

Segment Service

Segment Service

The Segment Service lets you create segment IDs, which are then used to create segment pixels for placement on inventory pages. If you are working with third-party data providers through the AppNexus platform, it will also show you a list of segments for those providers.

- Segments are associated with members, and can optionally be associated with a particular advertiser.
- All segment data will be stored in the server-side cookie store and passed to the bidder associated with the owning member on every bid request.
- An advertiser association is for reference and use in our Console user interface.

The shared "birthday cookie" segment has an ID of 1 and is available to all platform members. For more information, see [AppNexus Birthday Cookies](#) (Customer login required).

Once you have a segment ID, the basic format of a segment pixel is ``, where 123 is the segment ID. For more information about segment pixels, see [Working with Segments](#) (Customer login required).

Be sure to wait approximately 20 minutes before trying to add users to any newly created segments (to allow these segments to be propagated to all servers in our cloud). In addition, as a best practice, try to minimize the creation of new segments, re-use existing segments where possible or use segment "values" to further sub-divide users within existing segments. These practices will ensure successful user uploads to segments. For details on creating segment "values", see [Segment Pixels Advanced](#) and [Segment Targeting](#).

On This Page

- [REST API](#)
- [JSON Fields](#)
- [Examples](#)

REST API

Add a new segment:

```
POST https://api.appnexus.com/segment
(segment JSON)
```

Add a new advertiser segment:

```
POST https://api.appnexus.com/segment?advertiser_id=ADVERTISER_ID
POST https://api.appnexus.com/segment?advertiser_code=ADVERTISER_CODE
(segment JSON)
```

Modify an existing segment:

```
PUT https://api.appnexus.com/segment?id=SEGMENT_ID
PUT https://api.appnexus.com/segment?code=SEGMENT_CODE
(segment JSON)
```

Modify an existing advertiser segment:

```
PUT https://api.appnexus.com/segment?id=SEGMENT_ID&advertiser_id=ADVERTISER_ID
PUT https://api.appnexus.com/segment?code=SEGMENT_CODE&advertiser_code=ADVERTISER_CODE
(segment JSON)
```

To change a segment from advertiser-owned to network-owned, make a PUT call, passing the segment ID and member ID in the query string and setting `advertiser_id` to null in the JSON file. See this [example](#) for a demonstration.

View all segments:

```
GET https://api.appnexus.com/segment
```

View multiple segments by ID using a comma-separated list:

```
GET https://api.appnexus.com/segment?id=1,2,3
```

View a particular segment:

```
GET https://api.appnexus.com/segment?id=SEGMENT_ID
GET https://api.appnexus.com/segment?code=SEGMENT_CODE
```

Search for segments with IDs, short names, or codes containing certain characters:

```
GET https://api.appnexus.com/segment?search=SEARCH_TERM
```

Delete an existing segment:

```
DELETE https://api.appnexus.com/segment?id=SEGMENT_ID
```

JSON Fields

Field	Type	Description	Def
-------	------	-------------	-----

id	int	The AppNexus ID assigned by the API to reference the segment. When switching a segment from advertiser-owned to network-owned, you must pass this information in the querystring.	
code	string(50)	<p>The user-defined code for calling the segment. For more information, see Working with Segments (Customer login required).</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>The value of the <code>code</code> field is not case-sensitive (e.g., "Test" is the same as "test"), so each code must be unique regardless of case. However, when referring to the code in query string targeting, case-sensitivity matters (e.g., if the value of the <code>code</code> field is "Test", the parameter in a query string referring to it must also be "Test").</p> </div>	
state	enum	The state of the segment. This determines whether the segment can be used. Possible values: <code>active</code> or <code>inactive</code> .	act
short_name	string(255)	The short name used to describe the segment.	
description	string	The optional description for this segment.	
member_id	int	<p>The ID of the member that owns this segment.</p> <div style="border: 1px solid green; padding: 10px; margin-top: 10px;"> <p>When switching a segment from advertiser-owned to network-owned, you must pass this information in the query string. See this example for more details.</p> </div>	
category	string	Deprecated. This field is read-only.	

price	double	Deprecated. This field is currently not operational	0
expire_minutes	int	<p>The number of minutes the user is kept in the segment. Segments with no expiration time will be expired from AppNexus's server-side data store within 90 days. If you want to add the user to the segment only for the duration of the ad call, set to 0. Changing this value does not retroactively affect users already in the segment. Also, if a user is re-added, the expiration window resets.</p> <div style="border: 1px solid green; padding: 5px; margin: 10px 0;"> <p>To keep users in the segment for the 180 day maximum, set this to <code>null</code>.</p> </div>	
enable_rm_piggyback	boolean	If true, piggybacking RM pixels is enabled.	
max_usersync_pixels	int	The maximum number of third-party user sync pixels to piggyback onto the segment pixel. Set to 0 to block all third-party user sync pixels. If blank (null), the segment defaults to 0.	0
last_modified	timestamp	The date and time when the segment was last modified.	
provider_id	int	The ID of the data provider that owns the segment. It is read-only and can be used for filtering segments.	null
advertiser_id	int	The ID of the advertiser using the segment if the segment should be on the Advertiser level rather than the Network level. If <code>null</code> , the segment will be usable by all advertisers for that member. This information is for reference in Console.	null

piggyback_pixels	array	<p>The URLs of the pixels you want us to fire when the segment pixel fires. See Piggyback Pixels below for more details.</p>
parent_segment_id	int	<p>The ID of the parent segment. If the parent segment is targeted in a profile, the direct child segments are targeted as well.</p> <div data-bbox="831 464 1425 600" style="border: 1px solid #f0e68c; padding: 5px; margin: 10px 0;"> <p>The <code>parent_segment_id</code> field will be deprecated on January 1, 2019</p> </div> <div data-bbox="831 632 1425 1041" style="border: 1px solid #d9d9d9; padding: 5px; margin: 10px 0;"> <p>The parent-child logic extends only one level deep. Example: Segment A is the parent of segment B and segment C, and segment C is the parent of segment D. When segment A is targeted, segment B and segment C are targeted as well, but segment D is not.</p> </div>
querystring_mapping	object	<p>A query string that allows you to assign a set of parameters and values to a segment. See About Query Strings for a general description of query strings and Query String Mapping for more details.</p> <div data-bbox="831 1346 1425 1482" style="border: 1px solid #f0e68c; padding: 5px; margin: 10px 0;"> <p>Invalid if a <code>querystring_mapping_key_value</code> object is also included.</p> </div>

<code>querystring_mapping_key_value</code>	object	<p>A query string that allows you to reuse a key and assign a single key-value pair to a segment. See About Query Strings for a general description of query strings and Query String Mapping Key Value for more details.</p> <div data-bbox="829 390 1425 527" style="border: 1px solid #f0e68c; padding: 10px; margin-bottom: 10px;"> <p>Invalid if a <code>querystring_mapping</code> object is also included.</p> </div> <div data-bbox="829 558 1425 785" style="border: 1px solid #f0e68c; padding: 10px;"> <p>The value of the <code>querystring_mapping_key_value</code> field is case-insensitive when it is looked up in an auction.</p> </div>
--------------------------------------------	--------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Piggyback Pixels

When adding piggyback pixels, please keep the following in mind:

- Image pixels can only piggyback off other other image pixels, and JavaScript pixels can only piggyback other JavaScript pixels.
- Image pixels can only have one piggyback pixel. If you need to piggyback multiple pixels, be sure to use a JavaScript pixel.
- There are no character limits to piggybacked pixels in AppNexus, but browser/server URL limits may apply.

Each object in the `piggyback_pixels` array contains the following fields.

Field	Type	Description
<code>url</code>	string	The URL of the pixel to piggyback.
<code>pixel_type</code>	enum	The type of pixel to piggyback. Possible values: "js" or "img".

About Query Strings

AppNexus provides two methods of query string mapping to allow publishers to pass inventory-specific or user-specific information in the query strings of their placement tags: **query string mapping**, in which a set of parameter values assigned to the segment; and **query string mapping key value**, which allows you to assign one key/value pair to a segment and then reuse the same key with a different value in another segment.

For both types of mapped query strings:

- Whenever an ad call is made with a mapped query string, the associated user will automatically be added to the segment.
- You can target the query string in a campaign (via the `segment_targets` or `segment_group_targets` fields of the [Profile Service](#)).

Note that you can choose how long the user will be kept in the segment. If you want to keep the user in the segment for retargeting purposes, set `expire_minutes` to the correct number of minutes (`null` will set to system maximum value 180 days). If you want to add the user to the segment only for the duration of the ad call, set `expire_minutes` to 0.

See [Examples](#) below for various scenarios and formatting.

Query String Mapping

Query string mapping allows you to assign a parameter to a segment using the `querystring_mapping` field. Multiple values can be added to a parameter. This method is useful in cases where only one value makes sense for a user: for example, a user's age, salary range, or experience. If a user is in the `beginner` segment but later gains more experience, the user can be moved to the `advanced` segment and is automatically removed from the `beginner` segment. See [this example](#) for more details.

In this type of query string, the parameter can be provided with no values. For example, `{"param": "car_model", "value_type": "none"}`. This allows any value for that parameter to be provided.

Field	Type	Description
<code>param</code>	string	The query string parameter.
<code>value_type</code>	enum	The type of value accompanying the parameter. Possible values: "text", "numeric", or "none".
<code>values</code>	array	The strings that can accompany the parameter when <code>value_type</code> is "text". If <code>value_type</code> is "numeric" or "none", this field cannot be used. <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"><p>Need the Value IDs?</p><p>If you need the IDs of the values, pass the query string parameters <code>show_querystring_ids=true</code>, and <code>values</code> will instead be an array of objects with the keys "value" (string) and "id" (int).</p></div>
<code>allow_empty_text</code>	Boolean	When <code>true</code> , the <code>values</code> array may be null. May only be used when the <code>value_type</code> is "text". Defaults to <code>false</code> .
<code>publishers</code>	array	The publishers from which you expect the query string. Users associated with these publishers' placements will be added to the segment.

sites	array	The placement groups (sites) from which you expect the query string. Users associated with these placements will be added to the segment. Note that this field overrides publishers; if you specify a site that doesn't belong to one of the specified publishers, users associated with the placements in a placement group will nonetheless be added to the segment.
placements	array	The placements in which you expect the query string. Users associated with these placements will be added to the segment. Note that this field overrides placement groups and publishers. That is, if you specify a placement that doesn't belong to one of the specified placement groups or publishers, users associated with the placement will still be added to the segment.
include_shared	Boolean	Set this value to false to avoid retrieving shared segments.

Query String Mapping Key Value

Query string key/value mapping allows you to assign a single key-value pair to a segment using the `querystring_mapping_key_value` object. This type of query string allows a user to exist in multiple segments and is useful in cases where more than one value for a parameter makes sense: for example, a user's musical preferences. If a user is in the `rock` segment but also likes `funk`, he or she can exist in both segments simultaneously. See [the key-value mapping example](#) for more details.

In this type of query string, each `qs_key` must have at least one corresponding `qs_value`.

Field	Type	Description
<code>qs_key</code>	string	The query string parameter.
<code>qs_value</code>	string	A value for the query string key. The value can be empty or null. Multiple values can be added using the same key. A <code>qs_value</code> must be provided if a <code>qs_key</code> is used.

Examples

>> [Create a segment](#)

```

$ cat segment

{
  "segment": {
    "member_id": 25,
    "short_name": "remarketing - netflix user",
    "code": "netflix08",
    "price": 0.12
  }
}

$ curl -b cookies -c cookies -X POST -d @segment 'https://api.appnexus.com/segment'

{
  "response": {
    "status": "OK",
    "id": "5005"
  }
}

```

>> View a segment

```

$ curl -b cookies -c cookies 'https://api.appnexus.com/segment?id=11836'

{
  "response": {
    "status": "OK",
    "segments": [
      {
        "id": 11836,
        "code": null,
        "state": "active",
        "short_name": "March 10",
        "description": null,
        "member_id": 185,
        "category": null,
        "price": "0",
        "expire_minutes": null,
        "enable_rm_piggyback": true,
        "max_usersync_pixels": 0,
        "last_modified": "2010-03-10 23:23:48",
        "provider": null,
        "parent_segment_id": null,
        "advertiser_id": 51,
        "piggyback_pixels": null
      }
    ]
  }
}

```

>> Change a segment from advertiser-owned to network-owned

To change an advertiser-level segment into a network-level segment, you make a PUT call, passing the segment ID and member ID in the query string and setting `advertiser_id` to null in the JSON file.

```

$ cat segment_update

{
  "segment": {
    "advertiser_id": null
  }
}

$ curl -b cookies -c cookies -X PUT -d @segment_update
'https://api.appnexus.com/segment?id=175196&member_id=1066'

{
  "response": {
    "status": "OK",
    "count": 1,
    "id": "175196",
    "start_element": 0,
    "num_elements": 100,
    "segment": {
      "id": 175196,
      "code": null,
      "state": "active",
      "short_name": "Users who have purchased software",
      "description": null,
      "member_id": 1066,
      "category": null,
      "price": 0,
      "expire_minutes": null,
      "enable_rm_piggyback": true,
      "max_usersync_pixels": null,
      "advertiser_id": null,
      "last_modified": "2013-10-18 20:34:27",
      "provider": null,
      "parent_segment_id": null,
      "piggyback_pixels": null,
      "querystring_mapping": null
    }
  }
}

```

>> Add text query string mapping to a segment

Scenario: Publisher 6 tells you to expect one of the following values in the query string of placement 596411: "experience_level=beginner", "experience_level=intermediate", or "experience_level=advanced". When this placement comes in with the "experience_level" parameter, you want to add the associated user to segment 25 with the ability to distinguish between the values of beginner, intermediate, and advanced in campaign targeting. In this case, you would create the following JSON and make a PUT call to update the segment.

```

$ cat segment_update

{
  "segments": [
    {
      "id": 25,
      "member_id": 20,
      "querystring_mapping": {

```

```

        "param": "experience_level",
        "value_type": "text",
        "values": [
            "beginner",
            "intermediate",
            "advanced"
        ],
        "publishers": [
            {
                "id": 6
            }
        ],
        "placements": [
            {
                "id": 596411
            }
        ]
    }
}
]
}

```

```

$ curl -b cookies -c cookies -X PUT -d @segment_update
'https://api.appnexus.com/segment?id=25'

```

```

{
  "response": {
    "status": "OK",
    "count": 1,
    "id": "25",
    "start_element": null,
    "num_elements": null,
    "segment": {
      "id": 25,
      "code": null,
      "state": "active",
      "short_name": "test",
      "description": null,
      "member_id": 49,
      "category": null,
      "price": 23,
      "expire_minutes": 20,
      "enable_rm_piggyback": true,
      "max_usersync_pixels": null,
      "last_modified": "2012-01-24 20:15:18",
      "provider": null,
      "parent_segment_id": null,
      "advertiser_id": null,
      "piggyback_pixels": null,
      "querystring_mapping": {
        "param": "experience_level",
        "value_type": "text",
        "values": [
            "beginner",
            "intermediate",
            "advanced"
        ],
        "publishers": [
            {

```

```
        "id": 6,  
        "name": "Publisher 6"  
    }  
],  
"sites": null,  
"placements": [  
    {  
        "id": 596411,  
        "name": "Placement 596411"  
    }  
]  
}
```

```
}  
}  
}
```

>> Add numeric query string mapping to a segment

Scenario: Auto publisher 30 tells you to expect the "car_year" parameter followed by a year in the query strings of its placements. When a placement from this publisher comes in with "car_year=YYYY" in its query string, you want to add the associated user to segment 26. In this case, you would create the following JSON and make a PUT call to update the segment.

```
$ cat segment_update  
  
{  
  "segment": {  
    "member_id": 20,  
    "querystring_mapping": {  
      "param": "car_year",  
      "value_type": "numeric",  
      "publishers": [  
        {  
          "id": 30  
        }  
      ]  
    }  
  }  
}  
  
$ curl -b cookies -c cookies -X PUT -d @segment  
'https://api.appnexus.com/segment?id=26'  
  
{  
  "response": {  
    "status": "OK",  
    "count": 1,  
    "id": "26",  
    "start_element": null,  
    "num_elements": null,  
    "segment": {  
      "id": 26,  
      "code": null,  
      "state": "active",  
      "short_name": "test",  
      "description": null,  
      "member_id": 20,  
      "category": null,  
      "price": 23,  
      "expire_minutes": 20,  
      "enable_rm_piggyback": true,  
      "max_usersync_pixels": null,  
      "last_modified": "2012-01-24 20:15:18",  
      "provider": null,  
      "parent_segment_id": null,  
      "advertiser_id": null,  
      "piggyback_pixels": null,  
      "querystring_mapping": {  
        "param": "car_year",
```

```
"value_type": "numeric",
"values": null,
"publishers": [
  {
    "id": 30,
    "name": "Publisher 30"
  }
],
"sites": null,
"placements": null
}
```

```
}  
}  
}
```

>> Add query string mapping without values to a segment

Scenario: Both publishers 10 and 30 tell you to expect the parameter "car_model" (without values) in their placements. When a placement from either of these publishers comes in with "car_model" in its query string, you want to add the associated user to segment 2. In this case, you would create the following JSON and make a PUT call to update the segment.

```
$ cat segment_update  
  
{  
  "segment": {  
    "member_id": 20,  
    "querystring_mapping": {  
      "param": "car_model",  
      "value_type": "none",  
      "publishers": [  
        {  
          "id": 10,  
          "id": 30  
        }  
      ]  
    }  
  }  
}  
  
$ curl -b cookies -c cookies -X PUT -d @segment_update  
'https://api.appnexus.com/segment?id=2'  
  
{  
  "response": {  
    "status": "OK",  
    "count": 1,  
    "id": "2",  
    "start_element": null,  
    "num_elements": null,  
    "segment": {  
      "id": 2,  
      "code": null,  
      "state": "active",  
      "short_name": "test",  
      "description": null,  
      "member_id": 20,  
      "category": null,  
      "price": 23,  
      "expire_minutes": 20,  
      "enable_rm_piggyback": true,  
      "max_usersync_pixels": null,  
      "last_modified": "2012-01-24 20:15:18",  
      "provider": null,  
      "parent_segment_id": null,  
      "advertiser_id": null,  
      "piggyback_pixels": null,  
      "querystring_mapping": {
```

```
"param": "car_model",
"value_type": "none",
"values": null,
"publishers": [
  {
    "id": 10,
    "name": "Publisher 10"
  },
  {
    "id": 30
    "name": "Publisher 30"
  }
],
"sites": null,
"placements": null
}
```

```
}
}
}
```

>> Use query string key-value mapping to reuse a key with multiple values

Scenario: Publishers want to add users to segments based on their musical preferences. You want to use the same key, "music_genre" and offer multiple values: "rock", "pop", "hard rock", and "funk", assigning them to different segments and allow users to be placed in one or more of the segments. This reflects the fact that users may have diverse musical choices.

In this case, you would create the following JSON and make a PUT call to update the segments.

```
$ cat segment_update
{
  "segments": [
    {
      "id": 501,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "rock"
      }
    },
    {
      "id": 502,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "pop"
      }
    },
    {
      "id": 503,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "hard rock"
      }
    },
    {
      "id": 504,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "funk"
      }
    }
  ]
}

$ curl -b cookies -c cookies -X PUT -d @segment_update
'https://api.appnexus.com/segment?member_id=201'

{
  "response": {
    "status": "OK",
    "count": 2,
    "id": [
      501,
      502,
    ]
  }
}
```

```
    503,
    504
  ],
  "start_element": 0,
  "num_elements": 100,
  "ids": [
    501,
    502,
    503,
    504
  ],
  "segments": {
    "501": {
      "id": 501,
      "code": null,
      "state": "active",
      "short_name": "rock",
      "description": null,
      "member_id": 201,
      "category": null,
      "price": 0,
      "expire_minutes": null,
      "enable_rm_piggyback": true,
      "max_usersync_pixels": null,
      "advertiser_id": null,
      "last_modified": "2015-06-02 16:39:40",
      "provider": null,
      "parent_segment_id": null,
      "piggyback_pixels": null,
      "querystring_mapping": null,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "rock"
      }
    },
    "502": {
      "id": 502,
      "code": null,
      "state": "active",
      "short_name": "rock",
      "description": null,
      "member_id": 201,
      "category": null,
      "price": 0,
      "expire_minutes": null,
      "enable_rm_piggyback": true,
      "max_usersync_pixels": null,
      "advertiser_id": null,
      "last_modified": "2015-06-02 16:39:40",
      "provider": null,
      "parent_segment_id": null,
      "piggyback_pixels": null,
      "querystring_mapping": null,
      "querystring_mapping_key_value": {
        "qs_key": "music_genre",
        "qs_value": "pop"
      }
    },
    "503": {
```

```
    "id": 503,
    "code": null,
    "state": "active",
    "short_name": "rock",
    "description": null,
    "member_id": 201,
    "category": null,
    "price": 0,
    "expire_minutes": null,
    "enable_rm_piggyback": true,
    "max_usersync_pixels": null,
    "advertiser_id": null,
    "last_modified": "2015-06-02 16:39:40",
    "provider": null,
    "parent_segment_id": null,
    "piggyback_pixels": null,
    "querystring_mapping": null,
    "querystring_mapping_key_value": {
      "qs_key": "music_genre",
      "qs_value": "hard rock"
    }
  },
  "504": {
    "id": 504,
    "code": null,
    "state": "active",
    "short_name": null,
    "description": null,
    "member_id": 958,
    "category": null,
    "price": 0,
    "expire_minutes": null,
    "enable_rm_piggyback": true,
    "max_usersync_pixels": null,
    "advertiser_id": null,
    "last_modified": "2015-06-02 16:39:40",
    "provider": null,
    "parent_segment_id": null,
    "piggyback_pixels": null,
    "querystring_mapping": null,
    "querystring_mapping_key_value": {
      "qs_key": "music_genre",
      "qs_value": "funk"
    }
  }
}
```

