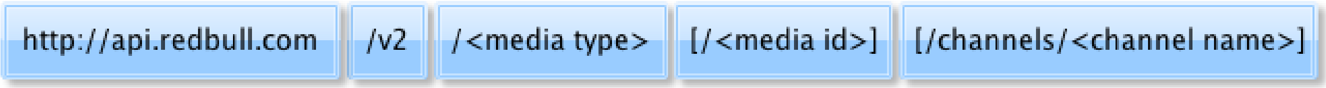
Basic Queries

Table of Contents

- Table of Contents
- Structure of Basic Queries
- Result
- Examples

Structure of Basic Queries



http://api.redbull.com /v2 /<media type> [/<media id>] [/channels/<channel name>]

A Basic Query may consist of the following five components:

- the hostname - usually api.redbull.com. replace it with stage-api.redbull.com for testing purposes
- the API version - the letter 'v' and a consecutive number (v1, v2, ...)
- the media type - a media type name in its plural form like stories, images, videos
- an optional media id - a unique id that represents exactly one media item in the Media API
- an optional channel name - a channel name to limit or filter the result. if no channel is passed to the Media API, it will return a list of all medias of a given media type, unless the caller passed a media id to the API

Result

Basic Queries always return a JSON Object that contain meta information and a JSON Array of media items.

Meta information attributes are:

- api_version - the version of the API
- api_url - the API query itself
- api_date_utc - the time, the API query's result was created in the UTC form
- type - the media type that was queried
- total_count - number of media items in the media items JSON Array
- profile - the name of the Profile that was used to transform the result

The media items JSON array contain one or more media items. The name of the JSON array's attribute is the same as the value of the meta information "type" attribute.

The API query

```
http://api.redbull.com/v2/stories/1331574771744
```

will return the following JSON object. The JSON object contain always a JSONArray of media items, even if the result is only one media item.

```
{
    "api_date_utc": "05.09.12 05:07",
    "api_url": "/stories/1331574771744",
    "stories": [
        {
            "location": "New York City, USA",
            "featuredimage":
"http://image.redbull.com/rbcom/0001/0/9999/640/001/15/843/Kevin Robinson and the Red
Bull Experiment 2008.jpg",
            "channels": "bike",
            "publisheddate": "2012-06-23 15:34:46",
            "urititle": "kevin-robinson-breaks-bmx-high-air-world-record",
            "type": "stories",
            "venue": "Central Park",
            "url_bike":
"http://www.redbull.com/en/bike/stories/1331574771744/kevin-robinson-breaks-bmx-high-a
ir-world-record",
            "url":
"http://www.redbull.com/en/bike/stories/1331574771744/kevin-robinson-breaks-bmx-high-a
ir-world-record",
            "apikey": "http://api.redbull.com/v2/stories/1331574771744",
            "id": "1331574771744",
            "author": "redbull.com team",
            "title": "Kevin Robinson breaks BMX high air world record",
            "startdate": "2008-06-13 00:00:00",
            "Publist": "Bike",
            "description": "",
            "name": "Kevin Robinson World Record 2008",
            "teaser": "Kevin Robinson sets a new BMX high air world record as part of
his Red Bull Experiment",
            "subtype": "Default",
            "longitude": "",
            "cs_type": "RedBullStory",
            "latitude": "",
            "featuredimage_blank":
"http://image.redbull.com/rbcom/0001/0/{image.width}/{image.height}/001/15/843/Kevin
Robinson and the Red Bull Experiment 2008.jpg"
        }
    ],
    "type": "stories",
    "api_version": "v2",
    "total_count": 1
}
```

Examples

API Query	Description
http://api.redbull.com/v2/stories	returns a list of all medias of type **stories**
http://api.redbull.com/v2/stories/1331574771744	returns exactly one media item of type **stories** with media id **1331574771744**
http://api.redbull.com/v2/stories/channels/bike	returns a list of all medias of type **stories** but only from the **bike** channel

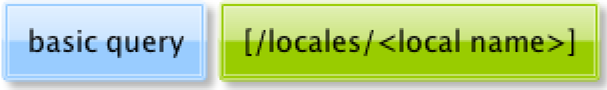
http://api.redbull.com/v2/stories/1331574771744/channels/bike	returns exactly one media item of type **stories** with media id **1331574771744**. *What means "/channels/bike"?* If a default transformation profile was created for the "channel/media type" combo, the result will be transformed with this profile. See Profiles for further information.

Filters for Basic Queries

Table of Contents

- Table of Contents
- Locales Filter
- Translation Filters
- Related Medias Filter
- Category Filter
- Discipline Filter
- Athlete Filter
- Contains Media Items Filter
- Referenced Medias Filters (parent relations)
- Published Date Filter
- EpisodeStartDate Filter for Episodes
- EventStartDate Filter for Events
- Current Date Filter for Events
- Sorting Filter
- Pagination Filter

Locales Filter



basic query [/locales/<local name>]

The Locales filters allows you to request media items in a specific language. The response will contain only media items of the given locale. To find out what locales are supported by the system, use the following query:

```
http://api.redbull.com/v2/misc/locales
```

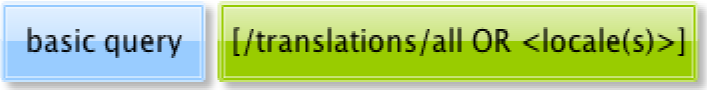
It will return a json feed, that contain all available locales. It doesn't include references to channels or media types.

Now you can add the Locales filter to your query:

```
http://api.redbull.com/v2/stories/locales/en_INT/sortby/publisheddate/sortorder/desc/p
agination/1-20
```

The response will contain all media items that match the passed locale. If no media item was found for the given locale, the response will contain zero items.

Translation Filters



basic query [/translations/all OR <locale(s)>]

Use this filter to get all translations of a Single Media Item, on condition that the Media API contains versions of the Media Item with different locales (@see: Media API Input Format Definition to see how to add translations to a Media Item).

The following query will return a Single Media Item (locale is **de_INT**)

```
http://api.redbull.com/v2/stories/1331579336326/channels/bike
```

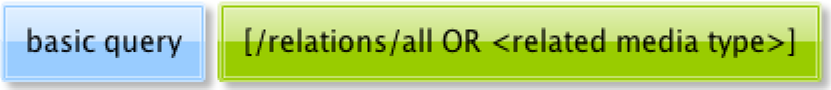
To get all translations of this Media Item, simply add **/translations/all** or **/translations/*** to the query

```
http://api.redbull.com//v2/stories/1331579336326/channels/bike/translations/all
http://api.redbull.com//v2/stories/1331579336326/channels/bike/translations/*
```

To get one or more specific translation of the Media Item add **/translations/[locale(s)]** to the query.

```
http://api.redbull.com/v2/stories/1331579336326/channels/bike/translations/en_INT
http://api.redbull.com/v2/stories/1331579336326/channels/bike/translations/en_INT/fr_I
NT/es_INT
```

Related Medias Filter



basic query [/relations/all OR <related media type>]

This filter adds related media items of a given **Media Type** or **all** related **Media Items** to a result (JSON feed) that was returned by the Basic Query. The related **Media Items** are shown as references and doesn't contain content. The filter adds a JSONArray "relations" to the JSON feed.

An API query like

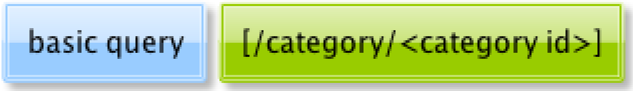
```
http://api.redbull.com/v2/stories/1331574771744/relations/images
```

will return the following relations JSONArray:

```
"relations": [
    {
        "relationname": "StoryContent",
        "childid": "1331574762671",
        "childtype": "images",
        "apikey": "http://api.redbull.com/v2/images/1331574762671"
    },{
        "relationname": "FeaturedImage",
        "childid": "1331574762671",
        "childtype": "images",
        "apikey": "http://api.redbull.com/v2/images/1331574762671"
    }
]
```

Use Transformation Profiles, if you need to return content in preference of references.

Category Filter


basic query [/category/<category id>]

This filter will return an JSON Array of Media Items that belong to a given category (category is a Redbull.com Media Type)

The query

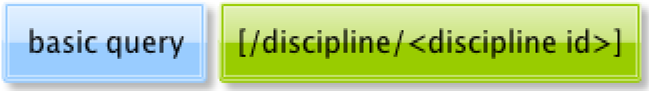
```
http://api.redbull.com/v2/stories/category/1331579450292/locales/en_INT
```

return all stories that are related to the category with id = 1331579450292 (Dance).

Use the following query to get a list of the available categories (change locales if necessary)

```
http://api.redbull.com/v2/categories/locales/en_INT
```

Discipline Filter


basic query [/discipline/<discipline id>]

This filter will return an JSON Array of Media Items that belong to a given discipline (discipline is a Redbull.com Media Type)

The query

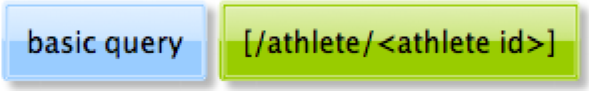
```
http://api.redbull.com/v2/stories/discipline/1326300799021/locales/en_INT
```

return all stories that are related to the discipline with id = 1326300799021 (MTB Downhill).

Use the following query to get a list of the available disciplines (change locales if necessary)

```
http://api.redbull.com/v2/disciplines/locales/en_INT
```

Athlete Filter


basic query [/athlete/<athlete id>]

The filter will return a JSON Array of Media Items (stories, events, images ...) that have references to a given Athlete

The query

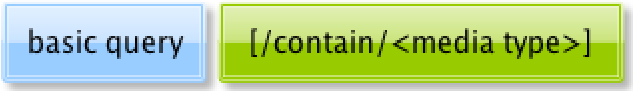
```
http://api.redbull.com/v2/stories/athlete/1331574617524/locales/en_INT
```

return all stories that have a reference to the athlete with id = 1331574617524 (Sebastian Vettel).

Use the following query to get a list of the available athletes (change locales if necessary)

```
http://api.redbull.com/v2/athletes/locales/en_INT
```

Contains Media Items Filter



basic query [/contain/<media type>]

This filter returns only those media items of a given **Media Type** that contain (has relations to) **Media Items** with the Media Type which was passed as value for the filter.

An API query like

```
http://api.redbull.com/v2/stories/contain/videos
```

will return a JSONArray of stories that contain videos (as relations). It will return an empty result if no stories were found that contain videos.

Use Transformation Profiles, if you need to return content in preference of references.

Examples

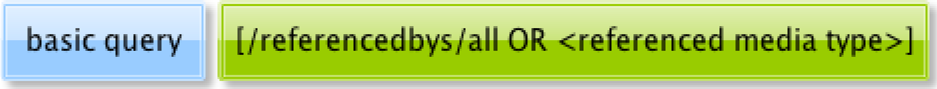
The first 5 **stories** that contain **videos**, of channel **bike**, with locale **en_INT**:

```
http://api.redbull.com/v2/stories/contain/videos/channels/bike/locales/en_INT/pagination/1-5
```

Alle **athletes** that contain **videos**, of channel **bike**, with locale **de_DE** (display relations in feed)

```
http://api.redbull.com/v2/athletes/contain/videos/channels/bike/locales/de_DE/relations/videos
```

Referenced Medias Filters (parent relations)



basic query [/referencedbys/all OR <referenced media type>]

The filter adds referenced media items of a given **Media Type** or all referenced **Media Items** to a feed. Referenced medias have a parent to child relation to the media item in the feed, where the media item is the child and the referenced **Media Items** are the parents.

An API query like:

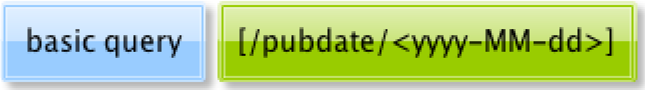
```
http://api.redbull.com/v2/stories/1331574771744/referencedbys/stories
```

returns the following feed.

```
"referencedbys": [
{
     "childid": "1331575162966",
     "childtype": "stories",
     "apikey": "http://api.redbull.com/v2/stories/1331575162966"
   }
]
```

Referencedbys can also be included in feeds that are transformed with a **Profile**.

Published Date Filter



The Publish Date attribute is a mandatory attribute for all **Media Items**. The value of this attribute is the time, the Media Item was published in the source system. Don't confuse it with the time, it was imported into the Media API.

The accept dates of the following formats:

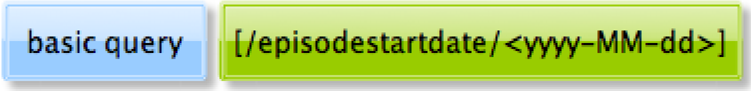
- yyyy-MM-dd
- yyyy-MM-ddTHH:mm
- yyyy-MM-ddTHH:mm:ss

The result of an API query with the Published Date filter contains all **Media Items** of a given **Media Type** that were published that time or later.

The following API query contain all "stories" of all "channels", that were published at October, 1st 2012 and later ...

```
http://api.redbull.com/v2/stories/pubdate/2012-10-01
```

EpisodeStartDate Filter for Episodes



The EpisodeStartdate filters applies only to the media type **episodes**. If one adds this filter to an api query, the Media API will return all episodes that will take place that date or later.

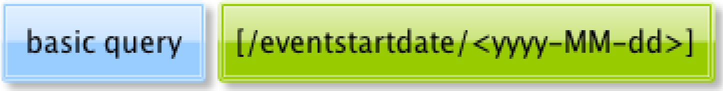
The accept dates of the following formats:

- yyyy-MM-dd
- yyyy-MM-ddTHH:mm
- yyyy-MM-ddTHH:mm:ss

The following API query will return all episodes of channel bike, that took place at October, 1st, 2012 or later.

```
http://api.redbull.com/v2/episodes/channels/bike/episodestartdate/2012-10-01/locales/e
n_INT
```

EventStartDate Filter for Events



basic query [/eventstartdate/<yyyy-MM-dd>]

The EventStartDate filters applies only to the media type **events**. If one adds this filter to an api query, the Media API will return all events that take place that date or later.

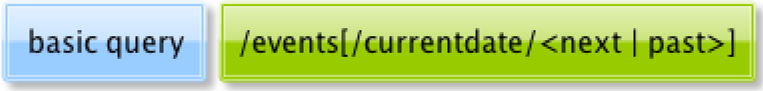
The accept dates of the following formats:

- yyyy-MM-dd
- yyyy-MM-ddTHH:mm
- yyyy-MM-ddTHH:mm:ss

The following API query will return all events of channel bike, that took place at October, 1st, 2012 or later.

```
http://api.redbull.com/v2/events/channels/bike/eventstartdate/2012-10-01
```

Current Date Filter for Events



basic query /events[/currentdate/<next | past>]

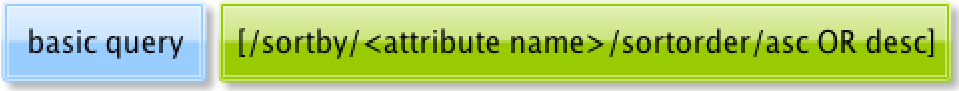
The CurrentDate Filter applies only to the media type **events**. The filter will be ignored for other media types. This filter knows two values: **past** and **next**.

- **past** - the result will contain only events that are already over. That means, the event's eventstartdate and the eventenddate are in the past (starting from now).
- **next** - the result will contain only events that will take place in the future. that means, the event's eventstartdate is in the future (starting from now).

```
http://api.redbull.com/v2/events/channels/bike/currentdate/past
```

The query above will return all past events of channel bike.

Sorting Filter



basic query [/sortby/<attribute name>/sortorder/asc OR desc]

The result of an Media API query is sortable by each attribute of the given **media type**.

- **sortby** - the name of the attribute you wish to sort by

- **sortorder** - controls the sort direction. asc = ascending, desc = descending (default asc)

To find out, what attributes a media type may contain, one can run the following Media API query:

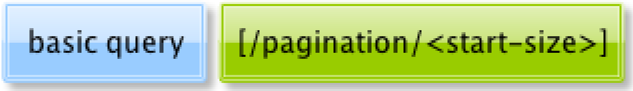
```
http://api.redbull.com/v2/misc/channels/bike/mediatypes
```

It will return a JSON feed, that contains a list of all media types of the channel bike. Each media type contains a "attributes" JSON array, which shows you the available attributes for the media type. Therefore some of the media types are shared between channels, the feed does contain only those attributes of the media type that are available within the channel!

Finally, the following Media API queries demonstrate how the filter works:

	http://api.redbull.com/v2/athletes/channels/bike/sortby/surname/sortorder/desc	returns a feed of all athletes of channel bike, which is sorted by the surname of the athletes.
	http://api.redbull.com/v2/athletes/channels/bike/sortby/surname/sortorder/desc	returns a feed of all events sorted by their published date in descending order (newest first

Pagination Filter



One can use this filter to decrease the number of media items in the result, that was returned by a Media API query.

The value of the pagination filter consists of two numbers. The first one is the start point (list index, where the index starts with 1!) of the list and the second one is the size of the result.

The query below will return the 5 latest events of channel bike. As one can see, the filters can be combined with each other. The first filter sorts the result by published date in descending order (the newest first) and the second filter decreases the number of the result to 5 items.

```
http://api.redbull.com/v2/events/channels/bike/sortby/publisheddates/sortorder/desc/pagination/1-5
```

Media API Search

The Media API provides a fulltext search. The search supports the following commands:

Search API

Search for media items of a given Channel „channel“ and a given Media Type „type“, where the value of Attribute „fields“ like „q“. The search is case-insensitive.

```
http://api.redbull.com/v2/query?q=bike&type=stories&fields=name,channels
```

Search for media items of a given Channel channel and a given Media Type and sort the result by publisheddate descending.

```
https://api.redbull.com/v2/query?q=skate&fields=channels,author&types=stories,events,a
thletes&sortby=publisheddatesortorder=desc
```

Additonal Parameters

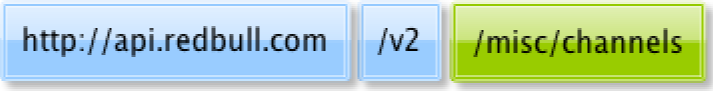
Parameter	default value	description
operator	and	matched documents where both terms exist anywhere in the text. use "or" to match documents with one term exists in the text
profile	mobileapp-stories	profile which is applied of the result
sortby	publisheddate	sort the result by a given attributename
sortorder	desc	sort direction abscending or descending
pagination	1-20	pagination the default is 1-20
locale	en_int	filter content by a locale

Meta Information Queries

Table of Contents

- Table of Contents
- Available Channels
    - Result
- Locales
- Media Types and their Attributes
    - Result
- Profiles
    - Result

Available Channels



http://api.redbull.com /v2 /misc/channels

Returns a list of available channels.

Result

A JSON object "channel" that contain an JSON array of channels:

```
{
   "channels":
    [
      {
        "description":"RBWebTV",
        "name":"rbwebtv"
      },{
        "description":"RBBCOne",
        "name":"rbbcone"
      },{
        "description":"Bike",
        "name":"bike"
      },{
        "description":"Formula1",
        "name":"formula1"
      },{
        "description":"OffRoad",
        "name":"offroad"
      },{
        "description": "MotoGP",
        "name":"motogp"
      }
    ]
}
```

Locales

http://api.redbull.com /v2 /misc/locales

The response will contain a list of all available locales. It does not include references to media types or channels.

```
{
   locales: [
   {
      name: "en_INT"
   },

   {
      name: "de_INT"
   },
   {
      name: "fr_INT"
   },
   {
      name: "es_INT"
   }
  ]
}
```

Media Types and their Attributes

http://api.redbull.com /v2 /misc/mediatypes

http://api.redbull.com /v2 /misc/channels/<channel name>/mediatypes

Returns a list of all present media types, or if a channel was passed, it returns a list of all media types of that channel. The result includes a list of attributes for each media type.

Result

The result is a JSON object, that includes a JSON array attributes.

- channel: name of the channel
- mediatypes: JSON array of media types
  - attributes: JSON array of attributes
  - name: name of the the media type
  - description: short description of the media type

```
{
    "channel": "bike",
    "mediatypes": [
        {
            "attributes": [
                {
                    "name": "Publist"
                },
                {
                    "name": "apikey"
                },
                {
                    "name": "author"
                },
                {
                    "name": "channels"
                },
                {
                    "name": "cs_type"
                },
                {
                    "name": "description"
                },
                {
                    "name": "enddate"
                },
                {
                    "name": "featuredimage"
                },
                {
                    "name": "featuredimage_blank"
                },
                {
                    "name": "id"
                },
                {
                    "name": "latitude"
                },
                {
                    "name": "location"
                },
                {
                    "name": "longitude"
                },
                {
                    "name": "name"
                },
                {
                    "name": "publisheddate"
                },
                {
                    "name": "startdate"
                },
                {
                    "name": "subtype"
                },
                {
                    "name": "teaser"
                },
                {
                    "name": "title"
                },
                {
                    "name": "type"
                },
                {
                    "name": "urititle"
                },
                {
                    "name": "url"
                },
                {
                    "name": "url_bike"
                },
                {
                    "name": "venue"
                }
            ],
            "description": "RedBullStory",
            "name": "stories"
        }
    ],
    "total_count": 13
}
```

Profiles

http://api.redbull.com /v2 /misc/profiles

Returns a list of profiles. Profiles are used to transform the result of an API request.

Result

The result contains a JSON array "profiles". Each profile contains the channel and the media type it belongs to. It also includes the attributes that are configured for the media type, the optional relations and the optional referencedbys (inkl. their attributes).

- profiles
  - attributes: JSON array of attributes
  - channel: name of the channel
  - defaultprofile: true | false
  - type: name of the media type
  - relations: JSON array of relations
    - name: name of the relation
    - childtypes: JSON array of child media types
      - attributes: attributes for the child media type
      - childtype: name of the child type

```
{
    "profiles": [
        {
            "attributes": [
                {
                    "name": "apikey"
                },
                {
                    "name": "author"
                },
                {
                    "name": "channel"
                },
                {
                    "name": "channels"
                },
                {
                    "name": "cs_type"
                },
                {
                    "name": "description"
                },
                {
                    "name": "enddate"
                },
                {
                    "name": "featuredimage"
                },
                {
                    "name": "id"
                },
                {
                    "name": "latitude"
                },
                {
                    "name": "location"
                },
                {
                    "name": "longitude"
                },
                {
                    "name": "name"
                },
                {
                    "name": "publisheddate"
                },
                {
                    "name": "publist"
                },
                {
                    "name": "startdate"
                },
                {
                    "name": "subtype"
                },
                {
                    "name": "teaser"
                },
                {
                    "name": "title"
                },
                {
                    "name": "type"
                },
                {
                    "name": "urititle"
                },
                {
                    "name": "url"
                },
                {
                    "name": "venue"
                }
            ],
            "channel": "bike",
            "defaultprofile": true,
            "name": "bike-stories-default",
            "relations": [
                {
                    "childtypes": [
                        {
                            "attributes": [
                                {
                                    "name": "apikey"
                                },
                                {
                                    "name": "fulltext"
                                    },
                                    {
                                        "name": "id"
                                    }
                                ],
                                "childtype": "texts"
                            },
                            {
                                "attributes": [
                                    {
                                        "name": "apikey"
                                    },
                                    {
                                        "name": "channels"
                                    },
                                    {
                                        "name": "referenceid"
                                    },
                                    {
                                        "name": "stillimage"
                                    },
                                    {
                                        "name": "subtype"
                                    },
                                    {
                                        "name": "teaser"
                                    },
                                    {
                                        "name": "title"
                                    },
                                    {
                                        "name": "url"
                                    },
                                    {
                                        "name": "videourl"
                                    },
                                    {
                                        "name": "videourl_1248_700"
                                    },
                                    {
                                        "name": "videourl_1280_544"
                                    },
                                    {
                                        "name": "videourl_1280_720"
                                    },
                                    {
                                        "name": "videourl_1920_1080"
                                    },
                                    {
                                        "name": "videourl_396_224"
                                    },
                                    {
                                        "name": "videourl_480_268"
                                    },
                                    {
                                        "name": "videourl_512_216"
                                    },
                                    {
                                        "name": "videourl_512_288"
```

```
                        },
                        {
                            "name": "videourl_640_360"
                        },
                        {
                            "name": "videourl_720_404"
                        },
                        {
                            "name": "videourl_768_324"
                        },
                        {
                            "name": "videourl_768_432"
                        },
                        {
                            "name": "videourl_960_540"
                        }
                    ],
                    "childtype": "videos"
                },
                {
                    "attributes": [
                        {
                            "name": "apikey"
                        },
                        {
                            "name": "caption"
                        },
                        {
                            "name": "description"
                        },
                        {
                            "name": "imageurl"
                        },
                        {
                            "name": "title"
                        }
                    ],
                    "childtype": "images"
                }
            ],
            "name": "StoryContent"
        }
    ],
    "type": "stories"
}
```