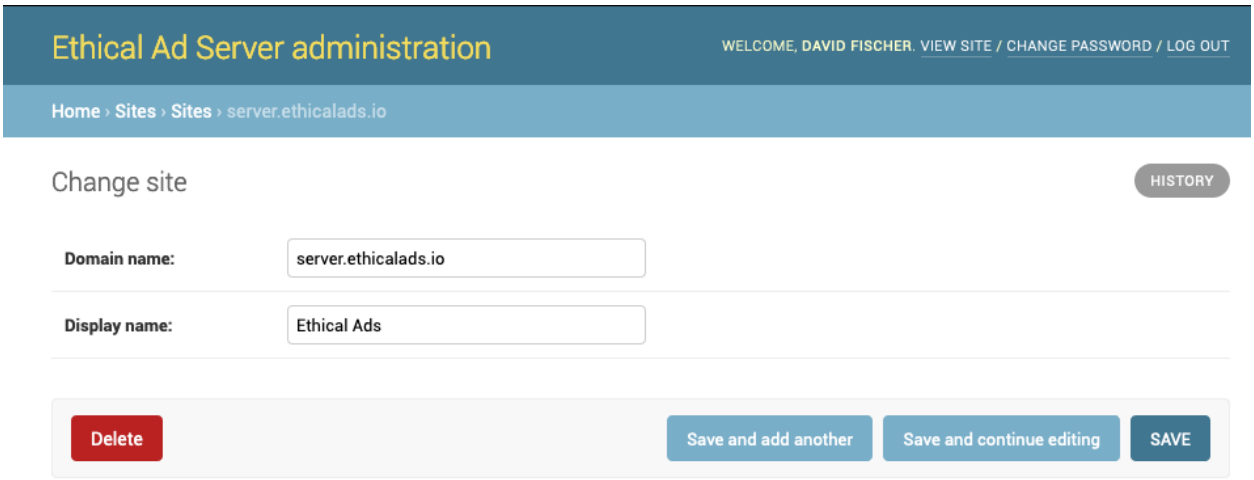

Fig. 1: Configuring the ad server URL

3.1.3 Building the Docker image

Production Docker support

The process of setting up your own production installation is not supported by us. The Docker image is provided as a template for your setup.

Building the Docker image is only necessary if you need to heavily customize the ad server. To build this, you'll need to have Docker installed and you'll probably want the GeoIP database command `geoipupdate` installed and configured so that the ad server can convert IP addresses to cities and countries for ad targeting purposes.

```
$ make geoip dockerprod
```

3.2 Configuration

The server is intended to be configured by setting **environment variables**. Most web hosts that will run a Dockerized application, such as Heroku, AWS Elastic Beanstalk, or Azure App Service, have ways to set environment variables in line with the Twelve Factor App.

3.2.1 Environment variables

This lists the commonly configured environment variables. For a complete list, please see `config/settings/production.py` for details.

There are a few required environment variables and the server will not start without them:

- *ALLOWED_HOSTS*
- *DATABASE_URL*
- *REDIS_URL*
- *SECRET_KEY*

- *SENDGRID_API_KEY*

ADSERVER_ADMIN_URL

Set to a unique and secret path to enable the *administration interface* (a lightly customized Django admin) at that path. For example, if this is set to `admin-path` then the admin interface will be available at the URL `http://adserver.example.com/admin-path/`. By default, this set to `/admin`.

ADSERVER_BLOCKLISTED_USER_AGENTS

Set this to a comma separated list of strings that are looked for anywhere in the User Agent of an ad request. Any user agents matching any of these will be completely ignored for counting clicks and views for billing purposes.

ADSERVER_BLOCKLISTED_REFERRERS

Set this to a comma separated list of strings that are looked for anywhere in the Referrer of an ad request. Any referrer matching any of these will be completely ignored for counting clicks and views for billing purposes.

ADSERVER_CLICK_RATELIMITS

Set this to a comma separated list of formats to specify how quickly a single IP can click on multiple ads. Clicks that happen faster than any of these rates will still click-through, but won't count toward billed clicks. By default, this is set to `"1/m,3/10m,10/h,25/d"` which is:

- 1 click per minute
- 3 click per 10 minutes
- 10 clicks per hour
- 25 clicks per day

ADSERVER_VIEW_RATELIMITS

Set this to a comma separated list of formats to specify how quickly a single IP can view ads. Views that happen faster than any of these rates will still allow viewing the ad or clicking through, but won't count toward billed impressions. By default, this is set to `"3/5m"` which is:

- 3 views per 5 minutes

ADSERVER_DECISION_BACKEND

Set to a dotted Python path to a decision backend to use for the ad server. Different publishers and ad networks may want different backends based on how different ads should be prioritized. For example, you may want to prioritize ads with the highest CPM/CPC or prioritize the most relevant. Defaults to `adserver.decisionengine.backends.ProbabilisticFlightBackend`, a backend that chooses ads based on how many more clicks and views are needed.

Set to `None` to disable all ads from serving. This can be useful during migrations.

ADSERVER_HTTPS

Set to `True` to enforce some security precautions that are recommended when run over HTTPS:

- The session and CSRF cookie are marked "secure" (not transmitted over insecure HTTP)
- HSTS is enabled

ADSERVER_RECORD_VIEWS

Whether to store metadata (a database record) each time an ad is viewed. This is `False` by default and can result in a bloated database and poor performance. It's `True` by default in development. This can be overridden on a per publisher basis by setting the `Publisher.record_views` flag.

ADSERVER_STICKY_DECISION_DURATION

Duration in seconds to show the same ad to the same user if multiple ads are requested in short succession. This ad stickiness helps make sure that quick navigation doesn't result in wasted ad views and it ensures that the correct ad view is attributed to a clickthrough if it occurs.

The default in production is 5 seconds.

ADSERVER_SUPPORT_TO_EMAIL

An email address where support email should go to. By default, it sends to `support@` followed by the domain configured on the `SITE_ID` (usually site #1).

ADSERVER_SUPPORT_FORM_ACTION

If set, the support form will submit to this external URL instead of sending email. This can be used to connect the support form to an external help desk.

ALLOWED_HOSTS

This setting will adjust Django's `ALLOWED_HOSTS` setting. Set this to the host you are using (eg. `server.ethicalads.io,server2.ethicalads.io`).

DATABASE_URL

This will set the address of the database used by the ad server. While any database supported by Django will work, PostgreSQL is preferred (eg. `psql://username:password@127.0.0.1:5432/database`) See Django's database documentation and the DATABASES setting for details.

DEBUG

This setting will turn on Django's `DEBUG` mode. It should be off in production (which is the default). Set to `True` to enable it.

DEFAULT_FILE_STORAGE

Adjusts Django's `DEFAULT_FILE_STORAGE` setting. Defaults to `storages.backends.azure_storage.AzureStorage` which can be used to storage uploaded ad images in Azure. See Django's storage documentation for details.

ENFORCE_HOST

If set, all requests to hosts other than this one will be redirected to this host. In production, this is typically `server.ethicalads.io`.

INTERNAL_IPS

This setting will adjust Django's `INTERNAL_IPS` setting. This setting has a few additional meanings for the ad server including:

- All ad impressions and clicks from `INTERNAL_IPS` are ignored for reporting purposes

REDIS_URL

A Redis cache is required to operate the ad server. The Redis connection is specified in URL format such as `redis://redis:6379/0`.

SECRET_KEY

This required setting will be your Django `SECRET_KEY`. Set this to something random like 50 random alphanumeric characters and keep it a secret. The server will refuse to start without this.

There are a few implications to changing this setting in a production deployment including:

- All sessions will be invalidated (everyone gets logged out)
- Password reset tokens are invalidated

SENDGRID_API_KEY

Set this to your Sendgrid API key to enable sending email through Sendgrid.

STRIPE_SECRET_KEY

Sets up the Stripe API where advertisers can be connected to a Stripe customer and invoices created directly through the ad server. Invoices are created in the *admin interface*.

3.2.2 Overriding settings entirely

While most options can be set by tuning environment variables, for a complex setup, you might consider completely overriding the settings.

To completely override the settings, create a new file `config/settings/mysettings.py` which should extend from `config/settings/base.py` and then you'll need to set the environment variable

`DJANGO_SETTINGS_MODULE` to `config.settings.mysettings` (note that the path is separated by dots and there is no file extension).

Once this is done, other *Environment variables* will be configured in your new `mysettings.py` rather than with environment variables.

CHAPTER 4

User Guide

This section of the docs has specific how-to guides to get the most from the Ethical Ad Server.

4.1 Getting Started

After the ad server is installed and you've *Set the ad server URL*, you can begin to create new user accounts and setup ad campaigns.

Unlike setting up something like Google Ad Words or many other ad platforms, you don't just copy a JavaScript blob into your pages. Instead, this ad server is run "server-to-server" meaning that the publisher's server (where the ads are shown) connects to the ad server to get an ad. This allows features like server-side rendering of ads rather than rendering them with JavaScript.

In the future, we may include a way to get ads with pure JavaScript but that isn't a high priority feature right now.

4.1.1 Understanding the ad server modeling

There's a few models and terminology to understand that will help understand how the ad server is set up:

Advertisers These are individual companies that are advertising on the server. This could include a house advertiser run by the ad server.

Publishers These are various sites where ads are shown. When an ad is requested, it is requested by a publisher. Advertisers can limit which publishers can show their ads. Revenue is tracked individually per publisher.

Campaigns Campaigns generally represents a group of ad flights from an advertiser. No ad targeting is done at the campaign level. While most campaigns are "paid", ad server admins can configure "affiliate", "house" or "community" campaigns which have a lower priority than paid (Paid > Affiliate > Community > House). Most advertisers will only have 1 campaign (but multiple Flights) although sometimes multiple are needed if the advertiser has both "paid" and "community" ads or if they want to have different campaigns for different Publishers.

Flights Flights handle the budgeting and targeting for an individual ad buy from an Advertiser. For example, an advertiser might buy $500 worth of ads at $1 CPM with certain keyword or geographic targeting. The flight is a

group of ads that are all part of the same ad buy and have the same targeting. This allows an advertiser to easily run multiple ads and find the best performing subset.

Advertisements This is an individual ad that can have an image, text, and a destination URL associated with it.

Ad types These are different ad types (and custom types) that the ad server supports. Ad types specify the parameters for an ad like whether it has an image or how long the text can be. Ad types can also control how an ad gets rendered and an ad can have one or more ad type.

Ad Impressions Ad impressions store quantity of views and clicks for each ad on each publisher each day.

Clicks Each time an ad is clicked and the click is "billed", a Click record is written to the database with the page where the click occurred, the publisher, datetime, the specific ad, and some metadata about the user such as an anonymized IP and anonymized user agent. A click is considered "billed" regardless of whether the click cost money for a CPC flight or not as long as the following conditions are met:

- This isn't a duplicate click
- The user isn't rate limited
- The user agent is a browser, not a known "bot", and not one of the *ADSERVER_BLOCKLISTED_USER_AGENTS*
- The flight targeting (which is rechecked) matches
- The user is not logged in as an advertiser, publisher, or staff
- The IP isn't one of the *INTERNAL_IPS* or a known proxy
- The referrer is valid and isn't one of the *ADSERVER_BLOCKLISTED_REFERRERS*

Views Just like clicks are stored each time an ad is clicked, it is possible to do the same each time an ad is viewed. By default, this is off in production and there should be no metadata on individual views. However, it can be enabled by toggling *ADSERVER_RECORD_VIEWS*.

4.1.2 Creating more users

Additional login accounts for staff, advertisers, and publishers are created in the *administration interface* under *Ad Server Auth > Users*. Users can be associated with specific advertisers or publishers. This will reduce the actions and reports that a specific user can see.

Staff users can see reports for all advertisers and publishers.

4.1.3 Setting up advertisements

Advertisers, campaigns, flights, and individual ads are created in the *administration interface* under *Ad Server Core*.

For the very first time, you'll need to create a record for an advertiser and a campaign. Then you can create a flight. Flights are where the details of the ad buy are stored such as how many clicks (CPC) or impressions (CPM) were purchased at a specific price. This is also where the targeting for a set of ads is configured.

Once an ad flight is configured, one or more ads can be setup for that flight. These are configured in the same interface.

Once the ads are setup, requests for an *ad decision* will pick up your new ads assuming the targeting matches.

4.1.4 Reporting

Reporting tables are available immediately upon logging in. Access to publisher or advertiser reports are restricted to users who have access to them.

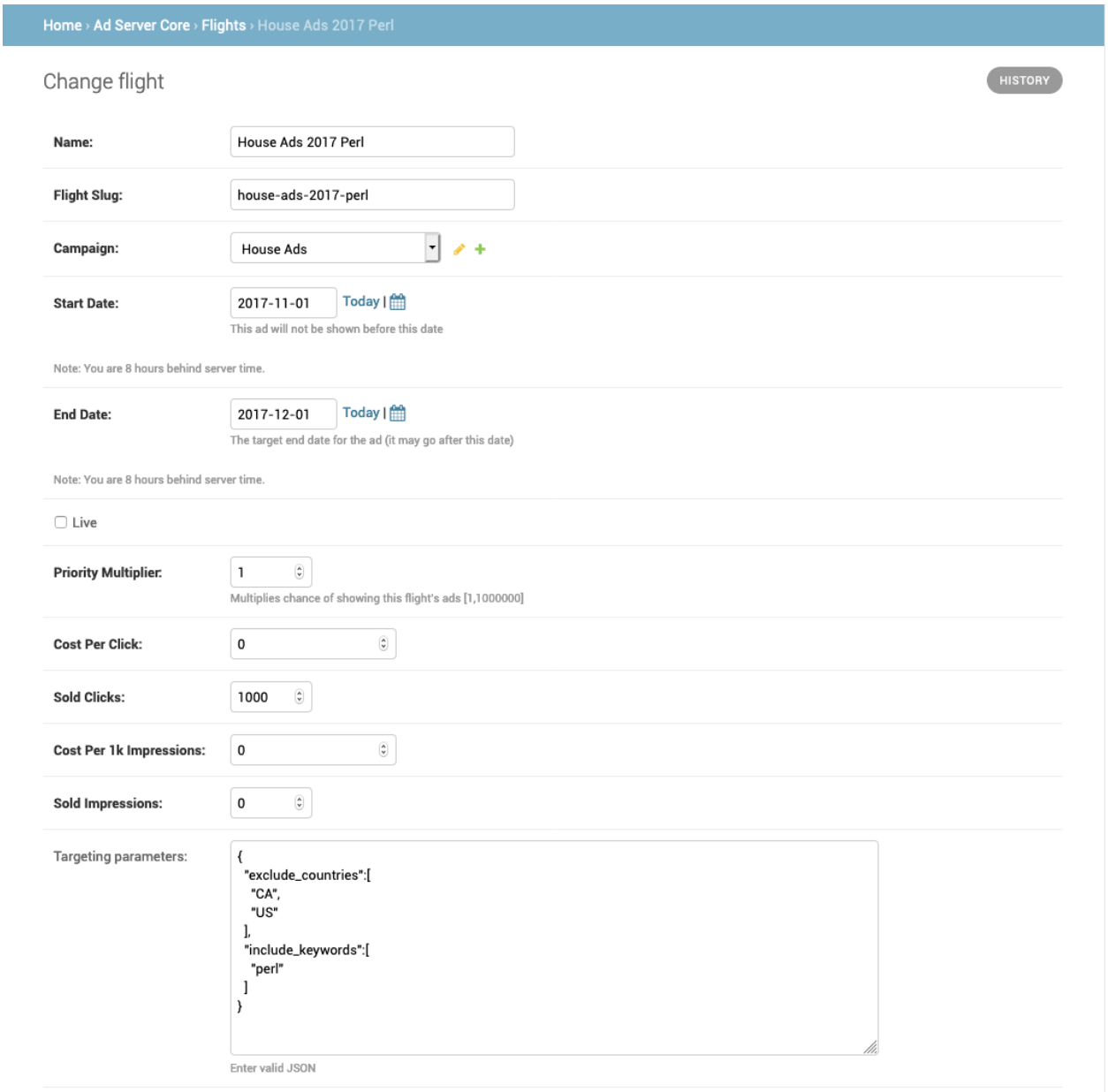

Fig. 1: Configuring an ad flight

4.2 Administering

More interfaces are actively being built, but most data is currently entered in this administration interface. Staff users get a link to the admin interface in the usual dashboard that shows reports (in the upper right menu). The URL is `/admin` by default but this can be customized by setting *ADSERVER_ADMIN_URL*

4.2.1 Deleting data

Most advertising records in the ad server such as clicks, impressions, advertisers, advertisements cannot be deleted through the administration interface after they are created. This is by design so that billing data is never deleted in the system. Ads and flights can be deactivated so they aren't used, but advertiser data is not deleted.

If you absolutely must delete data, you'll have to go to the database directly.

4.2.2 Invoicing advertisers

Assuming an advertiser has a connected Stripe Customer ID, invoices can be created for an advertiser directly from the ad server. In the advertiser admin section or in flight administration (for admins, not advertisers), select "Create draft invoice" from the actions dropdown, select an advertiser or the flights to invoice for, and click Go. This will create a draft invoice for the advertiser in Stripe which can customized further and sent.

4.2.3 Processing refunds

For billed clicks and views that need to be refunded or credited back to the advertiser, there is an administration action which will correctly update all the relevant models and mark the impression as refunded so it can't be refunded again.

Go to *Ad Server Core* > *Views* or *Clicks*, select the impressions to refund, choose the refund action from the dropdown, and hit "Go".

4.3 API

4.3.1 Authentication

We require authentication on some requests, but JSONP requests don't require them. Authentication works in one of two ways:

- For requests made in your web browser, your authenticated session will be used
- For all other requests, you need to use the `Authorization` HTTP header:

```
Authorization: Token 0000000000000000000000000000000000000000
```

  Where the value of the header is the string "Token" followed by a space and your 40 character API key.

Note: Creating an API token can be done under *Username Dropdown* > *API Token*.

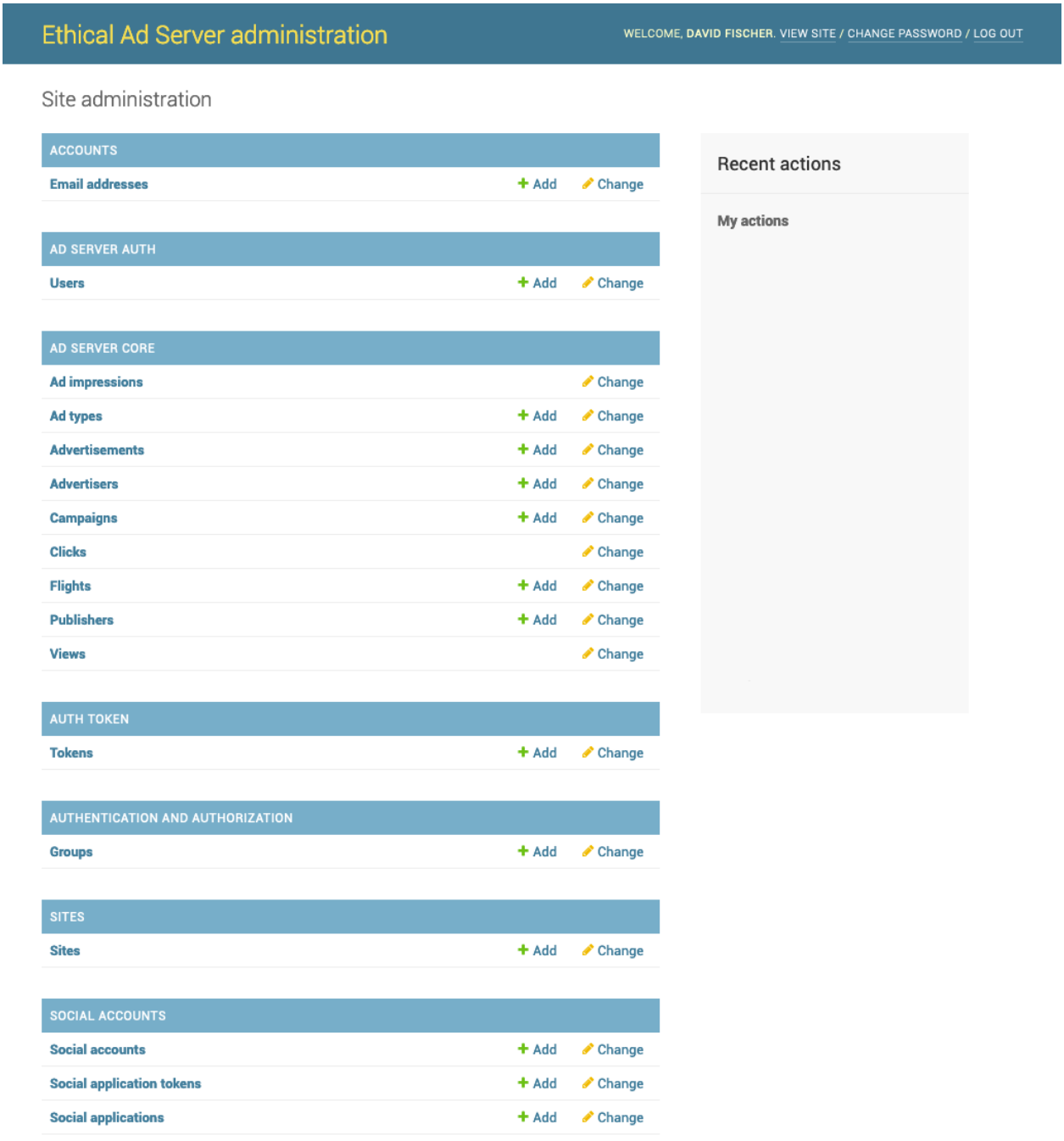

Fig. 2: The admin interface

4.3.2 Ad decision

class adserver.api.views.**AdDecisionView**(***kwargs*)

Make a decision on an *Advertisement* to show.

The ad decision is based on

- the publisher (ad campaigns may be publisher specific)
- the available placements (ad types and priorities)
- minimal user agent details (browser, mobile, operating system)
- geography (based on IP)
- keywords

GET /api/v1/decision/

Request an advertisement for a specific publisher.

The publisher must be explicitly permitted to allow unauthenticated requests. This is typically used as a JSONP call.

Request JSON Object

- **publisher** (*string*) – **Required**. The slug of the publisher. If using the ethical-ad-client, this comes from `data-ea-publisher`.
- **div_ids** (*string*) – A `|` delimited string of on-page ids. The number and order must correspond to the `ad_types`
- **ad_types** (*string*) – A `|` delimited string of ad types. The number and order must correspond to the `div_ids`.
- **priorities** (*string*) – An optional `|` delimited string of priorities for different ad types. The number and order matter, applying to `div_ids` and `ad_types`.
- **keywords** (*array*) – An optional `|` delimited string of case-insensitive keywords that describe content on the page where the ad is requested (eg. `python|docker|kubernetes`). Used for ad targeting and is additive with any publisher settings.
- **campaign_types** (*array*) – An optional `|` delimited string of campaign types (eg. `paid|community|house`) which can be used to limit to just certain types of ads. Can only further reduce campaign types, not allow ones prohibited for the publisher.
- **url** (*string*) – The URL of the requesting page. This is where the ad will appear.
- **format** (*string*) – Format can optionally be specified as `jsonp` to allow a callback.
- **callback** (*string*) – The name of the callback for a JSONP request (default is `callback`)
- **force_ad** (*string*) – Limit results to a specific ad
- **force_campaign** (*string*) – Limit results to ads from a specific campaign

Response JSON Object

- **id** (*string*) – The advertisement slug of the chosen ad
- **text** (*string*) – The HTML text of only the ad without any images (see `html` for full HTML)
- **body** (*string*) – The text of the ad, stripped of any HTML.

- **copy** (*object*) – A breakdown of the ad into its `content`, optional `headline` and optional call to action (`cta`). These fields are always plain text.
- **html** (*string*) – An HTML rendering of the ad
- **link** (*string*) – A click URL for the ad
- **view_url** (*string*) – A view URL to count an ad view
- **view_time_url** (*string*) – A URL endpoint that updates how long the ad was viewed
- **nonce** (*string*) – A one-time nonce used in the URLs so the ad is never double counted
- **display_type** (*string*) – The slug of type of ad (eg. sidebar)
- **div_id** (*string*) – The ID where the ad will be inserted
- **campaign_type** (*string*) – The type of campaign this as is from (eg. house, community, paid)

An example:

```
# Multiple type options
{
    "ad_types": "readthedocs-fixed-footer|readthedocs-sidebar",
    "div_ids": "text-div|image-div"
    "priorities": "3|5"
}

# Simplest case
{
    "ad_types": "readthedocs-sidebar",
    "div_ids": "sample-div"
}
```

POST /api/v1/decision/

Authentication is required for this endpoint. The POST version of the API is similar to the GET version with only a few changes:

Request JSON Object

- **publisher** (*string*) – **Required**. The slug of the publisher.
- **placements** (*array*) – **Required**. Various possible ad placements where an ad could go. This is a combination of `div_ids`, `ad_types`, and `priorities` in the GET API. Only one ad will be returned but you can prioritize one type of ad over another.
- **keywords** (*array*) – Case-insensitive strings that describe the page where the ad will go for targeting
- **campaign_types** (*array*) – Limit the ad results to certain campaign types.
- **user_ip** (*string*) – User's IP address used for targeting (the requestor's IP will be used if not present)
- **user_ua** (*string*) – User's user agent used for targeting (the requestor's UA will be used if not present)

The response is the same as the GET request above.

An example:

```
{
    "publisher": "your-publisher",
    "placements": [
        {
            "div_id": "ad-div-1",
            "ad_type": "image-v1",
            "priority": 10,
        }
    ],
    "campaign_types": ["paid"],  # request PAID ads only
    "keywords": [
        "python",
        "docker",
        "kubernetes",
    ],
}
```

4.3.3 Publisher APIs

class adserver.api.views.**PublisherViewSet**(***kwargs*)
Publisher API calls.

GET /api/v1/publishers/
Return a list of publishers the user has access to

Response JSON Object

- **count** (*int*) – The number of publisher returned
- **next** (*string*) – A URL to the next page of publisher (if needed)
- **previous** (*string*) – A URL to the previous page of publisher (if needed)
- **results** (*array*) – An array of publisher results (see publisher details call)

GET /api/v1/publishers/(**str:** *slug*)**/**
Return a specific publisher

Response JSON Object

- **url** (*string*) – The URL to this report
- **name** (*string*) – The name of the publisher
- **slug** (*string*) – A slug for the publisher
- **created** (*date*) – An array of publisher results
- **modified** (*date*) – The date the publisher was last modified

GET /api/v1/publishers/(**str:** *slug*)**/report/**
Return a report of ad performance for this publisher

Query Parameters

- **start_date** (*date*) – Start the report on a given day inclusive. If not specified, defaults to 30 days ago
- **end_date** (*date*) – End the report on a given day inclusive. If not specified, no end time is used (up to current)

Response JSON Object

- **days** (*array*) – An array of publisher results per day
- **total** (*object*) – An object of aggregated totals for the publisher

4.3.4 Advertiser APIs

class adserver.api.views.**AdvertiserViewSet**(***kwargs*)
Advertiser API calls.

GET /api/v1/advertisers/
Return a list of advertisers the user has access to

Response JSON Object

- **count** (*int*) – The number of advertisers returned
- **next** (*string*) – A URL to the next page of advertisers (if needed)
- **previous** (*string*) – A URL to the previous page of advertisers (if needed)
- **results** (*array*) – An array of advertiser results (see advertiser details call)

GET /api/v1/advertisers/(**str:** *slug*)**/**
Return a specific advertiser

Response JSON Object

- **url** (*string*) – The URL to this report
- **name** (*string*) – The name of the advertiser
- **slug** (*string*) – A slug for the advertiser
- **created** (*date*) – An array of advertiser results
- **modified** (*date*) – The date the advertiser was last modified

GET /api/v1/advertisers/(**str:** *slug*)**/report/**
Return a report of ad performance for this advertiser

Query Parameters

- **start_date** (*date*) – Start the report on a given day inclusive. If not specified, defaults to 30 days ago
- **end_date** (*date*) – End the report on a given day inclusive. If not specified, no end time is used (up to current)

Response JSON Object

- **days** (*array*) – An array of advertiser results per day
- **total** (*object*) – An object of aggregated totals for the advertiser
- **flights** (*array*) – An array of flights for this advertiser in the time period

CHAPTER 5

Developer Guide

If you are developing the Ethical Ad Server, then these are the docs for you.

5.1 Quickstart

5.1.1 Developing with Docker compose

To build a local Docker compose environment:

```
# This command can take quite a while the first time
$ make dockerbuild
```

Start a local multi-container application with Postgres, Redis, Celery, and Django:

```
$ make dockerserve
```

To get a shell into the Django container where you can run `./manage.py createsuperuser`, get a Django shell, or run other commands:

```
$ make dockershell
...
/app # ./manage.py createsuperuser
```

After setting up your super user account on your local development instance, you'll still need to set the *Set the ad server URL* to something like `localhost:5000`.

5.1.2 Developing locally

Docker compose is the recommended way to do development consistently. This section is more to document steps than to encourage you to develop outside of Docker.

Requirements

- Python 3.6
- Nodejs (tested with v12.3)

Front-end assets

To build the assets:

```
$ npm install
$ npm run build
```

Install Python dependencies

```
$ pip install -r requirements/development.txt
$ pre-commit install            # Install a code style pre-commit hook
```

Run the server

Run migrations:

```
$ python manage.py migrate
```

Create a superuser:

```
$ python manage.py createsuperuser
```

Run the server:

```
$ python manage.py runserver
```

5.1.3 Running the tests

To run the unit tests:

```
$ pip install -r requirements/testing.txt
$ make test
```

5.2 Changelog

5.2.1 Version v0.40.0

The big change in this release was that we're trying out some graphs. However, for this release, they are staff-only. Other than that, there was nothing user facing in this release.

date October 21, 2021

- @davidfischer: Charting/graphing with metabase (#475)

- @davidfischer: Remove the CTR publisher change alert (#473)
- @ericholscher: Show publisher name instead of slug in payout (#472)
- @davidfischer: Tweaks to the daily aggregation task (#471)
- @ericholscher: Make azure logging quiet (#470)
- @ericholscher: Fix a bug where existing AdType was excluded (#455)

5.2.2 Version v0.39.0

Most of this release were small bug fixes and tweaks to staff notifications.

date October 6, 2021

- @ericholscher: Force using the default DB during ad serving incr call (#467)
- @davidfischer: Small tweak to flight ordering (#466)
- @davidfischer: Fail silently on slack failures (#464)
- @davidfischer: Increase aggregation task time limit (#463)
- @davidfischer: Notify when daily reports are aggregated (#462)
- @ericholscher: Fix silly where bug data wasn't defined if we weren't caching. (#461)

5.2.3 Version v0.38.0

This release had a number of changes to support custom publishers and support for a read replica on our reporting.

date September 24, 2021

- @davidfischer: Fixes a bug with old-style ads (#458)
- @ericholscher: Add a read replica DB router & settings (#457)
- @ericholscher: Fix mailing list link. (#456)
- @ericholscher: Add ability to export region data (#454)
- @ericholscher: Update the link we're pointing to for CTR low messages (#452)
- @ericholscher: Add ability to uncache publisher ads (#451)
- @ericholscher: Fix payout filtering & show status in admin (#450)
- @davidfischer: When copying ads, put newest ads first (#448)
- @dependabot[bot]: Bump pillow from 8.2.0 to 8.3.2 in /requirements (#447)
- @davidfischer: Flight form improvements (#443)

5.2.4 Version v0.37.0

This release had a minor change to topic-based reporting only.

date September 13, 2021

- @ericholscher: Add *other* to the list of topics when none other apply. (#446)

5.2.5 Version v0.36.0

The big change in this release was to revamp our reporting to be more focused on topic and region rather than individual keywords and countries/regions. This should make be much faster than the previous geo and keyword reports which will be phased out.

date August 31, 2021

- @davidfischer: More tweaks to publisher notifications (#444)
- @ericholscher: Add "Stay updated" to the top of the payout email (#442)
- @ericholscher: Tweaks payouts with issues that we've found (#441)
- @ericholscher: Make advertiser flight ads linkable (#440)
- @ericholscher: Add StaffRegionReport (#431)
- @ericholscher: Make report queries faster (#376)

5.2.6 Version v0.35.0

The main change in this release involved the server side changes to store how long an ad is viewed. We believe this is a cool metric to show to advertisers and may separate us from competition and generate higher revenues for publishers.

date August 13, 2021

- @ericholscher: Fix silly bug with Payouts (#438)
- @davidfischer: Minor tweaks around view time (#437)
- @dependabot[bot]: Bump path-parse from 1.0.6 to 1.0.7 (#436)
- @davidfischer: Remove server side analytics which we weren't using (#435)
- @davidfischer: Fix the build (#434)
- @decaffeinatedio: No results from decision API despite valid(?) configuration (#432)

5.2.7 Version v0.34.0

This release had no significant user-facing changes. All the changes involved staff interfaces, staff notifications, or documentation.

date August 4, 2021

- @davidfischer: Fix form submission for flights with no targeting (#429)
- @davidfischer: Note that the prod dockerfile is unmaintained (#428)
- @decaffeinatedio: Update GeoIP Links (#427)
- @decaffeinatedio: Error when running *make dockerprod* (#426)
- @davidfischer: Interface to create a new flight (#425)
- @davidfischer: Improve difference notifications (#422)
- @ericholscher: Add option of *created* sort on Staff publisher report (#421)

5.2.8 Version v0.33.0

We added `noopener` to our ad links as a security precaution. The other big change was to allow ad types to be publisher (group) specific. We already have publisher specific ad types as Read the Docs has a compatible but slightly different ad format from EthicalAds. Some possible new publishers also expressed interest.

date July 22, 2021

- @davidfischer: Add permissions to see staff-only report fields (#419)
- @ericholscher: Use the right payout objects when finishing (#417)
- @davidfischer: Add noopener to external links (#416)
- @davidfischer: Raise a warning after validating landing pages (#415)
- @davidfischer: Publisher (group) specific ad types (#412)
- @davidfischer: Validate ad landing page gives a 200 (#175)

5.2.9 Version v0.32.0

Mostly we added some new staff additions to help with payouts and help manage targeting. We also added some callouts to help refer publishers. Lastly, we did add a task to send Slack notifications to staff when publisher metrics change significantly week to week.

date July 15, 2021

- @ericholscher: Add a more obvious callout for the publisher referral in payouts (#413)
- @ericholscher: Add some payout optimizations to make it faster (#411)
- @davidfischer: Notify when publisher metrics change (#410)
- @davidfischer: Initial staff interface for flight targeting and size updates (#409)

5.2.10 Version v0.31.0

This release adds a new staff-only interface to manage publishers. It also adds the ability to notify via Slack when a campaign completes. Currently, these notifications are just for staff but in the future we could allow notifications for advertisers as well.

date June 30, 2021

- @davidfischer: Send Slack notifications on completed flights (#407)
- @dependabot[bot]: Bump color-string from 1.5.3 to 1.5.5 (#406)
- @ericholscher: Add Staff Add Publisher View (#405)
- @ericholscher: Fix float data in payout form (#404)
- @dependabot[bot]: Bump set-getter from 0.1.0 to 0.1.1 (#403)
- @dependabot[bot]: Bump striptags from 3.1.1 to 3.2.0 (#402)

5.2.11 Version v0.30.0

This release added change tracking to most models and minor payout workflow improvements.

date June 17, 2021

- @ericholscher: Clean up a number of payout workflow issues (#400)
- @davidfischer: Track historical changes to select models (#399)
- @dependabot[bot]: Bump postcss from 7.0.17 to 7.0.36 (#398)

5.2.12 Version v0.29.0

This release improves payouts in the adserver, adds a RegionTopic index for improved reporting, and starts weighting CPC ads to publishers with higher CTR.

date June 15, 2021

- @davidfischer: This process is consuming the server (#396)
- @davidfischer: Updates the weighting algorithm (#395)
- @ericholscher: Add initial Staff Payouts view (#394)
- @davidfischer: Release v0.28.0 (#393)
- @dependabot[bot]: Bump django from 2.2.20 to 2.2.24 in /requirements (#392)
- @dependabot[bot]: Bump django from 2.2.20 to 2.2.22 in /requirements (#391)
- @dependabot[bot]: Bump pillow from 8.1.1 to 8.2.0 in /requirements (#390)
- @ericholscher: Add RegionTopic index modeling (#388)

5.2.13 Version v0.28.0

The biggest new changes here are a task to null out some old data periodically and a staff actions interface.

date June 10, 2021

- @dependabot[bot]: Bump django from 2.2.20 to 2.2.24 in /requirements (#392)
- @dependabot[bot]: Bump django from 2.2.20 to 2.2.22 in /requirements (#391)
- @dependabot[bot]: Bump pillow from 8.1.1 to 8.2.0 in /requirements (#390)
- @dependabot[bot]: Bump django from 2.2.20 to 2.2.21 in /requirements (#389)
- @davidfischer: Move the add advertiser interface to a staff action (#387)
- @davidfischer: Null out old client IDs (#386)
- @dependabot[bot]: Bump browserslist from 4.6.6 to 4.16.6 (#385)
- @davidfischer: Front form tweaks (#384)

5.2.14 Version v0.27.0

This release added some additional staff-only reports to understand advertising data. It also included a support form for advertisers and publishers to get in touch.

date May 17, 2021

- @davidfischer: The reports sometimes wrap the date ranges awkwardly (#382)
- @davidfischer: Setup a support form (#381)
- @davidfischer: I missed this when adding CTR to advertiser reports (#380)

- @dependabot[bot]: Bump hosted-git-info from 2.8.8 to 2.8.9 (#379)
- @dependabot[bot]: Bump lodash from 4.17.19 to 4.17.21 (#378)
- @ericholscher: Add geo & keyword staff reports (#375)

5.2.15 Version v0.26.0

This release included advertiser dashboard improvements. Advertisers can invite other users at their company to work with them on advertising. We also added some minor filtering and reporting improvements. There is also a migration to ensure certain fields are unique.

date May 5, 2021

- @davidfischer: Allow filtering advertiser reports by flight (#374)
- @davidfischer: Allow advertisers to control their authorized users (#373)
- @davidfischer: Ensure slugs are unique (#372)
- @davidfischer: Copy/Re-use an existing ad (#371)
- @davidfischer: Show upcoming flights on the overview screen (#370)
- @davidfischer: Silence the disallowed host logger again (#369)
- @davidfischer: Don't reject invalid values in the URL field (#368)

5.2.16 Version v0.25.0

The big change here is that the ad decision API now supports sending the URL where the ad will appear. In the future, we can use this for some additional targeting and automated fraud checking.

date April 20, 2021

- @dependabot[bot]: Bump ssri from 6.0.1 to 6.0.2 (#366)
- @davidfischer: Add an optional URL to the decision API (#365)
- @ericholscher: Add link to FAQ in CTR callout in payout email (#364)
- @davidfischer: Send URL with the ad request (#354)

5.2.17 Version v0.24.0

In our reporting interface, we added some more summary and high level data on ad and flight performance from a CTR perspective. The other big change was a tweak to ad prioritization to prioritize higher eCPM ads when making an ad decision.

date April 15, 2021

- @davidfischer: Mute the disallowed host logger in prod (#362)
- @dependabot[bot]: Bump django from 2.2.18 to 2.2.20 in /requirements (#361)
- @ericholscher: Add naive attempt at price targeting (#360)
- @davidfischer: Show CTR in summaries for ads and flights (#358)
- @davidfischer: Create security policy (#356)
- @davidfischer: Tweaks to the archive management command (#355)

- @davidfischer: Update JS dependencies (#347)

5.2.18 Version v0.23.0

The big change in this release was to add overview screens for advertisers and publishers. Another change was to include a `ea-publisher` query parameter with ad clicks. This release also had some minor UX improvements to the reporting interface and a few other minor changes.

date April 1, 2021

- @davidfischer: Reporting UX improvements (#351)
- @davidfischer: Advertiser/publisher overview screens (#350)
- @dependabot[bot]: Bump y18n from 4.0.0 to 4.0.1 (#349)
- @davidfischer: Add publisher query parameter to ad clicks (#348)
- @davidfischer: Changes needed now that cryptography requires rust (#346)
- @ericholscher: Tweaks payouts more (#345)
- @davidfischer: Advertiser overview page (#174)
- @davidfischer: Publisher overview page (#173)

5.2.19 Version v0.22.1

This was a tweak to the stickiness feature that rolled out earlier today.

date March 19, 2021

- @davidfischer: Tweaks to the new stickiness factor (#342)

5.2.20 Version v0.22.0

The main feature in this release was to make sticky ad decisions. This will make the same ad appear for the same user for a certain amount of time (default 15s) even if they load new pages.

date March 19, 2021

- @dependabot[bot]: Bump pillow from 7.1.2 to 8.1.1 in /requirements (#340)
- @dependabot[bot]: Bump django from 2.2.13 to 2.2.18 in /requirements (#339)
- @davidfischer: Enable sticky ad decisions (#338)
- @davidfischer: Fix the geo report (#337)

5.2.21 Version v0.21.0

This release fixes a bug in report sorting and adds a management command to archive offers

date March 15, 2021

- @ericholscher: Sort indexes based on raw data vs. display (#333)
- @davidfischer: Archive offers management command (#332)
- @dependabot[bot]: Bump elliptic from 6.5.3 to 6.5.4 (#331)

5.2.22 Version v0.20.0

This release made some small reporting updates primarily for performance reasons.

date March 8, 2021

- @davidfischer: Remove refunded offers from aggregate reports (#329)
- @davidfischer: Total revenue report improvements (#328)
- @ericholscher: Make the Geo report a bit faster (#326)
- @ericholscher: Calculate Fill Rate against only paid offers (#325)
- @ericholscher: Add debug flag to payout command (#324)
- @ericholscher: Publisher report cleanup (#323)
- @davidfischer: Uplift report updates (#319)

5.2.23 Version v0.19.1

This release is primarily bug fixes and minor changes to when scheduled tasks are run.

date March 3, 2021

- @davidfischer: Remove hourly report updates. (#321)
- @davidfischer: Fix off by 1 (actually 2) error in ad text size (#320)
- @davidfischer: Run previous days reports automatically (#318)
- @davidfischer: Fix a bug in the uplift report (#317)

5.2.24 Version v0.19.0

Most of these changes were minor quality of life improvements for managing the ad server. It did involve a small dependency bump so it is a minor version increase.

date February 4, 2021

- @davidfischer: Minor testing changes (#315)
- @davidfischer: Don't count ad display when a particular ad is forced (#314)
- @dependabot[bot]: Bump bleach from 3.1.4 to 3.3.0 in /requirements (#313)
- @davidfischer: Show whats left on a flight always (#312)
- @davidfischer: Add a management command for creating advertisers (#311)
- @davidfischer: Fix a typo in the help text (#310)
- @davidfischer: Small admin improvements (#309)
- @davidfischer: Remove the link to DockerHub in the docs (#307)
- @davidfischer: Show top publishers for an ad flight (#172)

5.2.25 Version v0.18.1

This change included just a new constraint to prevent a DB race condition. Depending on your database, you may need to remove some records to apply the constraint. See the migration file for a query to get the records that need to be removed.

date January 19, 2021

- @davidfischer: Add a null offer constraint (#306)

5.2.26 Version v0.18.0

We made a change to make it a little easier for advertisers to have compelling ads. Advertisers can now declare a headline for an ad, a body, and a call to action and our default styles bold the headline and CTA. These fields are broken out in our JSON API as well for ads if publishers do custom integrations. No changes were made to existing ads in our system.

date December 17, 2020

- @davidfischer: Break the ad headline and CTA from the body (#302)

5.2.27 Version v0.17.0

The big user-facing change on this is to enable the publisher and geo reports for advertisers. There's also an easy option to exclude a publisher for an advertiser if requested.

date December 15, 2020

- @davidfischer: Add a backend option to exclude publishers for an advertiser (#300)
- @davidfischer: Enable the geo and publisher report for advertisers (#299)
- @davidfischer: Fix a few issues with refunding (#298)

5.2.28 Version v0.16.0

date December 1, 2020

This release contained some minor reporting changes and some admin-specific reports. We are testing some new advertiser reports (showing top geos, top publishers) but those are staff-only now but will likely roll out to all advertisers in the next release.

- @davidfischer: Advertiser reporting breakdowns (#295)
- @ericholscher: Add uplift reporting (#294)
- @ericholscher: Additional payout automation (#285)

5.2.29 Version v0.15.0

date November 24, 2020

There were a few minor fixes and refactors in this release. We are defaulting new publishers to use viewport tracking (#292), and we found a slight bug which was hotfixed related to Acceptable Ads uplift. There were significant internal changes to reporting to make creating new reports easier but these should not have significant user-facing changes.

- @ericholscher: Update a few model method defaults (#292)

- @davidfischer: Report refactor (#291)
- @ericholscher: Don't overwrite Offer on uplift (#290)

5.2.30 Version v0.14.0

date November 17, 2020

This version adds additional reporting around keywords and offer rate. Both of these are behind admin-only flags until we do more testing, but will likely be enabled in the next release.

- @ericholscher: Add keyword reporting for publishers (#286)
- @ericholscher: Add Decision modeling to our indexes (#274)

5.2.31 Version v0.13.0

date November 10, 2020

This version ships two new publisher reports: Geos and Advertisers. It also adds uplift tracking for Acceptable Ads tracking, allowing the server to be used for AA-approved ad networks.

- @ericholscher: Add uplift to Offers (#279)
- @ericholscher: Ship Geo & Advertiser reports to publishers (#278)
- @ericholscher: Don't pass *advertiser* to the all publishers reports. (#277)
- @dependabot[bot]: Bump dot-prop from 4.2.0 to 4.2.1 (#276)

5.2.32 Version v0.12.0

date November 3, 2020

None of the changes in this release are user facing. There are improvements to track and understand the fill rate for publishers (why some requests don't result in a paid ad) and another change to prepare to show publishers details of the advertisers advertising on their site.

- @ericholscher: Make Offers nullable to track fill rate (#272)
- @ericholscher: Add a new report for Publishers showing their advertisers (#271)
- @ericholscher: Add ability to sort All Publishers report by all metrics (#273)

5.2.33 Version v0.11.1

date October 29, 2020

This release adds the ability do to viewport tracking on publisher sites. It is managed on the backend via an admin setting, and we'll be slowly rolling it out to publishers.

- @ericholscher: Add a render_pixel option to the publisher. (#269)
- @davidfischer: Performance workaround for the offer admin (#267)

5.2.34 Version v0.11.0

Date October 27, 2020

This release adds Celery tasks for indexing of all our generated reporting indexes. We also added a Geo index in beta for this release, along with a few performance improvements.

- @davidfischer: Add an estimated count paginator (#265)
- @davidfischer: Add get_absolute_url methods to flight and advertiser models (#264)
- @ericholscher: Show breakdown report on the Geo/Placement reports by default (#263)
- @ericholscher: Remove unused entrypoint from dockerfile (#262)
- @ericholscher: Properly sort Countries in Geo report by most views (#261)
- @ericholscher: Migrate PlacementImpressions to a Celery task (#260)
- @ericholscher: Clean up Publisher settings (#259)
- @ericholscher: Cleanup celery config to work with beat (#258)
- @davidfischer: Index the date fields on ad impressions, clicks, views, and offers (#257)
- @ericholscher: Callout to EA (#256)
- @ericholscher: Add an initial Geo report for publishers (#244)

5.2.35 Version v0.10.2

Date October 1, 2020

v0.10.2 finally fixed the slow migration issues.

- @ericholscher: Make ad_type a slug on the AdBase & PlacementImpression (#248)

5.2.36 Version v0.10.1

Date October 1, 2020

v0.10.0 caused a very long migration which we resolved in v0.10.1

- @ericholscher: Don't index *ad_type* on the AdBase (#246)

5.2.37 Version v0.10.0

Date October 1, 2020

The major change in this release was to allow publishers to individually track the performance of ads on certain pages/sections separately by adding an `id` attribute to the ad `<div>`. Behind the scenes, there was a rework in how we track when an ad is offered and viewed but those are not user facing.

- @ericholscher: Store placements and keywords and add reporting (#239)

5.2.38 Version v0.9.1

Date September 22, 2020

- @ericholscher: Update precommit deps to match latest (#240)
- @ericholscher: Improve automation around payouts (#237)
- @ericholscher: Add a management command to add a publisher (#236)
- @ericholscher: Allow sorting All Publishers list by revenue (#235)

5.2.39 Version v0.9.0

Date August 25, 2020

The largest change in this release was to store publisher payout settings and allow publishers to connect via Stripe to attach a bank account for payouts.

- @davidfischer: Turn down the rate limiting logging (#232)
- @davidfischer: Use Django2 style URLs everywhere (#231)
- @davidfischer: Refactor publisher tests (#230)
- @davidfischer: Store publisher payout settings (#229)
- @davidfischer: Refactor flight metadata view (#180)
- @davidfischer: Store publisher payout settings (#177)

5.2.40 Version v0.8.0

Date August 18, 2020

The two changes in this release were to add branding to the ad server which is only enabled in production and shouldn't be used by third-parties and to add the ability to group publishers into groups for targeting purposes.

- @davidfischer: Group publishers (#227)
- @davidfischer: Add EthicalAds branding to the adserver (#226)

5.2.41 Version v0.7.0

Date August 5, 2020

The main change in this version is to add a database model for storing publisher payouts and making that data visible to publishers.

- @davidfischer: Change some log levels around impressions blocking (#224)
- @davidfischer: Save publisher payouts (#223)
- @ericholscher: Make Publisher defaults line up with Ad Network defaults (#222)

5.2.42 Version v0.6.0

Date August 3, 2020

This release had a few minor changes but the larger changes involved adding the ability to rate limit ad views and an admin action for processing advertiser refunds/credits.

- @davidfischer: Admin action for processing refunds (#220)
- @davidfischer: Default ad creation to live (#218)
- @davidfischer: Ignore all known users (#217)
- @davidfischer: Update the all publishers report to show our revenue (#216)
- @davidfischer: Rate limit ad viewing (#212)

5.2.43 Version v0.5.0

Date July 29, 2020

- @davidfischer: Evaluate IP based proxy detection solution (#213)

5.2.44 Version v0.4.2

Date July 29, 2020

- @davidfischer: IP Geolocation and Proxy detection improvements (#210)

5.2.45 Version v0.4.1

Date July 28, 2020

This was purely a bugfix release.

- @davidfischer: Fix a bug around clicking an add after 4 hours (#208)

5.2.46 Version v0.4.0

Date July 28, 2020

There's two main changes in this release related to blocking referrers and UAs: Firstly, the setting `ADSERVER_BLACKLISTED_USER_AGENTS` became `ADSERVER_BLOCKLISTED_USER_AGENTS`. Also, we added a setting `ADSERVER_BLOCKLISTED_REFERRERS`.

- @davidfischer: Send warnings to Sentry (#206)
- @davidfischer: Allow blocking referrers for ad impressions with a setting (#205)

5.2.47 Version v0.3.2

Date July 28, 2020

This is a minor release that just changes some cookie settings to have shorter CSRF cookies and send them in fewer contexts. It also allows the link for an advertiser's ad to contain variables.

- @davidfischer: Allow simple variables in Advertisement.link (#201)

- @davidfischer: CSRF Cookie tweaks (#196)

5.2.48 Version v0.3.1

Date July 23, 2020

This is mostly a bugfix release and contains some slight operations tweaks. The biggest change is to allow mobile targeting or excluding mobile traffic.

- @davidfischer: Fix a secondary check on geo-targeting (#199)
- @davidfischer: Optimization to choose a flight with live ads (#198)
- @davidfischer: Add a link to the privacy policy (#197)
- @davidfischer: Remove request logging (#193)
- @davidfischer: Allow targeting mobile or non-mobile traffic (#192)
- @dependabot[bot]: Bump lodash from 4.17.15 to 4.17.19 (#190)
- @davidfischer: Flight targeting to include/exclude mobile traffic (#188)

5.2.49 Version v0.3.0

Date July 15, 2020

The major change in this version is the Stripe integration which allows tying advertisers to a Stripe customer ID and the automated creation of invoices (they're created as drafts for now) through the admin interface.

- @ericholscher: Order the Ad admin by created date, not slug (#187)
- @davidfischer: Use Django dev for Intersphinx (#186)
- @davidfischer: Stripe integration (#185)
- @ericholscher: Update docs to explain auth on POST request (#184)

5.3 Installation Guide

If you are looking to install your own instance of the Ethical Ad Server, these docs are for you.

5.4 User Guide

This section of the docs has specific how-to guides to get the most from the Ethical Ad Server.

5.5 Developer Guide

If you are developing the Ethical Ad Server, then these are the docs for you.

/api

Index

A

P