

# ERD of Campaign Management Database

## Table of Contents

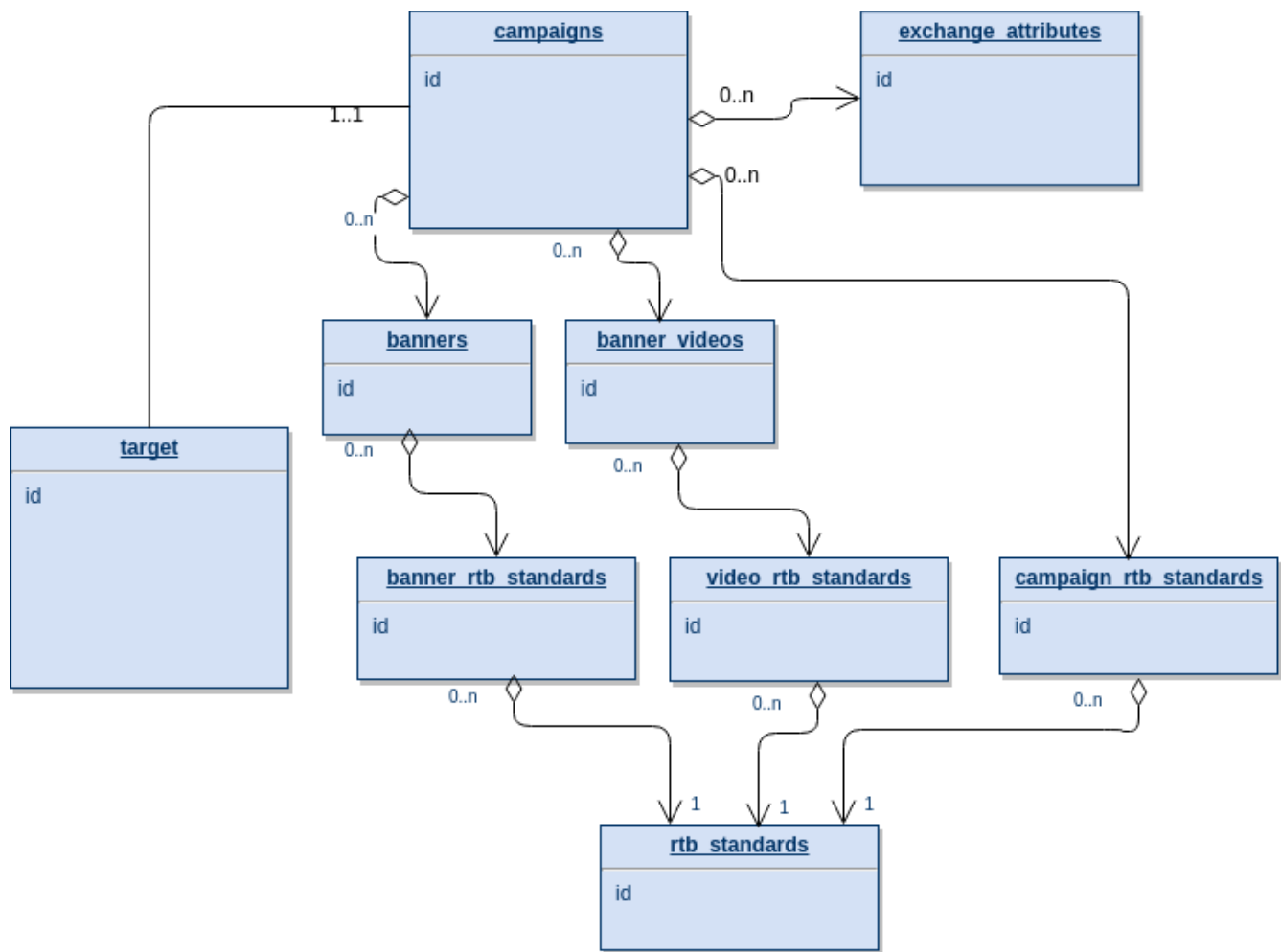
Entity Relationships.....	2
Overview.....	4
Table Definitions.....	5
Target.....	5
Campaigns.....	6
Banners.....	7
Banner_videos.....	9
Exchange_attributes.....	11
Campaigns_rtb_standard.....	11
Banners_rtb_standards.....	11
Banner_videos_rtb_standard.....	12
Rtb_standards.....	12
Linking the Tables Together.....	14
Minimum Viable Banner Ad.....	14
Minimum Viable Video Ad.....	15
Minimum Viable Target.....	16
Minimum Viable Campaign.....	16
Example Geo Rule.....	17
Example Rule with Multiple States.....	18
Further Documentation.....	19
Automatically Updating Campaigns.....	20
Basic Operations of Crosstalk API.....	21
Add Campaigns.....	21
Delete Campaigns.....	21
Update Campaign.....	21
Refresh.....	22

# Entity Relationships

The RTB4FREE database contains a number of tables that describe the campaigns and creatives used by the RTB4FREE system. Below is a list of those tables and a brief description of each of them:

1. campaigns – Contains the specification of the campaigns; a campaign is the root of the system. A campaign has one target, zero or one exchange\_attributes, at least one of (zero or more banners or zero or more banners\_video) and zero or one exchange\_attributes.
2. targets – Contains standard targeting definitions for a campaign. There is 1 target per campaign. Defines things like country, location, device make, device model.
3. banners – Describes banner creatives.
4. banner\_videos – Describes video creatives.
5. exchange\_attributes – Describes specialized exchange attributes. Used to describe exchange specific attributes that are required by the specific exchange for a campaign. These attributes are used by exchanges such as Stroer, Appnexus, Bidswitch and Adx.
6. rtb\_standards – Defines global specification of constraints on RTB objects. Used to describe constraints beyond those specified in the target table.
7. campaign\_rtb\_standards – Defines rtb\_standards applied to campaigns. It has 2 fields, one is the campaign id and then, the rtb\_standard applied to that campaign.
8. banner\_rtb\_standards – Defines rtb\_standards applied to banner creatives. It has 2 fields, one is the banners id, and then the rtb\_standard applied to it.
9. video\_rtb\_standards – Defines rtb\_standards applied to video creatives Defines rtb\_standards applied to video creatives. It has 2 fields, one is the banners\_video id, and then the rtb\_standard applied to it.

The key on every field is “id” a unique serial int. The following is the entity relationships in the database:



Relationship of Tables in Campaign Management.

The Crosstalk Docker compose file will denote where the SQL server, table and database is for these tables. Example:

JDBC: "jdbc:mysql://server-1/rtb4free?user=ben&password=MyPassWord

In this case, the database is located on server-1, the db base is rtb4free.

## Overview

To create a viable campaign you will have one row for a campaign in the “campaigns” table. There will be a single row in the “targets” table for each campaign. The target\_id field in the campaigns table is the value of the id in the targets table.

In either, or both “banners” table and “banner\_videos” table there will be at least one row in the combination of both of these tables of a campaign\_id field equal to the id in the campaigns table.

In the “campaign\_rtb\_standards” table there will be zero or more rows with the campaign\_id field equal to the id field in the “campaigns” table. The rtb\_standard\_id field is equal to the id field in the “rtb\_standards” table. This is how custom constraints are attached at the campaign level.

In the “banner\_rtb\_standards” table there will be zero or more rows with the campaign\_id field equal to the id field in the “banners” table. The rtb\_standard\_id field is equal to the id field in the “rtb\_standards” table. This is how custom constraints are attached at the banner creative level.

In the “video\_rtb\_standards” table there will be zero or more rows with the video\_id field equal to the id field in the “banners\_video” table. The rtb\_standards\_id field is equal to the id field in the “rtb\_standards” table. This is how custom constraints are attached at the video creative level.

When in doubt, look at a campaign created with the RTB4FREE campaign manager to see what the corresponding values look like.

# Table Definitions

## Target

The targets table contains fairly common constraints applied generally to campaign. For example, to create a campaign for a single geographic region using lat/lon/range, or whitelisting domains, or defining what country you want to apply a campaign to. Every campaign has a target; however, it is possible to recreate all the constraint checking found here using the rtb\_standards table.

```
CREATE TABLE targets (  
  id serial,  
  activate_time timestamp DEFAULT NULL,  
  expire_time timestamp DEFAULT NULL,  
  list_of_domains TEXT DEFAULT NULL,  
  domain_targetting text DEFAULT NULL,  
  geo_latitude decimal DEFAULT NULL,  
  geo_longitude decimal DEFAULT NULL,  
  geo_range decimal DEFAULT NULL,  
  country text DEFAULT NULL,  
  geo_region text DEFAULT NULL,  
  carrier text DEFAULT NULL,  
  os text DEFAULT NULL,  
  make text DEFAULT NULL,  
  model text DEFAULT NULL,  
  devicetype text DEFAULT NULL,  
  IAB_category text DEFAULT NULL,  
  IAB_category_bklist text DEFAULT NULL,  
  created_at timestamp DEFAULT NULL,  
  updated_at timestamp DEFAULT NULL,  
  name text DEFAULT NULL,  
  PRIMARY KEY (id)  
);
```

The following is a description of all the fields:

1. id - The unique key, auto generated.
2. activate\_time - unused.
3. expire\_time unused.
4. list\_of\_domains - A list of domains to target or avoid.
5. domain\_targetting - Set to "blacklist" to avoid domains in list\_of\_domains, or "whitelist" to target these domains.
6. geo\_latitude double - Latitude of the target.
7. geo\_longitude double - Longitude of the target.
8. geo\_range double - Range from the lat/lon in meters.
9. country text - ISO3 comma separated list of countries to target.
10. geo\_region - Unused.
11. carrier - Comma separated list of carriers to target.
12. Os - Name of the operating system/version of the mobile handset to target.
13. Make - Make of the handset.
14. model text - Model of the handset.
15. devicetype - Devicetype comma separated list of types as set forth in RTB Specification for device.devicetype.

16. IAB\_category - IAB categories to target.
17. IAB\_category\_bklist - IAB categories to avoid.
18. created\_ - When the target was created.
19. updated\_at When the target was updated.
20. name - Name of the target.

## Campaigns

This table defines the “campaign” where multiple creatives can be gathered together under the same budget and constraints.

```
CREATE TABLE campaigns (
  id SERIAL,
  activate_time timestamp DEFAULT NULL,
  expire_time timestamp DEFAULT NULL,
  cost decimal(15,6) DEFAULT NULL,
  ad_domain varchar(1024) DEFAULT NULL,
  clicks int DEFAULT NULL,
  pixels int DEFAULT NULL,
  wins int DEFAULT NULL,
  bids int DEFAULT NULL,
  name varchar(1024) DEFAULT NULL,
  status varchar(1024) DEFAULT NULL,
  conversion_type varchar(1024) DEFAULT NULL,
  budget_limit_daily decimal(15,6) DEFAULT NULL,
  budget_limit_hourly decimal(15,6) DEFAULT NULL,
  total_budget decimal(15,6) DEFAULT NULL,
  bid decimal(15,6) DEFAULT NULL,
  shard text,
  forensiq text,
  daily_cost decimal(15,6) DEFAULT NULL,
  updated_at timestamp DEFAULT NULL,
  deleted_at timestamp DEFAULT NULL,
  created_at timestamp DEFAULT NULL,
  hourly_cost decimal(15,6) DEFAULT NULL,
  exchanges varchar(255) DEFAULT NULL,
  regions varchar(255) DEFAULT NULL,
  target_id int DEFAULT NULL,
  PRIMARY KEY (id)
);
```

The following is a definition of all the fields:

1. id – The unique key id of the campaign. Auto generated.
2. activate\_time – The time to start this campaign. Required.
3. expire\_time – The time to end this campaign. Required.
4. cost decimal(15,6). Read only field, used by aggregations.
5. ad\_domain varchar(1024) - The ad domain of this campaign. Required.
6. clicks – Read only field used by aggregations.
7. pixels - Read onlt field used by aggregations.
8. wins - Read onlt field used by aggregations.
9. bids - Read onlt field used by aggregations.

10. name – Descriptive name of this campaign. Required.
11. status – Description of whether this is loadable in the bidder. Required, set to “running” to make it loadable. Any other value will make the bidder ignore this campaign.
12. conversion\_type. Unused.
13. budget\_limit\_daily – The daily budget. US Dollars. If set to null, unlimited budget is used.
14. budget\_limit\_hourly decimal – Hourly budget in US Dollars, if set to null, unlimited daily budget is used.
15. total\_budget – Total budget in US Dollars, if set to null, unlimited total budget is used.
16. bid – Unused.
17. shard – Unused.
18. forensiq – Set to “Y” to turn on anti-fraud (if defined in the bidder, otherwise no effect).
19. daily\_cost – Read only field.
20. updated\_at - Set to the timestamp of when this was last updated. Optional.
21. deleted\_at - Set to the timestamp of when you set status to “deleted”. Optional.
22. created\_at - Set to the timestamp of when you created the record. Optional.
23. hourly\_cost – Read only, used by aggregations.
24. exchanges varchar(255) DEFAULT NULL,
25. regions – Set to the region tag (e.g. “US”) that corresponds to the REGION environment variable in crosstalk.
26. target\_id – The ID of the row in the targets table that defines the target for this campaign. Required. A campaign with no targets\_id set is not viable.

## Banners

This table defines a banner ad.

```
CREATE TABLE banners (
  id SERIAL,
  campaign_id int DEFAULT NULL,
  interval_start timestamp NOT NULL,
  interval_end timestamp DEFAULT NULL,
  total_basket_value decimal DEFAULT NULL,
  width int DEFAULT NULL,
  height int DEFAULT NULL,
  bid_ecpm decimal(15,6) DEFAULT NULL,
  total_cost decimal(15,6) DEFAULT NULL,
  contenttype varchar(1024) DEFAULT NULL,
  iurl varchar(1024) DEFAULT NULL,
  htmltemplate text,
  bids int DEFAULT NULL,
  clicks int DEFAULT NULL,
  pixels int DEFAULT NULL,
  wins int DEFAULT NULL,
  daily_budget decimal(15,6) DEFAULT NULL,
  hourly_budget decimal(15,6) DEFAULT NULL,
  daily_cost decimal(15,6) DEFAULT NULL,
  target_id int DEFAULT NULL,
  created_at timestamp NOT NULL,
  updated_at timestamp NOT NULL,
  name varchar(255) DEFAULT NULL,
```

```

frequency_spec varchar(255) DEFAULT NULL,
frequency_expire int DEFAULT NULL,
frequency_count int DEFAULT NULL,
hourly_cost decimal(15,6) DEFAULT NULL,
deals varchar(255) DEFAULT NULL,
width_range varchar(255) DEFAULT NULL,
height_range varchar(255) DEFAULT NULL,
width_height_list varchar(255) DEFAULT NULL,
PRIMARY KEY (id)
);

```

The following is a description of all the fields.

1. Id - The unique key id of the campaign. Auto generated.
2. campaign\_id - The id of the campaign this creative belongs to. Required.
3. interval\_start – The time this creative becomes active. Required.
4. interval\_end – The time this creative becomes inactive. Required.
5. total\_basket\_value – The total budget for this creative. If not specified, bids unconstrained by budget.
6. width - The width of this creative. In the alternative, you can specify a list or a range, see below.
7. Height - The height of this creative. In the alternative, you can specify a list or a range, see below.
8. bid\_ecpm – Bid price. Required.
9. total\_cost (15,6) DEFAULT NULL,
10. contenttype (1024) DEFAULT NULL. Set to 'html' or 'javascript', depending on what you will put in htmltemplate. REQUIRED.
11. iurl – Image url of the creative. Required.
12. htmltemplate – The ad markup. Required. Use html or a <script> defintion.
13. Bids - Read only, used by aggregations.
14. Clicks - Read only, used by aggregations.
15. Pixels - Read only, used by aggregations.
16. Wins - Read only, used by aggregations.
17. daily\_budget – Daily budget in US Dollars, if null, no budget set.
18. hourly\_budget – Hourly budget in US Dollars, if null, no budget set.
19. daily\_cost decimal – Read only used by aggregations.
20. target\_id – Unused.
21. created\_at – Required - timestamp of when created.
22. updated\_at – Timestamp of when it was updated. Required.
23. name – Descriptive name of this creative. Required.
24. frequency\_spec – The specification for the RTB object to put the frequency cap on. Example: device.ip
25. frequency\_expire - The number of seconds the frequency cap will expire on for the value of frequency\_spec.
26. frequency\_count - The number of times the same value of frequency\_spec the creative is available to view within frequency\_expire seconds.
27. hourly\_cost – Read only, used by aggregations.
28. deals – An encoded string for a deal specification of the form: deal,deal,deal... Where deal is of the form: id:price. For example: deal1:.03,deal2:.04,deal3:.05



29. width\_range - Specifies a width range. Like 600-900. If null, use width and height or width\_height\_list.
30. height\_range - Specifies a height range. Like 300-500. If null use width and height or width\_height\_list.
31. width\_height\_list - If width/vast\_height+width\_range/height\_range are null, you can specify a list of width and height. Example: 300x200,400x900,200x600.

## Banner\_videos

This table describes a video ad.

Below is the POSTGRES (ANSI) SQL definition of the Banner\_videos table.

```
CREATE TABLE banner_videos (
  id SERIAL,
  campaign_id int DEFAULT NULL,
  interval_start timestamp DEFAULT NULL,
  interval_end timestamp DEFAULT NULL,
  total_basket_value decimal(15,6) DEFAULT NULL,
  total_budget decimal(15,6) DEFAULT NULL,
  vast_video_width int DEFAULT NULL,
  vast_video_height int DEFAULT NULL,
  vast_video_data TEXT DEFAULT NULL,
  bid_ecpm decimal DEFAULT NULL,
  vast_video_linearity int DEFAULT NULL,
  vast_video_duration int DEFAULT NULL,
  vast_video_type text,
  vast_video_outgoing_file text,
  bids int DEFAULT NULL,
  clicks int DEFAULT NULL,
  pixels int DEFAULT NULL,
  wins int DEFAULT NULL,
  total_cost decimal DEFAULT '0.000000',
  daily_cost decimal DEFAULT NULL,
  daily_budget decimal DEFAULT NULL,
  frequency_spec text,
  frequency_expire int DEFAULT NULL,
  frequency_count int DEFAULT NULL,
  created_at timestamp NOT NULL,
  updated_at timestamp NOT NULL,
  hourly_budget decimal DEFAULT NULL,
  name varchar(256) DEFAULT NULL,
  target_id int DEFAULT NULL,
  hourly_cost decimal DEFAULT NULL,
  bitrate int DEFAULT NULL,
  mime_type varchar(255) DEFAULT NULL,
  deals varchar(255) DEFAULT NULL,
  width_range varchar(255) DEFAULT NULL,
  height_range varchar(255) DEFAULT NULL,
  width_height_list varchar(255) DEFAULT NULL,
  PRIMARY KEY (id)
);
```

Below is a description of all the fields:

1. id SERIAL – Auto generated id of this video.
2. campaign\_id – The campaign this video belongs to. Required.
3. interval\_start – The timestamp of when the creative is active. Required.
4. interval\_end – The timestamp of when the creative is completed. Required.
5. total\_basket\_value – The total budget for this creative. If null, will bid unconstrained.
6. total\_budget - Unused
7. vast\_video\_width - Width of the video. If 0, use width\_range and height\_range instead.
8. vast\_video\_height - Height of the video. If 0, use width\_range and height\_range instead.
9. vast\_video\_data – The XML VAST tag.
10. bid\_ecpm – The price of the creative bid. Required.
11. vast\_video\_linearity – The video linearity. Required. See RTB Spec.
12. vast\_video\_duration - The video duration in seconds. Required.
13. vast\_video\_type – Unused.
14. vast\_video\_outgoing\_file. Unused.
15. bids – Read only, used by aggregations.
16. clicks - Read only, used by aggregations.
17. pixels - Read only, used by aggregations.
18. wins - Read only, used by aggregations.
19. total\_cost - Read only, used by aggregations.
20. daily\_cost - Read only, used by aggregations.
21. daily\_budget – When set, in US Dollars, will determine the daily budget. If not set, bids unconstrained.
22. frequency\_spec – The specification for the RTB object to put the frequency cap on. Example: device.ip
23. frequency\_expire - The number of seconds the frequency cap will expire on for the value of frequency\_spec.
24. frequency\_count - The number of times the same value of frequency\_spec the creative is available to view within frequency\_expire seconds.
25. updated\_at - Set to the timestamp of when you update this record.
26. created\_at - Set to the timestamp of when you created the record.
27. hourly\_budget - When set, in US Dollars,, will determine the hourly budget. If not set, bids unconstrained.
28. name – Descriptive name of the creative.
29. target\_id – Not used.
30. hourly\_cost – Read only, used by aggregations.
31. bitrate – Bitrate used for the video. Required.
32. mime\_type – Defines the type of the creative. Use valid HTTP mime types ONLY.
33. deal - An encoded string for a deal specification of the form: deal,deal,deal... Where deal is of the form: id:price. For example: deal1:.03,deal2:.04,deal3:.05
34. width\_range = Specifies a width range. Like 600-900. If null, use vast\_video\_width and vast\_video\_height.
35. height\_range – Specifies a height range. Like 300-500. If null use vast\_video\_width and vast\_video\_height.
36. width\_height\_list – If vast\_video\_width/vast\_video\_height+width\_range/height\_range are null, you can specify a list of width and height. Example: 300x200,400x900,200x600.

## Exchange\_attributes

Some exchanges, like Stroer, Adx, and Appnexus require extra fields be added to the bid response. These are attached via the Exchange\_attributes table.

```
CREATE TABLE exchange_attributes (  
  id SERIAL,  
  banner_id int DEFAULT NULL,  
  banner_video_id int DEFAULT NULL,  
  name varchar(255) DEFAULT NULL,  
  value varchar(255) DEFAULT NULL,  
  created_at timestamp DEFAULT NULL,  
  updated_at timestamp DEFAULT NULL,  
  exchange varchar(255) DEFAULT NULL,  
  PRIMARY KEY (id)  
);
```

The following is a description of all the fields for Exchange\_attributes:

1. id – Automatically generated id for this row.
2. banner\_id – The creative banner that this exchange attribute is assigned to.
3. banner\_video\_id – The creative video id that this exchange attribute is assigned to.
4. name – Name of the exchange specific attribute.
5. Value – The value of the attribute by this name.
6. created\_at – The timestamp when this was created.
7. updated\_at timestamp – The time this row was last updated.
8. exchange – The name of the exchange this name/value pair is associated with.

## Campaigns\_rtb\_standard

This table links campaigns to rules defined in rtb\_standards table.

```
CREATE TABLE campaigns_rtb_standards (  
  campaign_id int,  
  rtb_standard_id int DEFAULT NULL  
);
```

Below is a description of the fields.

1. campaign\_id – The id in the campaigns table this row belongs to.
2. rtb\_standard\_id – The id in rtb\_standards this entry is linked to.

## Banners\_rtb\_standards

This table links banner ads to rules defined in the rtb\_standards table.

```
CREATE TABLE banners_rtb_standards (  
  banner_id int,  
  rtb_standard_id int DEFAULT NULL  
);
```

Below is a description of the fields.

1. banner\_id – The id in the banners table this row belongs to.

2. `rtb_standard_id` – The id in `rtb_standards` this entry is linked to.

## Banner\_videos\_rtb\_standard

This table links video ads to rules defined in the `rtb_standards` table.

```
CREATE TABLE banner_videos_rtb_standards (  
  banner_video_id int,  
  rtb_standard_id int DEFAULT NULL  
);
```

Below is a description of the fields.

1. `campaign_id` – The id in the `banners` table this row belongs to.
2. `rtb_standard_id` – The id in `rtb_standards` this entry is linked to.

## Rtb\_standards

This table provides a mechanism to create custom rules for building constraints applied to RTB objects in bid request.

```
CREATE TABLE rtb_standards (  
  id SERIAL,  
  rtbspecification varchar(1024) DEFAULT NULL,  
  operator varchar(1024) DEFAULT NULL,  
  operand varchar(1024) DEFAULT NULL,  
  operand_type varchar(16) DEFAULT NULL,  
  operand_ordinal varchar(16) DEFAULT NULL,  
  rtb_required int DEFAULT NULL,  
  name varchar(255) DEFAULT NULL,  
  description varchar(255) DEFAULT NULL,  
  created_at timestamp DEFAULT NULL,  
  updated_at timestamp DEFAULT NULL,  
  operand_list_id int DEFAULT NULL,  
  PRIMARY KEY (id)  
);
```

It would be useful to read this section of the RTB4FREE website to understand the operations of the `rtb_standards` table: <http://www.rtb4free.com/rules.html>

Below is a description of all the elements in the table:

1. `id SERIAL` – Auto generated ID of this table.
2. `rtbspecification` – Defines the value of the RTB object you are creating a constraint for. Example: “device.os”. See the latest RTB specification for object specifications.
3. `Operator` – Defines the operation to apply to the operand and the value in the `rtbspecification` field. Example “EQUALS”. Supported operators can be found here: <http://www.rtb4free.com/rules.html#operators-section>
4. `Operand` – The string representation of the operand. For example “4” or “hello”.
5. `operand_type` The string representation of the operand’s type. For example “S” “D” or “I” (String or Decimal, or Integer)
6. `operand_ordinal` – Defined as either “L” or “S” for List or Scalar.

7. `rtb_required` int – Determines if the RTBSPECIFICATION is not found, whether to return ‘true’ or ‘false’ if the object is not found. If the field is 1, and the `rtbspecification` is not found, then returns false – meaning NOBID. If set to 0 and `rtbspecification` is not found, this constraint will return true (Will bid if all other constraints and target returns true).
8. `name` – Descriptive name of the constraint.
9. `description` – Description of the operation.
10. `created_at` – Timestamp of when created. Required.
11. `updated_at` timestamp – Timestamp of when updated.
12. `operand_list_id` – Unused.

## Linking the Tables Together

A Campaign has one entry in the “campaigns” table. So this has to be set up with some minimum information.

Then a Target is set up that the campaign will use. The target id field is placed into the “campaign” table at the field. target\_id. This links a target to the campaign. A campaign must have 1 target.

Creatives are located in “banners” table and “banner\_videos” tables. Define your creatives and link the creatives into campaigns table by placing the Campaign id field into the banners.campaign\_id field or the banner\_videos.campaign\_id field, as appropriate.

Now, to define a rule you place the rule definition into the “rtb\_standards” table.

To attach a rule to a campaign, create an entry in the campaigns\_rtb\_standards table and set the campaign\_id field to the id of the campaign in its campaign table. And set the rtb\_standard\_id field to the id of the rule in rtb\_standards.

Likewise, to attach a rule to a creative, create an entry in the banner\_videos\_rtb\_standards table if it is a video, or in the banners\_rtb\_standards table if the creative is a banner ad. Set the banner\_id field in the banners\_rtb\_standards table (if a banner) or set the banner\_video\_id field in banner\_videos\_rtb\_standards table (if a video). Then set the rtb\_standard\_id field in these two aforementioned tables with the id field of the rule in the rtb\_standards table.

In the next sections we show a minimal setup for a campaign with a video and a banner ad. Then we will create a target. Next we create the campaign definition. We build it up, in reverse order.

The table “exchange\_attributes” is used for attaching special attributes that certain exchanges require on the creatives (Most RTB exchanges don’t require this table). But here is a list of exchanges that do require fields.

1. Google Adx/DoubleClick
2. Google Adx/OpenRTB
3. Biddwitch
4. Appnexus

If your campaign is using one of these exchanges, you must define an entry in exchange\_attributes for it. You set the “name” field to the name of the attribute. Then “value” field is set to the value. Then put the banner\_id or banner\_video\_id for the creative that requires that id.

## Minimum Viable Banner Ad

A banner ad has at a minimum the following fields (note start time is now, and end time is 5 days from now (MySQL syntax):

```

insert into banners(interval_start,
                    interval_end,
                    width,
                    height,
                    bid_ecpm,
                    created_at,
                    updated_at,
                    htmltemplate,
                    contenttype) values(NOW(),
                    NOW() + '3 days',
                    3.5,
                    320,
                    50,
                    NOW(),
                    NOW(),
                    '<img src=' 'http://localhost/www.test.jpeg' '>',
                    'image/jpeg');

```

For example purposes we presume the id of this inserted banner is 20.

## Minimum Viable Video Ad

For example purposes we presume the id of this inserted video ad is 30.

```

insert into banner_videos (interval_start,
                            interval_end,
                            vast_video_width,
                            vast_video_height,
                            vast_video_linerarity,
                            vast_video_duration,
                            vast_video_outgoing_file,
                            bitrate,
                            mime_type,
                            created_at,

```

```
updated_at) VALUES(NOW(),
NOW(),
600,
400,
2,
120,
'your-vast-tag-goes-here',
64000,
'video/mpeg2',
NOW(),
NOW());
```

## Minimum Viable Target

Targets provide some basic constraint checking on a lot of RTB fields, but if you are making your own campaign manager, you only need to provide a minimal target. Here is one:

```
insert into targets(
country,
created_at,
updated_at,
name) values('USA',
NOW(),
NOW(),
'Sample target for USA');
```

For example purposes we presume the id assigned to this target is 123.

## Minimum Viable Campaign

Now that we have a target, we can set up the campaign. For example sake, the target id will be 123.

```
Insert into campaigns (
activate_time,
```



```
expire_time,  
regions,  
target_id,  
status,  
created_at,  
updated_at) values(NOW(),  
NOW() + '3 days',  
'USA',  
123,  
'runnable',  
NOW(),  
NOW());
```

Now we update the banners, banners\_video with the id of this newly created campaign. For example sake, the id of the campaign is 1, and the banners id of the creative is 20 and the banner\_videos id of the vast creative is 30.

```
update banners set campaign_id = 1 where id=20;
```

```
update banner_videos set campaign_id = 1 where id=30.
```

## Example Geo Rule

In this example we will create a rule that targets 5 zipcodes with a range of 10Km, such that any bid within 10Km of any of the listed zipcodes will be selected:

First we need to create a rule, we will call it '5 States':

```
insert into rtb_standards (created_at,  
updated_at,  
rtbspecification,  
operator,  
operand,  
operand_type,  
operand_ordinal,  
rtb_required,  
name,  
description) VALUES(now(),  
now(),  
'requests.device.geo',  
'INRANGE',  
'ZIPCODES 90501,90502,90503,90504,90505,10000',
```

```
'S',  
'L',  
1,  
'LAT/LON',  
'Demonstates 5 zipcodes and a range of 10km.');
```

While the ZIPCODES themselves are decimal, the type is set to String. It will be treated as a Scalar, so that the single line of 'ZIPCODES 90501...' is treated together.

For example sake, let's presume the id was set to 7. Now, lets add this rule to the banners creative whose id = 1.

```
insert into banners_rtb_specification set(banner_id, rtb_standard_id)  
VALUES(1,7);
```

## Example Rule with Multiple States

In this example we will create rule that covers 3 states: New York, California and Nevada. Then we will attach it to the video creative.

First we create the rule.

```
insert into rtb_standards (created_at,  
updated_at,  
rtbspecification,  
operator,  
operand,  
operand_type,  
operand_ordinal,  
rtb_required,  
name,  
description) VALUES(now(),  
now(),  
'requests.device.geo.region',  
'MEMBER',
```

```
'NY,CA,NV',  
'S',  
'L',  
1,  
'3 states',  
'Demonstrates a constraint of 3 states');
```

In this case, we treat NY,CA,NV as a list of strings. For example sake let's say this returned an ID of 2.  
Now to link this rule to the video we created earlier (in banner\_videos table) with id of 30.

```
insert into banner_videos_rtb_standards (banner_video_id,rtb_standard_id) values(30,2);
```

## Further Documentation

Campaign Manager - <http://rtb4free.com/campaigns-docker.html>

Operators – <http://www.rtb4free.com/rules.html#operators-section>

## Automatically Updating Campaigns

The Campaigns, if updated and are runnable, will be added to the Bidder within 1 minute of updating the campaign updated\_at field.

If you modify anything that affects the campaign (a rule, creative or target), the Bidders will not be reloaded until you signal the system that the campaign is ready to update. Using SQL, for example we decide to edit a banner that is part of campaign id 134. To signal that the campaign should be reloaded, we update the row as such:

```
# update campaigns set updated_at = now() where id = 134;
```

Within 1 minute, if campaign 134 is otherwise runnable, then it will be loaded into the system.

## Basic Operations of Crosstalk API

In general, you make whatever SQL changes you need, and then within one minute the Crosstalk system will automatically reload the campaign (if it is otherwise runnable).

However, if you want to immediately effect the load, then send an HTTP POST to the Crosstalk system on port 8100. Please read this section on the crosstalk-api to get an idea of how this all works:

[http://www.rtb4free.com/doc\\_core\\_crosstalk-api.html](http://www.rtb4free.com/doc_core_crosstalk-api.html)

## Add Campaigns

This does not add a campaign to the database – you are responsible for that. This command tells Crosstalk to add the campaign to the bidders.

Basically you define your campaign, creatives, target, and rtb\_standards for your campaign, as described in the previous sections.

After you create the entries in all the tables, make sure your campaigns.status is set to “running”

Then, you issue an HTTP post to the Crosstalk system on port 8100 like this:

```
{ "username": "xxx", "password": "yyy", "campaign": "23" "type": "AddCampaign#" }
```

This will cause Crosstalk to add this campaign id 23 to the bidders. Note, within 1 minute of any updates to campaigns table (updated\_at) Crosstalk will automatically add the campaign.

This will cause an attempt to reload the campaign, even if there were no changes to the campaign, or even if the campaign’s updated\_at field was not updated.

This command will not load a campaign that is not runnable (status is not set to ‘runnable’ or the campaign has expired, or the budget is expired).

## Delete Campaigns

This does not delete the campaign from the database – you are responsible for that. This command tells Crosstalk to delete a campaign from the bidders. Note, setting the campaign status field to anything but running, or, if the budget is exceeded on a campaign or its creatives, Crosstalk will automatically delete the campaign from the bidders. To delete the campaign issue the HTTP post to Crosstalk on port 8100 like this:

```
{ "username": "xxx", "password": "yyy", "campaign": "10", "type": "DeleteCampaign#" }
```

## Update Campaign

If you update a creative (banners or banner\_videos table, or targets and the rtb\_specifications) Crosstalk will not automatically update the campaign unless you update the updated\_at field in the campaigns field. You can manually tell Crosstalk the campaign was updated by issuing either another AddCampaign# HTTP POST, or send it an update on port 8100 of Crosstalk like this:

```
{ "username:" "xxx", "password": "yyy", "type": "Update#", "campaign":"14"}
```

## Refresh

The Refresh command will cause Crosstalk to scan the database and load ALL loadable campaigns. Issue this HTTP POST on port 8100 of Crosstalk to initiate the refresh:

```
{ "username:" "xxx", "password": "yyy", "type": "Refresh#"}
```