

# 1 Campaign API Reference

The following operations are available via the API:

1. Fetch a List of Campaigns - Fetch a list of Campaigns associated with a specific partner account.
2. Fetch a Single Campaign - Fetch a single Campaign resource using the Campaign ID.
3. Create a Campaign - Create a new Campaign under a specific partner account.
4. Update a Campaign - Update an existing Campaign of a specific partner account.

## 1.1 Cheat Sheet



This is merely a quick reference of the available end-points. For further information please continue to the sections below.

Description	Method	Path (prefix: /backstage/api/1.0)
Fetch a list of campaigns associated with an account <a href="#">More info</a>	GET	/[account-id]/campaigns/
Fetch a specific campaign associated with an account <a href="#">More info</a>	GET	/[account-id]/campaigns/[campaign-id]/
Create a new campaign for an account <a href="#">More info</a>	POST	/[account-id]/campaigns/
Update an existing campaign <a href="#">More info</a>	POST, PUT	/[account-id]/campaigns/[campaign-id]/



Reading, creating or updating Campaigns is only possible when the [account-id] references an Account with **ADVERTISER** in its **partner\_types** field. Trying to send a request with a non-Advertiser Account will result in a **404 Not Found** response.

## 1.2 Campaign Resource

### 1.2.1 Fields



**Required** fields are fields that must be sent to the server when creating a new resource.

**Read-only** fields are fields that should never be sent to the server, and will appear only when *fetching* a resource.

**Final** fields are set once, when creating the resource, and become *read-only* afterwards.

Name	Modifier & Defaults	Type	Description
id	Read-only	String	Unique numeric ID of a Campaign (returned as a string)
advertiser_id	Read-only	String	The <code>account_id</code> of the Campaign's Advertiser Account
name	Required	String Max length: 200	Human-readable name for the Campaign
branding_text	Required	String Max length: 25	This text will appear below the title of each of your items
tracking_code	Default: "utm_source=taboola&utm_medium=referral"	String Max length: 255	Query-string parameter which will be added to the URL of your items, so you can track the traffic generated from Taboola.
cpc	Required	Double Must be in the allowed range of the resource's account.	Cost-per-click
daily_cap	Default: 0.0	Double Must be lower than the <code>spending_limit</code>	

		0.0 means "unlimited"							
daily_ad_delivery_model	<p>Default:</p> <p>Depending on daily_cap field (for backward compatibility):</p> <p>If daily_cap exist and &gt; 0 - this field defaults to "STRICT"</p> <p>Otherwise it defaults to "ACCELERATED"</p>	<p>String</p> <p>Possible values:</p> <table><tr><td>BALANCED</td><td>The daily cap will be balanced each day of the month according to the monthly budget. Daily Cap must be empty.</td></tr><tr><td>ACCELERATED</td><td>The campaign will be delivered without a daily cap. Same as setting daily_cap to 0.0.</td></tr><tr><td>STRICT</td><td>The campaign's daily cap will be a strict one determined by the daily_cap field. Same as setting daily_cap higher than 0.0.</td></tr></table>	BALANCED	The daily cap will be balanced each day of the month according to the monthly budget. Daily Cap must be empty.	ACCELERATED	The campaign will be delivered without a daily cap. Same as setting daily_cap to 0.0.	STRICT	The campaign's daily cap will be a strict one determined by the daily_cap field. Same as setting daily_cap higher than 0.0.	The Daily Ad Delivery Model (combined with the Daily Cap) defines the way the campaign will be delivered on a daily basis. Each type represents a different approach to how the campaign budget will be spent.
BALANCED	The daily cap will be balanced each day of the month according to the monthly budget. Daily Cap must be empty.								
ACCELERATED	The campaign will be delivered without a daily cap. Same as setting daily_cap to 0.0.								
STRICT	The campaign's daily cap will be a strict one determined by the daily_cap field. Same as setting daily_cap higher than 0.0.								
publisher_bid_modifier	<p>Default:</p> <p><a href="#">Publisher Bid Modifier Object</a> with values=[]</p>	<a href="#">Publisher Bid Modifier Object</a>	An object representing the wanted publisher bid modifiers for this campaign						
spending_limit	Required	Double Must be higher than the following fields: cpc, daily_cap	The maximum amount of money a Campaign can consume.						

spending_limit_model	Required	String Most updated possible values are accessible via dictionary. Here's the current ones: <table><tr><td>MONTHLY</td><td>Monthly recurring budget</td></tr><tr><td>ENTIRE</td><td>Non recurring budget</td></tr></table>	MONTHLY	Monthly recurring budget	ENTIRE	Non recurring budget	The Spending Limit Model defines whether the campaign will have a monthly recurring budget (which replenishes automatically each month) or an overall budget.
MONTHLY	Monthly recurring budget						
ENTIRE	Non recurring budget						
country_targeting	Default: <a href="#">Targeting Object</a> with type=ALL	<a href="#">Targeting Object</a> NULL means run on all countries  Possible targeting values: 2-letter country code as defined by <a href="#">ISO-3166</a> ; Most updated possible values are accessible via dictionary.	List of courtiers the campaign will target or exclude.				
sub_country_targeting	Default: <a href="#">Targeting Object</a> with type=ALL	<a href="#">Targeting Object</a> NULL means run on all regions / DMA's.  This feature can only be used if only 1 country is used in country targeting in 'INCLUDE' mode, and the region or DMA codes must be relevant to that country. In addition, this can be used for region targeting or DMA targeting (DMA only in US) but not both at the same time.  Possible targeting values: Most updated possible values are accessible via dictionary.	List of regions or DMA's the campaign will target or exclude.				
postal_code_targeting	Read-only Default: <a href="#">Targeting Object</a> with type=ALL	<a href="#">Targeting Object</a> NULL means run on all postal codes.	In case of non default value, holds a reference to full postal code targeting details.				
platform_targeting	Default: <a href="#">Targeting Object</a> with type=ALL	<a href="#">Targeting Object</a> (INCLUDE only)	List of platform device types (desktop /				

		<p>NULL means to run on all platforms</p> <p>Most updated possible values are accessible via dictionary. Here's the current ones::</p> <table><tr><td>DESK</td><td>Desktop</td></tr><tr><td>PHON</td><td>Smartphone</td></tr><tr><td>TBLT</td><td>Tablet</td></tr></table>	DESK	Desktop	PHON	Smartphone	TBLT	Tablet	<p>tablet / smartphone) to run the campaign on.</p> <p>NULL means to run on all platform device types.</p>
DESK	Desktop								
PHON	Smartphone								
TBLT	Tablet								
publisher_targeting	<p>Default: <a href="#">Targeting Object</a> with type=ALL</p>	<p><a href="#">Targeting Object</a> (EXCLUDE only)</p> <p>NULL means to run on all publishers</p> <p>Possible targeting values: account_id of various Accounts</p>	<p>List of Partner Accounts' account_id to blacklist from publishing this Campaign. The account_id must reference an Account with type=PARTNER.</p>						
audience_segments_multi_targeting	<p>Read-only</p> <p>Default: <a href="#">Multi Targeting Object</a> with state=ALL</p>	<p><a href="#">Multi Targeting Object</a></p>	<p>In case of EXISTS state, holds a reference to full audience segments targeting details.</p>						
comments	<p>Default: ""</p> <p>(empty string)</p>	<p>String</p> <p>Max length: 1000</p>	<p>Campaign comments</p>						
start_date	<p>Final</p> <p>Default: &lt;Now&gt;</p> <p>OR:</p> <p>NULL (for non-privileged users)</p>	<p>Date (ISO 8601)</p> <p>Must be today or later.</p>							
end_date	<p>Default: "9999-12-31"</p>	<p>Date (ISO 8601)</p> <p>Must be later than start_date</p> <p>Can be modified only if status!=EXPIRED</p>							

approval_state	Read-only Default: PENDING	String Requires certain permissions to modify  Possible values: <table><tr><td>APPROVED</td></tr><tr><td>REJECTED</td></tr><tr><td>PENDING</td></tr></table>	APPROVED	REJECTED	PENDING	<p>This field determines whether the Campaign is approved to be served. The Campaign can only be served if its approval_state is APPROVED.</p> <p>This field affects the status of the Campaign.</p>					
APPROVED											
REJECTED											
PENDING											
is_active	Default: true	Boolean	<p>If true, Campaign is active and may serve ads if its status is RUNNING.</p> <p>If false, Campaign is currently not serving ads, and its status will be PAUSED</p>								
spent	Read-only	Double	Represents the estimated amount of money this Campaign has consumed								
status	Read-only	String Most updated possible values are accessible via dictionary. Here are the current ones: <table><tr><td>RUNNING</td><td>Campaign is currently serving ads</td></tr><tr><td>PAUSED</td><td>Campaign is currently not serving ads</td></tr><tr><td>PENDING_START_DATE</td><td>Campaign is pending its start_date</td></tr><tr><td>DEPLETED_MONTHLY</td><td>Campaign has depleted its MONTHLY spending_limit</td></tr></table>		RUNNING	Campaign is currently serving ads	PAUSED	Campaign is currently not serving ads	PENDING_START_DATE	Campaign is pending its start_date	DEPLETED_MONTHLY	Campaign has depleted its MONTHLY spending_limit
RUNNING	Campaign is currently serving ads										
PAUSED	Campaign is currently not serving ads										
PENDING_START_DATE	Campaign is pending its start_date										
DEPLETED_MONTHLY	Campaign has depleted its MONTHLY spending_limit										

		<table><tr><td>DEPLETED</td><td>Campaign has depleted its ENTIRE spending_limit</td></tr><tr><td>EXPIRED</td><td>Campaign's end_date has passed</td></tr><tr><td>TERMINATED</td><td>Campaign was manually terminated</td></tr><tr><td>FROZEN</td><td>Campaign's Advertiser account is Frozen</td></tr><tr><td>PENDING_APPROVAL</td><td>Campaign has approval_status set to PENDING</td></tr><tr><td>REJECTED</td><td>Campaign has approval_status set to REJECTED</td></tr></table>	DEPLETED	Campaign has depleted its ENTIRE spending_limit	EXPIRED	Campaign's end_date has passed	TERMINATED	Campaign was manually terminated	FROZEN	Campaign's Advertiser account is Frozen	PENDING_APPROVAL	Campaign has approval_status set to PENDING	REJECTED	Campaign has approval_status set to REJECTED
DEPLETED	Campaign has depleted its ENTIRE spending_limit													
EXPIRED	Campaign's end_date has passed													
TERMINATED	Campaign was manually terminated													
FROZEN	Campaign's Advertiser account is Frozen													
PENDING_APPROVAL	Campaign has approval_status set to PENDING													
REJECTED	Campaign has approval_status set to REJECTED													
traffic_allocation_mode	Default OPTIMIZED	<div>String Max length: 20</div> <table><tr><th>value</th><th>description</th></tr><tr><td>OPTIMIZED</td><td>(Recommended) The exposure of the creatives in your campaign will be determined by our algorithm, and your most engaging creatives will be served more than others.</td></tr><tr><td>EVEN</td><td>the creatives of your campaign will be exposed evenly, based on the traffic that is being allocated for your campaign.</td></tr></table>	value	description	OPTIMIZED	(Recommended) The exposure of the creatives in your campaign will be determined by our algorithm, and your most engaging creatives will be served more than others.	EVEN	the creatives of your campaign will be exposed evenly, based on the traffic that is being allocated for your campaign.						
value	description													
OPTIMIZED	(Recommended) The exposure of the creatives in your campaign will be determined by our algorithm, and your most engaging creatives will be served more than others.													
EVEN	the creatives of your campaign will be exposed evenly, based on the traffic that is being allocated for your campaign.													



Trying to set field values which do not adhere to their restrictions, will result in a **400 Bad Request** error response.

### 1.2.2 Campaign Spending Models

Generally speaking, Campaigns have two possible "Spending Models":

- **Entire Budget** - The `spending_limit` field of the Campaign represents the entire amount of money this Campaign can consume. This means that once it depletes this budget, the Campaign will stop running, and its `status` will be `DEPLETED`.
- **Monthly Budget** - The `spending_limit` field of the Campaign represents the amount of money this Campaign can consume *each month*. This means that once a Campaign depletes its budget, the Campaign will stop running, and its `status` will be `MONTHLY_DEPLETED`. Once a new month begins, the Campaign's budget will replenish and the Campaign will start running again.

The Spending Model type of a certain Campaign is determined by setting the value of its `spending_limit_model` field. Setting the field to `ENTIRE` will set the Spending Model of the Campaign to "Entire Budget", while `MONTHLY` will set it to "Monthly Budget".

### 1.2.3 Publisher Bid Modifier Object

#### 1.2.4 Fields

Name	Modifier & Defaults	Type	Description
values	<b>Default:</b> []	Array< <a href="#">Targeted Bid Modifier Object</a> > NULL means no change to targeted bid modifiers [] means no targeted bid modifiers	An object representing the targeted publishers bid modifiers

### 1.2.5 Targeted Bid Modifier Object

#### 1.2.6 Fields

Name	Modifier & Defaults	Type	Description
target	<b>Required</b>	String The name of the target to associated this bid boost modifier	When using in context of " <a href="#">Publisher Bid Modifier Object</a> ",



			target is publisher name
cpc_modification	Required	Double Must be between values 0.5 - 1.5	

## 1.3 Targeting

For more details about Campaign targeting see Backstage API Targeting documentation.

## 1.4 Reading

Reading can be performed in two ways:

1. Fetch a list of Campaigns
2. Fetch a single Campaign

### 1.4.1 Fetching a List

#### Request

To retrieve a list of an Account's Campaigns, send an HTTP *GET* request in the following form:

```
GET /backstage/api/1.0/[account-id]/campaigns/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```

#### Example Response:

200 OK

```
{
  "results": [
    {
      "id" : "1",
      "advertiser_id" : "taboola-demo-advertiser",
      "name" : "Demo Campaign",
      "tracking_code" : "taboola-track",
      "cpc" : 0.25,
      "daily_cap" : 100,
      "spending_limit" : 1000,
      "spending_limit_model": "MONTHLY",
      "country_targeting": {
        "type" : "INCLUDE",
        "value" : ["AU", "GB"]
      },
      "sub_country_targeting": null,
      "postal_code_targeting": null,
      "platform_targeting": {
```

```

        "type" : "INCLUDE",
        "value" : ["TBLT", "PHON"]
    },
    "publisher_targeting": null,
    "comments" : null,
    "start_date" : "2015-04-30",
    "end_date" : "2015-05-30",
    "approval_state" : "APPROVED",
    "is_active" : true,
    "spent" : 2.23,
    "status" : "RUNNING",
    "daily_ad_delivery_model" : "STRICT",
    "traffic_allocation_mode" : "EVEN",
    "publisher_bid_modifier" : {
        "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
                    {"target" : "publisher2", "cpc_modification" : 0.9}
        ]
    },
    "audience_segments_multi_targeting": {
        "state" : "EXISTS",
        "href" : "http://backstage/api/1.0/[account-id]/campaigns/[campaign-id]
                /targeting/audience_segments"
    }
},
// ... more results
]
}

```

## 1.4.2 Fetching a Specific Instance

### Request

To retrieve a specific Campaign, send an HTTP *GET* request in the following form:

```

GET /backstage/api/1.0/[account-id]/campaigns/[campaign-id]/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]

```

The response will include a valid JSON object in its body representing the Campaign resource.

### Example Response:

```

200 OK

{
  "id": "1",
  "advertiser_id" : "taboola-demo-advertiser",
  "branding_text": "Pizza",
  "name": "Demo Campaign",
  "tracking_code": "taboola-track",

```

```

    "cpc": 0.25,
    "daily_cap": 100,
    "spending_limit": 1000,
    "spending_limit_model": "MONTHLY",
    "country_targeting": {
      "type" : "INCLUDE",
      "value" : ["AU", "GB"]
    },
    "sub_country_targeting": null,
    "postal_code_targeting": null,
    "platform_targeting": {
      "type" : "INCLUDE",
      "value" : ["TBLT", "PHON"]
    },
    "publisher_targeting": null,
    "comments" : null,
    "start_date": "2015-04-30",
    "end_date": "2015-05-30",
    "approval_state" : "APPROVED",
    "is_active": true,
    "spent": 0.0,
    "status": "RUNNING",
    "daily_ad_delivery_model" : "STRICT",
    "traffic_allocation_mode" : "EVEN",
    "publisher_bid_modifier" : {
      "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
                 {"target" : "publisher2", "cpc_modification" : 0.9}
            ]
    }
  }
}

```

## 1.5 Creating

### 1.5.1 Request Structure

Creating a new Campaign is performed by sending an HTTP *POST* request to a URL in the following form:

```

POST /backstage/api/1.0/[account-id]/campaigns/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]

```

The request must include a JSON object representing the Campaign in its body.



The body of the request must contain *at least* all of the *required* fields. An incomplete set of fields will result in a **400 Bad Request** error response.

The response for such a request will be **200 OK** and will include the newly created campaign as a valid JSON object in its body.

## 1.5.2 Example: Only Required Fields

**POST** /backstage/api/1.0/taboola-demo-advertiser/campaigns/

Host: https://backstage.taboola.com

Authorization: Bearer [access-token]

```
{
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "cpc": 0.25,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY"
}
```

### Response:

200 OK

```
{
  "id": "123",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "tracking_code": "",
  "cpc": 0.25,
  "daily_cap": null,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": null,
  "sub_country_targeting": null,
  "postal_code_targeting": null,
  "platform_targeting": null,
  "publisher_targeting": null,
  "comments" : null,
  "start_date": null,
  "end_date": null,
  "approval_state" : "APPROVED",
  "is_active": true,
  "spent": 0.0,
  "status": "RUNNING",
  "daily_ad_delivery_model" : "ACCELERATED",
  "traffic_allocation_mode" : "EVEN",
  "publisher_bid_modifier": {
    "values": []
  }
}
```



The response includes the read-only fields `id` and `status`.

### 1.5.3 Example: Full Set of Fields

POST /backstage/api/1.0/taboola-demo-advertiser/campaigns/

Host: https://backstage.taboola.com

Authorization: Bearer [access-token]

```
{
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "tracking_code": "taboola-track",
  "cpc": 0.25,
  "daily_cap": 100,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
  "sub_country_targeting": null,
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": null,
  "daily_ad_delivery_model" : "STRICT",
  "publisher_bid_modifier" : {
    "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
               {"target" : "publisher2", "cpc_modification" : 0.9}
    ]
  },
  "comments" : null,
  "start_date": "2015-04-30",
  "end_date": "2015-05-30",
  "approval_state" : "APPROVED",
  "is_active": true
}
```

#### Response:

200 OK

```
{
  "id": "124",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "tracking_code": "taboola-track",
  "cpc": 0.25,
  "daily_cap": 100,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
}
```

```

"sub_country_targeting": null,
"postal_code_targeting": null,
"platform_targeting": {
  "type" : "INCLUDE",
  "value" : ["TBLT", "PHON"]
},
"publisher_targeting": null,
"comments" : null,
"start_date": "2015-04-30",
"end_date": "2015-05-30",
"approval_state" : "APPROVED",
"is_active": true,
"spent": 0.0,
"status": "RUNNING",
"daily_ad_delivery_model" : "STRICT",
"traffic_allocation_mode" : "EVEN",
"publisher_bid_modifier" : {
  "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
             {"target" : "publisher2", "cpc_modification" : 0.9}
            ]
},
"audience_segments_multi_targeting": {
  "state" : "EXISTS",
  "href" : "http://backstage/api/1.0/[account-id]/campaigns/[campaign-id]
           /targeting/audience_segments"
}
}

```

### 1.5.4 Example: Insufficient Set of Fields

POST /backstage/api/1.0/**taboola-demo-advertiser**/campaigns/  
Host: https://backstage.taboola.com  
Authorization: Bearer [**access-token**]

```

{
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
}

```

#### Response:

**400** Bad Request

```

{
  "http_status": 400,
  "message": "\"cpc\" field is missing."
}

```

## 1.6 Updating

Updating is performed by sending an HTTP *POST* or *PUT* request, which includes a JSON object representing the fields to update in its body:

```
POST /backstage/api/1.0/[account-id]/campaigns/[campaign-id]/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```



When updating resources, it is possible to use either the *POST* method, or the *PUT* method. The API will respect both, but the documentation only uses the *POST* approach in examples.



An update request may include only a subset of the resource's fields. Included fields with *NULL* value will be treated as if they are missing from the request.  
*Fields which are missing or set to null in the request will not be modified.*

### 1.6.1 Example: Update a Single Field

```
POST /backstage/api/1.0/taboola-demo-advertiser/campaigns/124/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```

```
{
  "name": "Demo Campaign - Edited"
}
```

#### Response:

```
200 OK
```

```

{
  "id": "124",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign - Edited",
  "branding_text": "Pizza",
  "tracking_code": "taboola-track",
  "cpc": 0.25,
  "daily_cap": 100,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
  "sub_country_targeting": null,
  "postal_code_targeting": null,
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": null,
  "comments" : null,
  "start_date": "2015-04-30",
  "end_date": "2015-05-30",
  "approval_state" : "APPROVED",
  "is_active": true,
  "spent": 0.0,
  "status": "RUNNING"
  "daily_ad_delivery_model" : "STRICT",
  "traffic_allocation_mode" : "EVEN",
  "publisher_bid_modifier" : {
    "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
               {"target" : "publisher2", "cpc_modification" : 0.9}
    ]
  }
}

```

### 1.6.2 Example: Updating Multiple Fields

```

POST /backstage/api/1.0/taboola-demo-advertiser/campaigns/124
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]

```



```
{
  "name": "Demo Campaign",
  "branding_text": "Yummy Pizza",
  "tracking_code": "taboola-tracking-code",
  "cpc": 0.5,
  "daily_cap": 500,
  "spending_limit": 5000,
  "spending_limit_model": "ENTIRE",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
  "sub_country_targeting": null,
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": null,
  "publisher_bid_modifier" : {
    "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
               {"target" : "publisher2", "cpc_modification" : 0.9}
    ]
  },
  "comments" : null,
  "start_date": "2015-04-24",
  "end_date": "2015-05-24"
}
```

## Response:

200 OK

```
{
  "id": "124",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Yummy Pizza",
  "tracking_code": "taboola-tracking-code",
  "cpc": 0.5,
  "daily_cap": 500,
  "spending_limit": 5000,
  "spending_limit_model": "ENTIRE",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
  "sub_country_targeting": null,
  "postal_code_targeting": null,
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": null,
  "comments" : null,
  "start_date": "2015-04-24",
  "end_date": "2015-05-24",
  "approval_state" : "APPROVED",
  "is_active": true,
  "spent": 0.0,
  "status": "RUNNING",
  "daily_ad_delivery_model" : "STRICT",
  "traffic_allocation_mode" : "EVEN",
  "publisher_bid_modifier" : {
    "values": [{"target" : "publisher1", "cpc_modification" : 1.5},
               {"target" : "publisher2", "cpc_modification" : 0.9}
    ]
  }
}
```

## 1.7 Pausing / Unpausing Campaigns

The Campaign's `status` field is *read-only*, and thus cannot be modified via an update request. Instead, one must change the Campaign's `is_active` field. The `is_active` field has two possible values: `true` and `false`. When an Item has both `is_active=true` and `status=RUNNING`, it is considered "active". If the Item has `is_active=false` its `status` will become `PAUSED`.



When a Campaign is `PAUSED`, it is not served in Taboola widgets.

### 1.7.1 Pause

In order to *pause* a Campaign, send an *update* request in the following form:

```
POST /backstage/api/1.0/[account-id]/campaigns/[campaign-id]/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```

```
{
  "is_active" : false
}
```

#### 1.7.1.1 Example Response

200 OK

```
{
  "id": "123",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "tracking_code": "",
  "cpc": 0.25,
  "daily_cap": 0.0,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": null,
  "sub_country_targeting": null,
  "postal_code_targeting": null,
  "platform_targeting": null,
  "publisher_targeting": null,
  "comments" : null,
  "start_date": null,
  "end_date": null,
  "approval_state" : "APPROVED",
  "is_active": false,
  "spent": 0.0,
  "status": "PAUSED",
  "daily_ad_delivery_model" : "ACCELERATED"
}
```

### 1.7.2 Unpause

In order to *unpause* a Campaign, send an HTTP *POST* request in the following form:

```
POST /backstage/api/1.0/[account-id]/campaigns/[campaign-id]/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```

```
{
  "is_active" : true
}
```

### 1.7.2.1 Example Response

200 OK

```
{
  "id": "123",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Pizza",
  "tracking_code": "",
  "cpc": 0.25,
  "daily_cap": 0.0,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": null,
  "sub_country_targeting": null,
  "postal_code_targeting": null,
  "platform_targeting": null,
  "publisher_targeting": null,
  "comments" : null,
  "start_date": null,
  "end_date": null,
  "approval_state" : "APPROVED",
  "is_active": true,
  "spent": 0.0,
  "status": "RUNNING",
  "daily_ad_delivery_model" : "ACCELERATED"
}
```

## 1.8 Country / Sub country Campaign targeting

The Campaign's geo targeting can be by country or by sub country. For country targeting possible values are two letter country code. For sub country targeting you may only target sub region / DMA of a single country that is being targeted.



Most updated possible country and sub country codes are accessible via dictionary.



Sub country targeting can only be used if only 1 country is used in country targeting in **INCLUDE** mode, and the region or DMA codes must be relevant to that country. In addition, this can be used for region targeting or DMA targeting (DMA only in US) but not both at the same time.

### 1.8.1 Example: Updating Campaign's Sub country targeting

POST /backstage/api/1.0/**taboola-demo-advertiser**/campaigns/**124**

Host: https://backstage.taboola.com

Authorization: Bearer [**access-token**]

```
{
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["US"]
  },
  "sub_country_targeting": {
    "type" : "INCLUDE",
    "value" : ["NY"]
  }
}
```

## Response:

200 OK

```
{
  "id": "124",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign",
  "branding_text": "Yummy Pizza",
  "tracking_code": "taboola-tracking-code",
  "cpc": 0.5,
  "daily_cap": 500,
  "spending_limit": 5000,
  "spending_limit_model": "ENTIRE",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["US"]
  },
  "sub_country_targeting": {
    "type" : "INCLUDE",
    "value" : ["NY"]
  },
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": null,
  "comments" : null,
  "start_date": "2015-04-24",
  "end_date": "2015-05-24",
  "approval_state" : "APPROVED",
  "is_active": true,
  "spent": 0.0,
  "status": "RUNNING",
  "daily_ad_delivery_model" : "STRICT",
  "traffic_allocation_mode" : "EVEN"
}
```

## 1.9 Using publisher targeting (blacklisting publishers)

It is possible to prevent your campaign from appearing in specific publishers (please note that this is limited up to 430 publishers per campaign). This is performed via the **publisher\_targeting** field of the Campaign. This field can contain an array of "Account IDs" representing publishers to block.

### 1.9.1 Updating the **publisher\_targeting** field

To set up a list of blocked publishers for a certain Campaign, you should send an update request like so:

```
POST /backstage/api/1.0/taboola-demo-advertiser/campaigns/124/
Host: https://backstage.taboola.com
Authorization: Bearer [access-token]
```

```
{
  "publisher_targeting": {
    "type": "EXCLUDE",
    "value": ["taboola-demo-publisher"]
  }
}
```

#### Response:

200 OK

```
{
  "id": "124",
  "advertiser_id" : "taboola-demo-advertiser",
  "name": "Demo Campaign - Edited",
  "branding_text": "Pizza",
  "tracking_code": "taboola-track",
  "cpc": 0.25,
  "daily_cap": 100,
  "spending_limit": 1000,
  "spending_limit_model": "MONTHLY",
  "country_targeting": {
    "type" : "INCLUDE",
    "value" : ["AU", "GB"]
  },
  "platform_targeting": {
    "type" : "INCLUDE",
    "value" : ["TBLT", "PHON"]
  },
  "publisher_targeting": {
    "type": "EXCLUDE",
    "value": ["taboola-demo-publisher"]
  },
  "comments" : null,
  "start_date": "2015-04-30",
  "end_date": "2015-05-30",
  "approval_state" : "APPROVED",
```

```
"is_active": true,  
"spent": 0.0,  
"status": "RUNNING"  
}
```

After performing such an update, this Campaign will stop appearing on this publisher's website.

### 1.9.2 Fetching a List of Available Publishers

In order to be able to block specific Publishers, one must know the Publisher's `account_id`. To *fetch* a list of all Publisher Accounts in the Taboola network, send an HTTP *GET* request in the following form:

```
GET /backstage/api/1.0/taboola-network/publishers/  
Host: https://backstage.taboola.com  
Authorization: Bearer [access-token]
```



To access this endpoint you must know the Admin Network to which your account belongs. This information is exposed via the Token Details resource.



This endpoint is used for getting a list of all Publishers under an Admin Network.

#### 1.9.2.1 Example Request

```
GET /backstage/api/1.0/taboola-network/publishers/  
Host: https://backstage.taboola.com  
Authorization: Bearer [access-token]
```

#### 1.9.2.2 Example Response

200 OK

```
{  
  "results": [  
    {  
      "id"       : "124",  
      "name"     : "Taboola Demo - Publisher",  
      "account_id" : "taboola-demo-publisher"  
    },  
    // ... more results  
  ]  
}
```