RTB4FREE Documentation

Release 1.0.0

Ben Faul, Peter Loh, Scott Switzer

Jul 10, 2019

Getting Started

Welcome to the RTB4FREE documentation hub. Here you can find the online manual and living documentation repository for RTB4FREE, the leading digital advertising platform for businesses who want to reach their audience on the open web.

Living manual Documentation is updated in the same manner as code - and a build is made for each version of RTB4FREE. You can easily view current and previous versions of documentation from this website.

International first While the english language version of the documentation is the reference version, other languages are available as well. Our documentation is hosted in Github along with all other parts of our codebase - so you can update the doc in your favorite language and submit a pull request to keep it current.

Free and open source The documentation is built like our software - community driven, open source, free for all to view.

CHAPTER 1

Getting Started With RTB4FREE

If you are new to RTB4FREE, the first thing you will probably want to do is demo the platform to see if it is right for your needs. While RTB4FREE is a fully functional platform capable of handling advertising on a massive scale, it is also simple to install for demo purposes.

Follow the instructions below to quickly get started:

- **Quick installation**: *With Docker* | *With Amazon AWS*
- **Connect bidstream supply**: *Demo bidstream* | *Autoconnect real supply*
- **Create your first campaign**: *Make a campaign*

1.1 Getting Started with Docker

The simplest way to get started with RTB4Free is to use Docker Compose to install all components onto a single system, so that you can experience the functionality and quickly see if RTB4FREE is right for your business.

Note: All services are installed on a single machine in this installation method. This is not recommended for production use - only for demonstration purposes.

This quickstart process performs the following tasks:

- Starts a complete RTB system, along with a campaign manager and reporting system.
- A demo campaign configuration is loaded and can be edited in the campaign manager.
- An SSP exchange simulator is started, sending bid requests to the system. This shows a working RTB application processing real bids.
- View how data flows through the system in real-time with the deployed reporting system.

1.1.1 Quick start video

This screencast will help you get started or you can *read our guide below*.

1.1.2 Quick start

This quick start process will deploy a Docker swarm application. You must have Docker Compose installed on your system before you start.

Create the docker compose file. Edit a new file called docker-compose.yml, then add the following contents:

```
#
# Contains the full rtb4free stack (zookeeper, kafka, zerospike, bidder, crosstalk,␣
↪db, web, elastic, logstash, simulator)
#
version: '3.5'

services:

  zookeeper:
    image: zookeeper:3.5
    ports:
      - '2181:2181'
    networks:
#       - r4f_control
      - r4f_net

  kafka:
    image: wurstmeister/kafka
    environment:
      KAFKA_ADVERTISED_HOST_NAME: kafka
      KAFKA_AUTO_CREATE_TOPICS_ENABLE: 'false'
      KAFKA_CREATE_TOPICS: bids:1:1,wins:1:1,requests:1:1,clicks:1:1,pixels:1:1,
↪videoevents:1:1,postbackevents:1:1,status:1:1,reasons:1:1,logs:1:1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
    ports:
      - '9092:9092'
    depends_on:
      - zookeeper
    networks:
      # - r4f_kafka
      # - r4f_control
      - r4f_net
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock

  zerospike:
    image: rtb4free/zerospike
    environment:
      BROKERLIST: "kafka:9092"
    ports:
      - "6000:6000"
      - "6001:6001"
      - "6002:6002"
      - "7001:7001"
    depends_on:
      - kafka
```

(continues on next page)

(continued from previous page)

```
    networks:
      # - r4f_kafka
      # - r4f_control
      - r4f_net
    command: bash -c "sleep 5 && ./wait-for-it.sh kafka:9092 -t 120 && sleep 1; ./
↪zerospike"

  bidder:
    image: rtb4free/bidder
    environment:
      # GDPR mode - set to true if operating inside a region that enforces GDPR
      GDPR_MODE: "false"
      # The Kafka setup - all of the channels which to deliver events
      BROKERLIST: "kafka:9092"
      BIDSCHANNEL: "kafka://[$$BROKERLIST]&topic=bids"
      WINSCHANNEL: "kafka://[$$BROKERLIST]&topic=wins"
      REQUESTSCHANNEL: "kafka://[$$BROKERLIST]&topic=requests"
      CLICKSCHANNEL: "kafka://[$$BROKERLIST]&topic=clicks"
      PIXELSCHANNEL: "kafka://[$$BROKERLIST]&topic=pixels"
      VIDEOEVENTSCHANNEL: "kafka://[$$BROKERLIST]&topic=videoevents"
      POSTBACKEVENTSCHANNEL: "kafka://[$$BROKERLIST]&topic=postbackevents"
      STATUSCHANNEL: "kafka://[$$BROKERLIST]&topic=status"
      REASONSCHANNEL: "kafka://[$$BROKERLIST]&topic=reasons"
      LOGCHANNEL: "kafka://[$$BROKERLIST]&topic=logs"
      # If using Amazon S3 to store data, enter configuration settings below
      S3BUCKET: ""
      S3REGION: ""
      S3SECRETKEY: ""
      S3ACCESSKEY: ""
      AWSACCESSKEY: ""
      AWSSECRETKEY: ""
      AWSREGION: ""
      # Enter SSP information below
      GOOGLE_EKEY: ""
      GOOGLE_IKEY: ""
      OPENX_EKEY: ""
      OPENX_IKEY: ""
      ADX_EKEY:   ""
      ADX_IKEY:   ""
      BIDSWITCH_ID: "1234"
      # Enter information about how to access
      PUBSUB: "zerospike"
      EXTERNAL: "http://localhost:8080"
      ADMINPORT: "0"
      ACCOUNTING: "accountingsystem"
      FREQGOV: "false"
      INDEXPAGE: "/index.html"
    ports:
      - "8080:8080"
      - "8155:8155"
      - "7379:7379"
      - "7000:7000"
    depends_on:
      - kafka
      - crosstalk
      - zerospike
    networks:
```

(continues on next page)

(continued from previous page)

```
      # - r4f_kafka
      # - r4f_control
      # - r4f_rtb
      - r4f_net
    command: bash -c "sleep 5 && ./wait-for-it.sh kafka:9092 -t 120 && ./wait-for-it.
↪sh zerospike:6000 -t 120 && sleep 1; ./rtb4free"

  # Enable if you'd like to use the bidding simulator to test connections
  simulator:
    image: rtb4free/bidder
    environment:
      BIDDER: "bidder:8080"
      WIN:    "10"
      PIXEL:  "95"
      CLICK:  "2"
      SLEEP:  "100"
      SILENT: "-silent"
    networks:
      # - r4f_rtb
      - r4f_net
    command: bash -c "./wait-for-it.sh bidder:8080 -t 120 && sleep 60;  ./load-
↪elastic -host $$BIDDER -win $$WIN -pixel $$PIXEL -click $$CLICK -sleep $$SLEEP"

  crosstalk:
    image: rtb4free/crosstalk
    environment:
      REGION: "US"
      GHOST: "elastic"
      AHOST: "elastic"
      BROKERLIST: "kafka:9092"
      PUBSUB: "zerospike"
      CONTROL: "8100"
      JDBC: "jdbc:mysql://db/rtb4free?user=rtb4free&password=rtb4free"
      PASSWORD: "rtb4free"
    depends_on:
      - kafka
      - zerospike
      - elastic
      - db
    networks:
      # - r4f_kafka
      # - r4f_control
      - r4f_net

  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    ports:
      - '3306:3306'
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=rtb4free
      - MYSQL_DATABASE=rtb4free
      - MYSQL_USER=rtb4free
      - MYSQL_PASSWORD=rtb4free
    healthcheck:
```

(continues on next page)

(continued from previous page)

```
        test: ["CMD-SHELL", "mysqladmin -h 'localhost' -u root -prtb4free ping --
↪silent"]
        interval: 30s
        timeout: 30s
        retries: 3
    networks:
      # - r4f_control
      - r4f_net
    # custom command is necessary to override 'latin' default charset
    command: mysqld --character-set-server=utf8 --collation-server=utf8_general_ci

  web:
    image: rtb4free/campaign-manager
    ports:
      - "3000:3000"
    environment:
      - CUSTOMER_NAME=RTB4FREE
      - DB_HOST=db
      - DB_PORT=3306
      - DB_USERNAME=rtb4free
      - DB_PASSWORD=rtb4free
      - DB_NAME=rtb4free
      - ELASTICSEARCH_ENABLE=true
      - ELASTICSEARCH_HOST=elastic:9200
      - ELASTICSEARCH_KIBANA_URL=http://kibana:5601/
      - RTB_CROSSTALK_REGION_HOSTS={"US" => "crosstalk"}
    depends_on:
      - db
    networks:
      # - r4f_control
      - r4f_net
    command: bundle exec rails s -p 3000 -b '0.0.0.0'

  elastic:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.2.0
    ports:
      - "9200:9200"
      - "9300:9300"
    environment:
      - discovery.type=single-node
      - "ES_JAVA_OPTS=-Xms1g -Xmx1g"
    depends_on:
      - db
    healthcheck:
        test: ["CMD-SHELL", "curl --silent --fail localhost:9200/_cluster/health ||␣
↪exit 1"]
        interval: 30s
        timeout: 30s
        retries: 3
    volumes:
      - esdata1:/usr/share/elasticsearch/data
    networks:
      # - r4f_kafka
      # - r4f_control
      - r4f_net

  logstash:
```

(continues on next page)

(continued from previous page)

```
    image: docker.elastic.co/logstash/logstash:7.2.0
    ports:
      - "5044:5044"
    environment:
      - "XPACK_MONITORING_ELASTICSEARCH_URL=http://elastic:9200"
      - "XPACK_MONITORING_ENABLED=true"
      - "LS_JAVA_OPTS=-Xmx1g"
    networks:
      # - r4f_kafka
      # - r4f_control
      - r4f_net
    volumes:
      - "./logstash:/usr/share/logstash/pipeline"

  kibana:
    image: docker.elastic.co/kibana/kibana:7.2.0
    environment:
      - SERVER_NAME=elastic
      - ELASTICSEARCH_URL=http://elastic:9200
    ports:
      - "5601:5601"
    networks:
      # - r4f_control
      - r4f_net

volumes:
  db_data: {}
  esdata1:
    driver: local

networks:
  # r4f_rtb:
  # r4f_control:
  # r4f_kafka:
  r4f_net:
```

Build the project. From the directory where you saved the docker-compose.yml file, run the following command:

```
$ docker-compose up -d
```

You should see containers starting each RTB4Free component.

1.1.3 Looking around

To view each of the components running:

```
$ docker ps
```

To stop the containers:

```
$ docker-compose down
```

COMPONENT	LOCATION	LOGIN
Campaign Manager	http://localhost:8080	username: password:
MySQL Database		

To view the containers:

```
$ docker stack ps rtb4free
```

You should see the following response.

```
$ xxxx
```

After the system is started, you can try the following actions.

- **Access the campaing manager by opening a browser to URL http://localhost:3000/. You can log in with user ID demo@rtb**
    - View how a sample campaign is defined.
    - View how a sample creative is defined.
    - View how a sample target is defined.
    - View the sample reports dashboard.
    - Edit the sample campaigns.
- **The application includes the ELK stack (http://elastic.co) for ingesting RTB log events.**
    - You can access Kibana reporting system at URL http://localhost:5601.
    - On the discover screem, the index drop down will allow you to show requests, bid and wins processed by the bidder.
    - Build custom visualizations and dashboards using Kibana.
- Explore the internal works of the bidder by logging into the bidder console at http://localhost:8080/admin_login.

1.1.4 External resources

Here are some external resources to help you learn more about RTB4FREE.

- RTB4FREE documentation

1.2 Getting Started with Amazon AWS

1.3 The Campaign Manager

1.3.1 Creating Your First Campaign

# ARCHIVE

This project is a web application that allows users to define advertising campaigns used by the [RTB4FREE bidder](https://github.com/benmfaul/XRTB). The campaigns bid on Open RTB requests from ad exchanges. Winning bids, pixel impressions, clicks and cost events generated by the bidder are aggregated and presented.

Key Features are:

1. Supports multiple users with login authorization.
2. Define campaigns bidding for video and banner ads. 1. Campaign parameters
        1. Include mutiple banner/videos
        2. Date range

    3. Exchanges 3. Hourly, daily, total budgets 5. Complex matching rules

   1. Banner match parameters 1. Bananer size - height/width 2. Bid price 3. Hourly, daily, total budgets 4. Frequency limits 5. Complex matching rules

   2. Video match paramters 1. Video size and parameters 2. VAST XML file with support for events 3. Hourly, daily, total budgets 4. Frequency limits

5. Define targeting parameters 1. Domain white/blacklists 3. Device make, model, os 4. IAB category inclusion/exclusion 2. Test inclusion/exclusion in large list, ie IP, cookies, domain, etc. 3. Map area for mobile phones

1. Build your own specialized rules if standard targetting is not sufficient.

2. Report summary metrics by campaign. 1. Optionally, link reporting to [ELK](http://elastic.co). stack

1. Easy deployment using Docker containers on Swarm.

2. Built using standard MySQL, Ruby and Ruby on Rails for easy customization and extensiblity.

To see sample screen shots or try a demo using Play with Docker, go [here](http://rtb4free.com),

- On the left side menu, click on the Campaigns icon. You can examine how the demo campaign is defined.

- To define a new campaign, 1. Add a new Campaign. 2. Add a new Banner or Video

   - After parameters are defined, select the newly added campaign.

   3. Add a new Target.

   4. Return to the added Campaign, then select the newly added target.

   5. If a bidder is part of your configuration, after a campaign is updated it is automatically loaded into the RTB4FREE bidder.

- On the left side menu, click on the Users icon to define additional users.

- On the left side menu, click on the Dashboard icon to see defined Campaigns. Since campaign metrics are derived from Elasticsearch, you will not see metrics unless the [DMP](http://rtb4free.com) stack is deployed.

1.4 Bidder

The Bidder engine is the key component of the RTB4FREE platform. It interfaces directly with the SSP exchanges using OpenRTB protocols or proprietary protocols, if necessary, such as Google AdX or OpenX.

The bidder is is a Java version 1.8, and supports OpenRTB version 2.5. The source code with program description can be found here: https://github.com/benmfaul/XRTB

The Bidder features include:

- OpenRTB 2.5 Compliant.
- Supports many SSPs.
- Supports SSL.
- Supports GZIP Compressed Bids.
- Simplified standalone Bidder Management.

The Bidder can operate as a stand-alone system. Campaigns and Exchange interfaces can be defined in its configuration files - see the github source or the demo instructions on this site for detail on how to set up these files.

A sample configuration file, stored at Campaigns/payday.json.

```
{
  "NOfraud" : {
    "type" : "forensiq",
    "threshhold" : 0,
    "ck" : "none",
    "endpoint" : "",
    "bidOnError" : false,
    "db" : "local/GeoIP2-ISP.mmdb",
    "watchlist" : [ "Hetzner", "Amazon web services", "Linode", "Digital Ocean",
↪"Vultr" ]
  },
  "app" : {
    "indexPage": "$INDEXPAGE",
    "concurrency" : "3",
    "threads": "2000",
    "stopped" : false,
    "deadmanswitch": "$ACCOUNTING",
    "ttl" : 300,
    "pixel-tracking-url" : "$EXTERNAL/pixel",
    "winurl" : "$EXTERNAL/rtb/win",
    "redirect-url" : "$EXTERNAL/redirect",
    "vasturl" : "$EXTERNAL/vast",
    "eventurl" : "$EXTERNAL/track",
    "postbackurl" : "$EXTERNAL/postback",
    "adminPort" : "$ADMINPORT",
    "adminSSL" : false,
    "password" : "startrekisbetterthanstarwars",
    "zeromq" : {
      "bidchannel" : "kafka://[$BROKERLIST]&topic=bids",
      "winchannel" : "kafka://[$BROKERLIST]&topic=wins",
      "requests" : "kafka://[$BROKERLIST]&topic=requests",
      "clicks" : "kafka://[$BROKERLIST]&topic=clicks",
      "pixels" : "kafka://[$BROKERLIST]&topic=pixels",
      "videoevents": "kafka://[$BROKERLIST]&topic=videoevents",
      "postbackevents": "kafka://[$BROKERLIST]&topic=postbackevents",
      "status" : "kafka://[$BROKERLIST]&topic=status",
      "reasons" : "kafka://[$BROKERLIST]&topic=reasons",

      "commands": "tcp://$PUBSUB:6001&commands",
      "responses": "tcp://$PUBSUB:6000&responses",
      "xfrport": "6002",
      "frequencygoverner": "$FREQGOV",

      "requeststrategy" : "$REQUESTSTRATEGY"
    },

    "template" : {
      "default" : "{creative_forward_url}",
      "exchange" : {
        "adx" : "",
        "mobclix" : "",
        "nexage" : "",
        "smartyads" : "{creative_forward_url}",
        "atomx" : "{creative_forward_url}",
        "adventurefeeds" : "{creative_forward_url}",
        "gotham" : "{creative_forward_url}",
        "epomx" : "{creative_forward_url}",
```

(continues on next page)

(continued from previous page)

```
        "citenko" : "{creative_forward_url}",
        "kadam" : "{creative_forward_url}",
        "taggify" : "{creative_forward_url}",
        "cappture" : "cappture/{creative_forward_url}",
        "republer" : "{creative_forward_url}",
        "admedia" : "{creative_forward_url}",
        "ssphwy" : "{creative_forward_url}",
        "privatex" : "",
        "smaato" : "richMediaBeacon='%%smaato_ct_url%%'; script='{creative_forward_
↪url}'; clickurl='{redirect_url}/exchange={pub}/{ad_id}/creative_id={creative_id}/
↪price=${AUCTION_PRICE}/lat={lat}/lon={lon}/bid_id={bid_id}?url={creative_forward_
↪url}'; imageurl='{creative_image_url}'; pixelurl='{pixel_url}/exchange={pub}/ad_id=
↪{ad_id}/creative_id={creative_id}/{bid_id}/price=${AUCTION_PRICE}/lat={lat}/lon=
↪{lon}/bid_id={bid_id}';",
        "smaato-builtin" : "{creative_forward_url}",
        "pubmatic" : "{creative_forward_url}"
      }
    },
    "campaigns" : [ "" ]
  },
  "seats" : [ {
    "name" : "google",
    "id" : "google-id",
    "bid" : "/rtb/bids/google=com.jacamars.dsp.rtb.exchanges.google.OpenRTB",
    "extension" : {
      "e_key" : "$GOOGLE_EKEY",
      "i_key" : "$GOOGLE_IKEY"
    }
  }, {
    "name" : "openx",
    "id" : "openx-id",
    "bid" : "/rtb/bids/openx=com.jacamars.dsp.rtb.exchanges.openx.OpenX",
    "extension" : {
      "e_key" : "$OPENX_EKEY",
      "i_key" : "$OPENX_IKEY"
    }
  }, {
    "name" : "appnexus",
    "id" : "test-appnexus-id",
    "bid" : "/rtb/bids/appnexus=com.jacamars.dsp.rtb.exchanges.appnexus.Appnexus"
  }, {
    "name" : "adx",
    "id" : "adx-seat-id",
    "bid" : "/rtb/bids/adx=com.jacamars.dsp.rtb.exchanges.adx.DoubleClick",
    "extension" : {
      "e_key" : "$ADX_EKEY",
      "i_key" : "$ADX_IKEY"
    }
  }, {
    "name" : "c1xus",
    "id" : "c1xus",
    "bid" : "/rtb/bids/c1xus=com.jacamars.dsp.rtb.exchanges.C1XUS&usesPiggyBackWins"
  }, {
    "name" : "stroer",
    "id" : "stroer-id",
    "bid" : "/rtb/bids/stroer=com.jacamars.dsp.rtb.exchanges.Stroer"
  }, {
```

(continues on next page)

(continued from previous page)

```
    "name" : "waardx",
    "id" : "waardx-id",
    "bid" : "/rtb/bids/waardx=com.jacamars.dsp.rtb.exchanges.Generic&!usesEncodedAdm"
  }, {
    "name" : "index",
    "id" : "index-id",
    "bid" : "/rtb/bids/index=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "intango",
    "id" : "intango-id",
    "bid" : "/rtb/bids/intango=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "vdopia",
    "id" : "vdopia-id",
    "bid" : "/rtb/bids/vdopia=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "vertamedia",
    "id" : "vertamedia-id",
    "bid" : "/rtb/bids/vertamedia=com.jacamars.dsp.rtb.exchanges.Generic&!
↪usesEncodedAdm&usesPiggyBackWins"
  }, {
    "name" : "ventuno",
    "id" : "ventuno-id",
    "bid" : "/rtb/bids/ventuno=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "medianexusnetwork",
    "id" : "mnn-id",
    "bid" : "/rtb/bids/medianexusnetwork=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "wideorbit",
    "id" : "wideorbit-id",
    "bid" : "/rtb/bids/wideorbit=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "smartadserver",
    "id" : "smartadserver-id",
    "bid" : "/rtb/bids/smartadserver=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "c1x",
    "id" : "c1x",
    "bid" : "/rtb/bids/c1x=com.jacamars.dsp.rtb.exchanges.C1X&usesPiggyBackWins"
  }, {
    "name" : "axonix",
    "id" : "axonix-id",
    "bid" : "/rtb/bids/axonix=com.jacamars.dsp.rtb.exchanges.Generic"
  }, {
    "name" : "adventurefeeds",
    "id" : "adventurefeedid",
    "bid" : "/rtb/bids/adventurefeeds=com.jacamars.dsp.rtb.exchanges.Adventurefeeds"
  }, {
    "name" : "adprudence",
    "id" : "adprudenceid",
    "bid" : "/rtb/bids/adprudence=com.jacamars.dsp.rtb.exchanges.Adprudence"
  }, {
    "name" : "citenko",
    "id" : "citenkoid",
    "bid" : "/rtb/bids/citenko=com.jacamars.dsp.rtb.exchanges.Citenko"
  }, {
```

(continues on next page)

(continued from previous page)

```
    "name" : "kadam",
    "id" : "kadamid",
    "bid" : "/rtb/bids/kadam=com.jacamars.dsp.rtb.exchanges.Kadam"
  }, {
    "name" : "gotham",
    "id" : "gothamid",
    "bid" : "/rtb/bids/gotham=com.jacamars.dsp.rtb.exchanges.Gotham"
  }, {
    "name" : "atomx",
    "id" : "atomxseatid",
    "bid" : "/rtb/bids/atomx=com.jacamars.dsp.rtb.exchanges.Atomx"
  }, {
    "name" : "smartyads",
    "id" : "smartypants",
    "bid" : "/rtb/bids/smartyads=com.jacamars.dsp.rtb.exchanges.Smartyads"
  }, {
    "name" : "nexage",
    "id" : "99999999",
    "bid" : "/rtb/bids/nexage=com.jacamars.dsp.rtb.exchanges.Nexage"
  }, {
    "name" : "privatex",
    "id" : "5555555",
    "bid" : "/rtb/bids/privatex=com.jacamars.dsp.rtb.exchanges.Privatex"
  }, {
    "name" : "fyber",
    "id" : "seat1",
    "bid" : "/rtb/bids/fyber=com.jacamars.dsp.rtb.exchanges.Fyber"
  }, {
    "name" : "smaato-builtin",
    "id" : "seat1",
    "bid" : "/rtb/bids/smaato-builtin=com.jacamars.dsp.rtb.exchanges.Smaato"
  }, {
    "name" : "smaato",
    "id" : "seat1",
    "bid" : "/rtb/bids/smaato=com.jacamars.dsp.rtb.exchanges.Smaato"
  }, {
    "name" : "epomx",
    "id" : "seat1",
    "bid" : "/rtb/bids/epomx=com.jacamars.dsp.rtb.exchanges.Epomx"
  }, {
    "name" : "cappture",
    "id" : "cappptureseatid",
    "bid" : "/rtb/bids/cappture=com.jacamars.dsp.rtb.exchanges.Cappture"
  }, {
    "name" : "taggify",
    "id" : "taggifyid",
    "bid" : "/rtb/bids/taggify=com.jacamars.dsp.rtb.exchanges.Taggify"
  }, {
    "name" : "republer",
    "id" : "republerid",
    "bid" : "/rtb/bids/republer=com.jacamars.dsp.rtb.exchanges.Republer"
  }, {
    "name" : "admedia",
    "id" : "admediaid",
    "bid" : "/rtb/bids/admedia=com.jacamars.dsp.rtb.exchanges.AdMedia"
  }, {
    "name" : "ssphwy",
```

(continues on next page)

(continued from previous page)

```
    "id" : "ssphwyid",
    "bid" : "/rtb/bids/ssphwy=com.jacamars.dsp.rtb.exchanges.SSPHwy"
  }, {
    "name" : "pubmatic",
    "id" : "pubmaticid",
    "bid" : "/rtb/bids/pubmatic=com.jacamars.dsp.rtb.exchanges.Pubmatic"
  }, {
    "name" : "xapads",
    "id" : "xapads-id",
    "bid" : "/rtb/bids/xapads=com.jacamars.dsp.rtb.exchanges.Generic&!usesEncodedAdm"
  }, {
    "name" : "admedia",
    "id" : "admedia-id",
    "bid" : "/rtb/bids/admedia=com.jacamars.dsp.rtb.exchanges.Generic&!usesEncodedAdm"
  }],
  "NOs3" : {
    "access_key_id" : "$S3ACCESSKEY",
    "secret_access_key" : "S3SECRETKEY",
    "region" : "S3REGION",
    "bucket" : "S3BUCKET",
    "NOproxyprotocol" : "http",
    "NOproxyhost" : "edgecache.kdc3-dt",
    "NOproxyport" : 9000
  },
  "lists" : [ {
    "filename" : "data/adxgeo.csv",
    "name" : "@ISO2-3",
    "type" : "com.jacamars.dsp.rtb.tools.IsoTwo2Iso3"
  }, {
    "filename" : "data/METHBOT.txt",
    "name" : "@CIDR",
    "type" : "CidrMap"
  }, {
    "filename" : "data/adxgeo.csv",
    "name" : "@ADXGEO",
    "type" : "com.jacamars.dsp.rtb.exchanges.adx.AdxGeoCodes"
  }, {
    "filename" : "data/zip_codes_states.csv",
    "name" : "@ZIPCODES",
    "type" : "com.jacamars.dsp.rtb.tools.LookingGlass"
  } ]
}
```

1.5 Data Management Platform

1.6 Analytics

1.7 Developer Resources

1.7.1 Connecting to Bidstream Suppliers

1.8 Installation

1.9 System Configuration

1.10 Customization

1.11 Running Tests

1.12 Issues

issue-labels

1.13 Changelog

1.14 Security

1.15 Contribute

1.16 Internationalization

1.17 Frequently Asked Questions

1.18 Consulting

1.19 Support

1.20 Roadmap

1.21 Team