Integrate a Bidder

Integrate a Bidder

This page describes how to integrate a bidder with AppNexus. It begins with an overview of the different "layers" of the integration, and ends with a worked example (using actual API calls) of a simple integration that will get you up and running quickly in our testing environment. It also provides links to more detailed information elsewhere on our wiki.

On This Page

System Overview

At a high level, there are two "layers" of the system we need to be concerned with during bidder setup:

- **Real-Time Layer (RTB)**: This is the heart of the action, where your bidder will participate in the real-time auction.
- **Configuration Layer (API)**: This is where you will configure your bidder' s "business logic" so it can bid on impressions; in other words, filtering out unwanted impressions, setting up users, adding the creatives your members want to serve, etc.

The diagram below shows the high-level architecture of a bidder and the various requests and responses it receives and sends.

> In the diagram below, click on the **request** and **response** bubbles in the **Real-Time Layer** and the names of API services (such as the **Bidder Profile**) in the **Configuration Layer** to open the corresponding documentation in a new browser tab.

How to Set Up Your Bidder

In this section we'll walk through the entire process of setting up a bidder on the platform. We'll begin by making the API calls necessary to hook up the pipes.

Authenticate with the API

Before we can do anything else, we have to log in.

> For more detailed information about authenticating via our API, see the Authentication Service.

```
$ cat auth.json

       {
           "auth":
           {
               "username" : "rloveland",
               "password" : "AppNexus1!"
           }
       }
```

Post to the test environment to authenticate:

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -c cookies -X POST -d @auth.json $IB_TESTING/auth

{ response": { "status": "OK", ... } }
```

View your Bidder Object

The bidder object represents your bidder in our system. As such, it has a lot of fields that you can use to configure how your bidder interacts with our platform. Think of it as the central "hook" on which you'll hang much of the rest of your configuration. A bidder object should already have been created for you by your AppNexus representative.

> In the example below, we make a `GET` call to view the bidder object, but we don't explain any of its details. For more detailed information about the bidder object, see the Bidder Service.

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies $IB_TESTING/bidder/497

{
    "response":{
        "status":"OK",
        "count":1,
        "start_element":null,
        "num_elements":null,
        "bidder":{
            "id":497,
            "name":"Rich's Cool Bidder",
            "short_name":"RCB",
            "throttling":"none",
            "active":true,
            "bid_uri":"\/bid",
            "notify_uri":"\/notify",
            "click_uri":"\/click",
            "ready_uri":"\/ready",
            "pixel_uri":"\/pixel",
            "audit_notify_uri":null,
            "send_exelate":false,
            "send_ixi":false,
            "send_lucid":false,
            "send_datran":false,
```

```
        "send_class_2":true,
        "send_class_3":true,
        "notify_full_auction":false,
        "userdata_javascript":null,
        "userdata_entity_id":null,
        "send_owned_blacklist":false,
        "exclude_unowned":false,
        "send_unaudited":true,
        "lucid_level":0,
        "dongle":"ABFAB",
        "num_conns":3,
        "bid_percent":100,
        "is_dp":false,
        "setuid_function":null,
        "parent_profile_id":431739,
        "last_activity":"2015-02-05 22:28:55",
        "default_currency":"USD",
        "max_allowed_profiles":5,
        "notify_lost":false,
        "notify_pending":false,
        "notify_no_bid":false,
        "always_send_owned_segments":true,
        "object_limit_notify_email":null,
        "account_owner_user_id":null,
        "protocol_id":1,
        "protocol_name":"none",
        "child_profiles":null,
        "lifetime_budget":null,
        "lifetime_budget_imps":null,
        "daily_budget":null,
        "daily_budget_imps":null
    },
    "dbg":{
        "instance":"01.api.client03.lax1",
        "slave_hit":false,
        "db":"master",
        "time":26.602983474731,
        "version":"1.14.128",
        "warnings":[
        ],
        "start_microtime":1423681754.7112
    }
```

```
    }
}
```

Update your Bidder object to receive unaudited inventory

In order to avoid drastically limiting the number of bid requests sent to your bidder, you need to set the `send_unaudited` field on the bidder object to `true` using the Bidder Service. This is true even though there is a `non_audited_url_action` (which defaults to `"include"`) on the Legacy Bidder Profile Service.

```
$ cat send-unaudited.json
{
    "bidder": {
        "send_unaudited": true,
        "id": 497
    }
}
```

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X PUT -d @send-unaudited.json $IB_TESTING/bidder/497

{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":497
,"bidder":{"id":497,"name":"Rich's Cool
Bidder","short_name":"RCB","throttling":"none","active":true,"bid_uri":"\/bid","notify
_uri":"\/notify","click_uri":"\/click","ready_uri":"\/ready","pixel_uri":"\/pixel","au
dit_notify_uri":null,"send_exelate":false,"send_ixi":false,"send_lucid":false,"send_da
tran":false,"send_class_2":true,"send_class_3":true,"notify_full_auction":false,"userd
ata_javascript":null,"userdata_entity_id":null,"send_owned_blacklist":false,"exclude_u
nowned":false,"send_unaudited":true,"lucid_level":0,"dongle":"ABFAB","num_conns":3,"bi
d_percent":100,"is_dp":false,"setuid_function":null,"parent_profile_id":431739,"last_a
ctivity":"2015-02-05
22:28:55","default_currency":"USD","max_allowed_profiles":5,"notify_lost":false,"notif
y_pending":false,"notify_no_bid":false,"always_send_owned_segments":true,"object_limit
_notify_email":null,"account_owner_user_id":null,"protocol_id":1,"protocol_name":"none
","child_profiles":null,"lifetime_budget":null,"lifetime_budget_imps":null,"daily_budg
et":null,"daily_budget_imps":null},"dbg":{"instance":"01.api.client03.lax1","slave_hit
":false,"db":"master","time":52.114009857178,"version":"1.14.128","warnings":[],"start
_microtime":1423176120.4424}}}
```

View your Member Object

You need to have at least one member that buys through your bidder. You should have had a member created for you by your AppNexus representative as part of the onboarding process. The member is where you will configure much of the "business logic" such as user segments, creatives, etc.

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b $IB_TESTING/member/4782

{
    "response":{
        "status":"OK",
```

```
"count":1,
"start_element":0,
"num_elements":100,
"member":{
    "id":4782,
    "bidder_id":497,
    "pitbull_segment_id":0,
    "pitbull_segment_value":0,
    "agent_id":null,
    "code":null,
    "default_buyer_group_id":null,
    "active":true,
    "contract_approved":true,
    "buyer_credit_limit":2500,
    "billing_name":"Rich's Ad Network",
    "billing_address_1":null,
    "billing_address_2":null,
    "billing_city":null,
    "billing_region":null,
    "billing_postal_code":null,
    "billing_country":null,
    "email_code":null,
    "seller_revshare_pct":95,
    "default_tag_id":null,
    "default_ad_profile_id":302266,
    "description":null,
    "ym_profile_id":null,
    "default_venue_id":155164,
    "expose_segments":false,
    "dw_member":false,
    "default_inv_source_id":null,
    "expose_global_inventory_sources":true,
    "buyer_clearing_fee_pct":null,
    "default_discrepancy_pct":null,
    "max_daily_credit_pct":5,
    "cname":null,
    "timezone":"EST5EDT",
    "dongle":null,
    "audit_notify_email":null,
    "sherlock_notify_email":null,
    "last_activity":"2015-01-22 21:12:28",
    "is_billable":true,
    "note":null,
    "primary_type":"network",
    "platform_exposure":"private",
    "contact_email":null,
    "allow_ad_profile_override":true,
    "pops_enabled_UI":false,
    "default_accept_supply_partner_usersync":true,
    "default_accept_data_provider_usersync":true,
    "default_accept_demand_partner_usersync":true,
    "short_name":null,
    "daily_imps_verified":null,
    "daily_imps_self_audited":null,
    "daily_imps_unaudited":null,
    "is_iash_compliant":false,
    "expose_eap_ecp_placement_settings":false,
    "visibility_profile_id":null,
    "enable_budget_check":true,
```

```
"allow_priority_audit":false,
"default_external_audit":false,
"reporting_decimal_type":"decimal",
"safety_budget_cap":null,
"enable_click_and_imp_trackers":false,
"plugins_enabled":false,
"bid_landscape_sampling_pct":0,
"default_auction_timeout_ms":null,
"delay_suspicious_creative_deactivation":false,
"max_hosted_video_size":null,
"domain_blacklist_email":null,
"enable_facebook":false,
"require_facebook_preaudit":true,
"asi_second_price":false,
"usersync_segment_id":null,
"default_referrer_url":null,
"enable_real_time_bidding":true,
"enable_reselling":true,
"throttle_pct":"100.00",
"custom_pub_querystring_enabled":false,
"enable_targeted_segment_logging":false,
"default_content_retrieval_timeout_ms":0,
"default_enable_for_mediation":false,
"audit_tier":3,
"max_active_optimization_zones":100,
"prioritize_margin":false,
"developer_id":null,
"lifetime_budget":null,
"lifetime_budget_imps":null,
"daily_budget":null,
"daily_budget_imps":null,
"default_creatives":null,
"account_owner_user":null,
"contracts":[
    {
        "id":3583,
        "last_activity":"2015-01-23 21:14:18",
        "start_date":"2015-01-24 00:00:00",
        "end_date":"2018-12-31 23:59:59",
        "auction_revshare":null,
        "auction_minimum_cpm":null,
        "ad_serving_cpm":null,
        "auditing_fee_per_creative":null,
        "creative_size_minimum_bytes":null,
        "creative_size_fee_per_gb":null,
        "monthly_minimum_spend":null,
        "clearing_revshare":null,
        "auction_revshare_pct":null,
        "clearing_revshare_pct":null,
        "auction_maximum_cpm":null,
        "apply_min_cpm_to_clearing":false,
        "auction_revshare_type":"deduction",
        "seller_type":"platform",
        "note":null,
        "data_siphon_fee":null,
        "mapuid_fee":"0.00",
        "monthly_minimum_imps":null,
        "waive_ad_serving_fees":false,
        "adx_auction_service_fee_pct":null,
```

```
                "direct_clear_fee_pct":null,
                "contract_type":null,
                "creative_audit_fee":"0.00",
                "creative_priority_fee_1":"0.00",
                "creative_priority_fee_2":"25.00",
                "imptracker_cpm":null,
                "clicktracker_cpc":null,
                "auto_renewal_term":null,
                "seller_serving_cpm":"0.0000",
                "monthly_spend_based_minimum":null,
                "seller_revshare_pct":null,
                "seller_revshare_minimum":0,
                "secure_whitelabel_pixel_fee":"0.00",
                "monthly_minimum_requests":null,
                "managed_hosted_video_cpm":null,
                "cross_net_hosted_video_cpm":null,
                "seller_auction_request_cpm":null,
                "monthly_service_fee_minimum_1":null,
                "monthly_service_fee_minimum_2":null,
                "selling_enabled":true,
                "pricing_version":"2013.01.02",
                "seller_console_buyer_revshare_pct":null,
                "seller_bidder_buyer_revshare_pct":null,
                "auction_revshare_platform_inv_pct":"0.00",
                "auction_revshare_platform_inv_type":"deduction",
                "auction_revshare_partner_inv_pct":"0.00",
                "auction_revshare_partner_inv_type":"deduction",
                "is_active":true
            }
        ],
        "contact_info":null,
        "price_buckets":null,
        "serving_domain":null,
        "features":null,
        "active_contract":{
            "auditing_fee_per_creative":null,
            "creative_priority_fee_1":"0.00",
            "creative_priority_fee_2":"25.00"
        }
    },
    "dbg":{
        "instance":"01.api.client03.lax1",
        "slave_hit":false,
        "db":"master",
        "time":92.861175537109,
        "version":"1.14.128",
        "warnings":[
        ],
        "start_microtime":1423851476.6719
    }
```

```
    }
}
```

Create a Segment for your Member and add yourself to it

In this step, we'll create a segment and put ourselves in it to test the integration. Later on, we'll target this segment with the Bidder Profile during testing. This will serve several purposes:

- Test that the "pipes" are open
- Test that our ability to set up the profile's targeting works as expected
- Allow us to send our bidder a very small amount of traffic (since presumably only a few users will be added to the testing segment)

```
$ cat create-segment-json
{
    "segment":
    {
        "member_id": 4782,
        "active": true,
        "short_name": "ethical kangaroo segment 00",
        "code":"ethical-kangaroo-00"
    }
}
```

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X POST -d @create-segment.json $IB_TESTING/segment/4782

{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":110
944,"segment":{"id":110944,"active":true,"description":null,"member_id":4782,"code":"e
thical-kangaroo-00","provider":"","price":0,"short_name":"ethical kangaroo segment
00","expire_minutes":null,"category":null,"enable_rm_piggyback":false,"last_activity":
"2015-01-21
23:23:22","max_usersync_pixels":null,"parent_segment_id":null,"querystring_mapping":nu
ll},"dbg":{"instance":"01.api.client03.lax1","slave_hit":false,"db":"master","time":49
.130916595459,"version":"1.14.128","warnings":[],"start_microtime":1421883270.4392}}}
```

Add yourself to the segment by visiting this URL:

```
http://ib-test.adnxs.com/seg?add=110944
```

Create a Bidder Profile

In this step, we'll create a bidder profile that only allows impressions shown to users in our test segment (created in the last step) to be sent in bid requests.

In the example below, the targeting breaks down like this:

- We include (that is, target) the testing segment ID we just created.
- Although we set the passthrough percent to 0, we should still receive bid requests for users in our members' segments. For details, see

the docs for `passthrough_percent` on the Legacy Bidder Profile Service.

- The segment boolean operator is set to "or" in case we'd like to add more testing segments later.
- Finally, we target a few common mobile sizes just for fun.

> This profile is meant to provide a simple example for onboarding use. It's unlikely to be appropriate for production scenarios. For more information about all of the options available with profiles, see the Legacy Bidder Profile Service and the Bidder Profile - FAQ.

```
$ cat create-bidder-profile-json

{
    "profile": {
        "description": "Ethical Kangaroo Test Segment",
        "passthrough_percent": 0,
        "code": "ethical-kangaroo-00",
        "segment_targets": [
            {
                "id": 110944,
                "action": "include"
            }
        ],
        "segment_boolean_operator": "or",
        "size_targets": [
            {
                "width": 300,
                "height": 250
            },
            {
                "width": 300,
                "height": 50
            }
        ]
    }
}
```

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X POST -d @create-bidder-profile.json $IB_TESTING/profile/497

{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":431
739,"profile":{"id":431739,"code":"ethical-kangaroo-00","description":"Ethical
Kangaroo Test
Segment","country_action":"exclude","region_action":"exclude","city_action":"exclude",
"browser_action":"exclude","use_inventory_attribute_targets":false,"last_activity":"20
15-01-22
05:01:25","daypart_timezone":null,"dma_action":"exclude","domain_action":"exclude","do
main_list_action":"exclude","inventory_action":"exclude","language_action":"exclude","
segment_boolean_operator":"or","min_session_imps":null,"session_freq_type":"platform",
"carrier_action":"exclude","supply_type_action":"exclude","device_type_action":"exclud
e","screen_size_action":"exclude","device_model_action":"exclude","location_target_rad
ius":null,"location_target_latitude":null,"location_target_longitude":null,"querystrin
g_action":"exclude","querystring_boolean_operator":"and","is_expired":false,"non_audit
ed_url_action":"include","member_id":null,"daypart_bitmap":null,"passthrough_percent":
0,"country_targets":null,"region_targets":null,"city_targets":null,"inv_class_targets"
:null,"inventory_source_targets":null,"inventory_attribute_targets":null,"age_targets"
:null,"daypart_targets":null,"browser_targets":null,"browser_family_targets":null,"dma
_targets":null,"domain_targets":null,"domain_list_targets":null,"language_targets":nul
l,"size_targets":[{"width":300,"height":50},{"width":300,"height":250}],"postal_code_t
argets":null,"member_targets":[{"id":4782,"action":"include","third_party_auditor_id":
null,"billing_name":"unexposed"}],"segment_group_targets":null,"carrier_targets":null,
"supply_type_targets":null,"device_type_targets":null,"screen_size_targets":null,"devi
ce_model_targets":null,"querystring_targets":null,"gender_targets":null,"intended_audi
ence_targets":null,"inventory_group_targets":null,"inventory_network_resold_targets":n
ull,"ip_targets":null,"operating_system_targets":null,"operating_system_family_targets
":null,"position_targets":null,"site_targets":null,"venue_targets":null,"operating_sys
tem_extended_targets":null,"segment_targets":[{"id":110944,"action":"include","start_m
inutes":null,"expire_minutes":null,"other_less":null,"other_greater":null,"other_equal
s":null,"code":"ethical-kangaroo-00","name":"ethical kangaroo segment
00","deleted":false,"other_in_list":null}]},"dbg":{"instance":"01.api.client03.lax1","
slave_hit":false,"db":"master","time":301.28717422485,"version":"1.14.128","warnings":
[],"start_microtime":1421903554.9082}}}
```

Associate your Profile with your Bidder Object

The profile we've just configured can't take effect until it's associated with your bidder object. Since we're doing integration testing, we'll set the profile we just created as the parent profile.

> Much more complex targeting configurations are possible in production settings with the right combination of parent and child profiles. For more information, see the Legacy Bidder Profile Service and the Bidder Profile - FAQ.

```
$ cat update-bidder-with-bidder-profile-json
{
    "bidder": {
        "id": 497,
        "parent_profile_id" : 431739
    }
}
```

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X PUT -d @update-bidder-with-bidder-profile.json
$IB_TESTING/bidder/497
{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":497
,"bidder":{"id":497,"name":"Rich's Cool
Bidder","short_name":"RCB","throttling":"none","active":false,"bid_uri":"\/bid","notif
y_uri":"\/notify","click_uri":"\/click","ready_uri":"\/ready","pixel_uri":"\/pixel","a
udit_notify_uri":null,"send_exelate":false,"send_ixi":false,"send_lucid":false,"send_d
atran":false,"send_class_2":true,"send_class_3":true,"notify_full_auction":false,"user
data_javascript":null,"userdata_entity_id":null,"send_owned_blacklist":false,"exclude_
unowned":false,"send_unaudited":false,"lucid_level":0,"dongle":"ABFAB","num_conns":3,"
bid_percent":100,"is_dp":false,"setuid_function":null,"parent_profile_id":431739,"last
_activity":"2015-01-22
05:07:10","default_currency":"USD","max_allowed_profiles":5,"notify_lost":false,"notif
y_pending":false,"notify_no_bid":false,"always_send_owned_segments":true,"object_limit
_notify_email":null,"account_owner_user_id":null,"protocol_id":1,"protocol_name":"none
","child_profiles":null,"lifetime_budget":null,"lifetime_budget_imps":null,"daily_budg
et":null,"daily_budget_imps":null},"dbg":{"instance":"01.api.client03.lax1","slave_hit
":false,"db":"master","time":56.746959686279,"version":"1.14.128","warnings":[],"start
_microtime":1421903900.3984}}}
```

Add a Test Creative

In this step, we'll add a creative to serve to the users in our testing segment. After we upload this creative, we'll need to set up our bidder to respond to a Bid Request in the testing environment with a Bid Response that includes this creative when you bid on the test TinyTags provided to you by your AppNexus representative. This will test that our profile (along with the rest of our integration) is working as expected.

> This simple example creative uses very few of the creative configuration options we support. For more detailed information about the many types of creative configurations, see the Creative Service.

```
$ cat add-creative.json
      {
          "creative": {
              "width": 300,
              "height": 250,
              "media_url": "https://placekitten.com/g/300/250",
              "click_url": "http://www.reddit.com/r/aww/",
              "template": {
                  "id": 4
              }
          }
      }
```

For more information on using the Client Testing environment to test creatives, see Using the Client Testing environment below.

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X POST -d @add-creative.json $IB_TESTING/creative/4782

{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":976
504,"creative":{"id":976504,"active":true,"member_id":4782,"description":null,"code":n
ull,"code2":null,"media_url":"https:\/\/placekitten.com\/g\/300\/250","audit_status":"
pending","allow_audit":true,"size_in_bytes":0,"last_checked":null,"not_found":0,"added
_by_bidder":null,"campaign":null,"placement":null,"format":null,"width":300,"height":2
50,"click_url":"http:\/\/www.reddit.com\/r\/aww\/","landing_page_url":null,"flash_clic
k_variable":null,"no_iframes":false,"content":null,"original_content":null,"track_clic
ks":true,"audit_feedback":null,"created_on":"2015-01-22
16:32:52","flash_backup_url":null,"is_self_audited":false,"file_name":null,"is_prohibi
ted":false,"last_activity":"2015-01-22
16:32:52","passed_sherlock_audit":true,"is_expired":false,"creative_upload_status":nul
l,"backup_upload_status":null,"filter":null,"is_suspicious":false,"suspicious_activity
_timestamp":null,"media_subtypes":["banner"],"no_adservers":false,"google_audit_status
":"pending","google_audit_feedback":null,"msft_audit_status":"pending","msft_audit_fee
dback":null,"msft_external_audit_status":"pending","msft_external_audit_feedback":null
,"is_blanking":false,"is_rotating":false,"text_title":null,"text_description":null,"te
xt_display_url":null,"click_action":"click-to-web","click_target":"http:\/\/www.reddit
.com\/r\/aww\/","ssl_status":"disabled","allow_ssl_audit":false,"media_url_secure":nul
l,"content_secure":null,"original_content_secure":null,"flash_backup_url_secure":null,
"is_hosted":false,"content_source":"standard","thirdparty_creative_id":null,"thirdpart
y_campaign_id":null,"facebook_audit_status":null,"facebook_audit_feedback":null,"custo
m_request_template":null,"pop_values":null,"brand":{"id":1,"name":"Unknown","category_
id":0},"language":{"id":1,"name":"English"},"template":{"id":4},"thirdparty_page":null
,"segments":null,"pixels":null,"mobile":null,"brand_id":1,"sla":"0","sla_eta":"2015-01
-23
06:33:52"},"dbg":{"instance":"01.api.client03.lax1","slave_hit":false,"db":"master","t
ime":286.93890571594,"version":"1.14.128","warnings":[],"start_microtime":1421945046.2
26}}}
```

Add a Bidder Instance

The bidder instance object represents a particular bidder server running in the data center. In this example, we set the data center ID to (NYM), since that's where the client testing environment lives.

> This step assumes that you already have a bidder up and running that can respond to bid requests, ready requests, etc., as detailed in the System Overview.

> For more information about configuring a bidder instance, see the Bidder Instance Service

```
$ cat create-bidder-instance.json
{
    "instance": {
        "bidder_id": 497,
        "active": true,
        "datacenter_id": 5,
        "ip_address": "10.3.64.215",
        "port": 49995
    }
}
```

```
$ export IB_TESTING="http://api-test.adnxs.com";
$ curl -b cookies -X POST -d @create-bidder-instance.json
$IB_TESTING/bidder-instance/497
{"response":{"status":"OK","count":1,"start_element":null,"num_elements":null,"id":154
3,"instance":{"id":1543,"bidder_id":497,"active":true,"datacenter_id":2,"datacenter":"
la","ip_address":"10.3.64.215","port":49995,"last_activity":"2015-01-26
19:08:33","hostname":null,"qps_limit":null,"dns_interval":null,"min_conns":1,"max_conn
s":null,"receive_type_id":0},"dbg":{"instance":"01.api.client03.lax1","slave_hit":fals
e,"db":"master","time":147.5989818573,"version":"1.14.128","warnings":[],"start_microt
ime":1422300019.7515}}}
```

Test the Integration

First, you should test the integration using our Client Testing environment, explained below. If this works as expected, you should be able to pull a debug log for the TinyTags provided to you by your AppNexus contact as shown on our Troubleshooting page. If all is well, you will be able to compare the debug logs with your bidder's own logging to make sure that the integration is working as expected.

If you are still not seeing the bid requests you expect:

1. Double check your configuration against the instructions on this page.
2. Check out our Troubleshooting documentation

Using the Client Testing environment

The Client Testing environment provides a version of the Impbus and Impbus API that you can use to test your workflows and API implementations without incurring spend (as you would in production). The Client Testing environment's codebase and data are now updated every month. This means your testing environment will never be more than 30 days (and often less) behind the version of AppNexus code that is running in Production. In addition, all Production data will also automatically be copied over to the Client Testing environment (including your member accounts and credentials) each month. This will allow far more robust testing against the latest features.

For reference, here are the endpoints for the Production and new Client Testing environments.

Product	Product endpoint	Client Testing endpoint
Impbus	http://ib.adnxs.com	http://ib-test.adnxs.com
Impbus API	http://api.adnxs.com	http://api-test.adnxs.com

Read more about the Client Testing Environment.

Related Topics

- Troubleshooting
- Bidder Instance Service
- Legacy Bidder Profile Service

- Bidder Profile - FAQ
- Creative Service