AudioVisual API

Introduction

The **Sportradar AudioVisual API** is an API used to integrate video streams from events covered by the AudioVisual unit of Sportradar into a web application or a mobile app.
This document describes how to integrate the API and is intended for developers.

Table of contents

New concepts compared to the Live Channel Online API

(only relevant for clients which integrated the Live Channel Online API)

The AudioVisual API is a completely new API. If you have already integrated the Live Channel Online API, we strongly recommend you to move to the new AudioVisual API in order to benefit from all of its new features and improvements. The AudioVisual API introduces new concepts in comparison to the Live Channel Online API, and it therefore requires a new integration. Below is a list of the main new concepts:

- New and simpler authentication mechanism: Bearer token instead of JWT token.
- No need to pass the “client ID” as “cid” parameter in the API URL: the Bearer token is now used to identify the client.
- Support for all AudioVisual products: Live Channel Online, Live Channel Retail Flex, Live Channel Trading, etc.
- AudioVisual events instead of “matches”: the Live Channel Online API was centered around the concept of Sportradar matches (with “match IDs”, also known in the Sportradar terminology as “sport events”). The new AudioVisual API is centered around the concept of AudioVisual events which are not necessarily associated to a Sportradar sport event. It provides more flexibility and clarity, access to more content, as well as support for a better data structure. Please note, however, that AudioVisual events for betting products such as Live Channel Online, Live Channel Retail Flex and Live Channel Trading will almost constantly be associated to a Sportradar sport event.
- A same event contains one or multiple contents (videos). There is always one content marked as “main” content for a given event. The main content is generally the stream of the full duration of the event.
- Support for Live and Video on demand (VOD).
- The properties “isActive” and “isLive” from the /streamed-matches endpoint of the Live Channel Online API have been replaced respectively by “event_status” (on event level) and “stream_status” (on stream level). See below in this documentation the section related to the GET /events endpoint for more details.
- No more support for SMIL files: SMIL format was used as a manifest for multi-bitrate RTMP streaming in the Live Channel Online API. The AudioVisual API does not support this format any more.
- Additional filter parameters. See below in this documentation the sections related to GET /events endpoint for more details.
- New URN-style IDs as ID scheme (see dedicated section below).
- Support for geo-restrictions on country subdivisions such as US states (support to be activated soon).

Please note that support for new features and technologies will be added only to the AudioVisual API. If you are using the Live Channel Online API, we urge you to integrate the new AudioVisual API instead.

Access

The API is available via the HTTP protocol and must be called from the server(s) of the client and not from a frontend application.
Access to the API is restricted by IP. In order to access the API, Sportradar needs to whitelist the IP (s) of the client server(s) from which the API will be called.

Please note: In case you need to whitelist Sportradar's servers on your firewall, the following hostname must be used in the firewall rules: av-api.betradar.com
If your firewall does not support whitelisting of hostnames or if there are any further questions please contact the integration team of Sportradar's Audiovisual unit.

URL

The API is hosted on **https://av-api.betradar.com/**.
The URLs of the endpoints follow the following pattern: **https://av-api.betradar.com/<version>/<endpoint_path>?<query_parameters>**
Example of URL: **https://av-api.betradar.com/v1/events**
Setting the version in the URL is optional. If it is omitted, the latest version of the API will be used. The latest version of the API is “v1”.

Response formats

The API supports at the moment only **JSON** as response format. The support for **XML** will come soon. The response format is defined via the “Accept” HTTP header. For JSON, use “application/json”. For XML, it will be “application/xml”. If no “Accept” HTTP header is sent, the API will use JSON by default.

Authentication

Authentication is required for all of the endpoints of the API. The authentication scheme used by the API is **Bearer authentication** (also called token authentication). To authenticate, an API key (bearer token) must be sent in the “Authorization” HTTP header when making requests to the API. The string “Bearer ” must be included before the API key, as follows:
Authorization: Bearer <api-key>

The API key is provided by Sportradar as part of an AudioVisual product. If you already have an API key with the Live Channel Online API, it is not possible to re-use it with the AudioVisual API. Please contact us if you need a new API key.

The API key should never be shared with a third-party and should never be exposed publicly. It is the responsibility of the client to keep it secret.

ID scheme

Like most of other APIs provided by Sportradar, the values for representing IDs are using URN-style IDs with the following format: “XXX:YYY:###”.

- “XXX” is some kind of context and can be “sr” or “av”. “sr” refers to an entity which is part of the Sportradar world and which is globally used for other products in the Sportradar catalogue. “av” refers to an entity which is specific to the Sportradar AudioVisual products.
- “YYY” is the name of the entity within the context. It is generally a string.
- “###” is typically a raw identifier for that entity. It is generally an integer.

Expected workflow

The following workflow is recommended for integrating the AudioVisual Client API:

1. The client server requests the GET /events endpoint to get the list of events and to have an overview of the different streams available for them.
2. When an end user wants to play the video stream of a content (e.g. via the client’s web site), a request is made from the client server to the GET /streams endpoint to get the stream URL. The IP of the end user must be passed to the API via the “X-Real-IP” HTTP header.
3. The stream URL returned in the response of the API should be passed on to the end user’s video player, which will then connect to a streaming server.
4. The streaming server will validate the stream URL (by doing an IP validation among others) and play the stream back to the end user.

Special requirements for streaming in the USA, Canada and Mexico

When the end-user is located in one of these countries:

1. Client is only permitted to send requests to the API endpoints which return stream URLs if the end-user is located in a territory which is defined as a whitelisted area according to the agreement with Sportradar.
2. Additionally, two more HTTP headers must be sent in the request:
    - “Geo-Position”: comma-separated string containing the latitude and longitude values specifying the location of the end-user, in WGS84 format. For example: 40.714224,-73.961452.
    - “Geo-Position-Timestamp”: Unix timestamp of the latitude/longitude coordinates retrieval.

Based on this information, the API will perform a check to verify that the end-user is located in a territory which is allowed, and will make sure that the timestamp is not older than 1 minute ago. If this check fails, the stream URL will not be returned. Instead, an error will be returned in the response.

Please note that this check is not yet active in our API but clients which are streaming or planning to stream in these countries need to implement these two new HTTP headers in order to be able to use the API endpoints which return stream URLs properly once we activate the check on our side.

API endpoints

GET /events

HTTP method: GET
Path: /events

This endpoint returns the lists of booked events (booked manually via the Live Booking Calendar in betradar.com or booked automatically via a client package).

An event is defined by various properties described below and contains one or multiple **“contents”**. A content is a video in relation to the event. For instance, an event such as a football match will contain a video of the full duration of the match but can also contain videos for goal clips, highlights, interviews, etc.
A content itself contains one or multiple **“streams”**. Streams are distincted by AudioVisual product (Live Channel Online, Live Channel Retail Flex, Live Channel Trading, etc) and distribution (Live or VOD).

By using this endpoint, you will be able to retrieve only events in the time frame between now – 1 day and now + 14 days. Events out of this time frame are not available, whatever are the filtering parameters used.

Events are sorted by event start time.

Parameters as query parameters in the URL
The following parameters can be passed as query parameters in the URL. They are all optional. If none of these parameters are passed, the API will return by default the events which have a start time between now – 6 hours and now + 3 days or which have at least one stream with distribution type **“Live”** and broadcast status **“On Air Broadcast”**.

Name	Type	Optional?	Description
“from”	String	Yes	Lower bound of the event time frame in ISO-8601 format. Has to be used in combination with “to” parameter.
“to”	String	Yes	Upper bound of the event time frame in ISO-8601 format. Has to be used in combination with “from” parameter.
“event”	String	Yes	Event ID. Support both single value (event=) and multi-values (event[]=). The ID scheme “av:event:<integer>” must be used. Filter the list of events for the given event(s).
“client_event”	String	Yes	Client event ID. This parameter works only if a mapping exists between the Sportradar sport event ID and the client event ID. Support both single value (client_event=) and multi-values (client_event[]=). Does not use any ID scheme. Filter the list of events for the given client event(s).
“sport_event”	String	Yes	Sportradar sport event ID. Support both single value (sport_event=) and multi-values (sport_event[]=). For now, the ID scheme “sr:match:<integer>” must be used. Some other ID schemes may be available in the future. Filter the list of events for the given Sportradar sport event(s).
“event_status”	String	Yes	Event status ID. Support both single value (event_status=) and multi-values (event_status[]=). The ID scheme “av:event_status:<integer>” must be used. Filter the events for the given event status(es). Use GET /event-statuses endpoint to get the list of possible options. See the section "Description of statuses" at the end of this document for more information.
“content”	String	Yes	Content ID. Support both single value (content=) and multi-values (content[]=). The ID scheme “av:content:<integer>” must be used. Filter the list of events/contents for the given content(s).
“content_type”	String	Yes	Content type ID. Support both single value (content_type=) and multi-values (content_type[]=). The ID scheme “av:content_type:<integer>” must be used. Filter the list of events/contents for the given content type(s). Use GET /content-types endpoint to get the list of possible options.
“is_main_content”	Boolean	Yes	Is the content the main content of the event? Possible values: 0, 1. Filter the list of events/contents for the given value.
“stream”	String	Yes	Stream ID. Support both single value (stream=) and multi-values (stream[]=). The ID scheme “av:stream:<integer>” must be used. Filter the list of events/contents/streams for the given stream(s).
“product”	String	Yes	AudioVisual product ID. Support both single value (product=) and multi-values (product[]=). The ID scheme “av:product:<integer>” must be used. Filter the list of events/contents/streams for the given product(s). Use GET /products endpoint to get the list of possible options.
“distribution”	String	Yes	Distribution ID. Support both single value (distribution=) and multi-values (distribution[]=). The ID scheme “av:distribution:<integer>” must be used. Filter the list of events/contents/streams for the given distribution(s). Use GET /distributions endpoint to get the list of possible options.
“stream_status”	String	Yes	Stream status ID. Support both single value (stream_status=) and multi-values (stream_status[]=). The ID scheme “av:stream_status:<integer>” must be used. Filter the list of events/contents/streams for the given stream status(es). Use GET /stream-statuses endpoint to get the list of possible options. See the section "Description of statuses" at the end of this document for more information.

Example of response in JSON:

```
[
   {
      "id":"av:event:36852",
      "client_event_id": "ABCD-1234",
      "sport_event_id":"sr:match:17517635",
      "start_time":"2019-02-28T11:40:00+00:00",
      "end_time":null,
      "event_status":{
         "id":"av:event_status:2",
         "name":"Confirmed"
      },
      "first_level_category":{
         "id":"sr:sport:5",
         "name":"Tennis"
      },
      "second_level_category":{
         "id":"sr:category:785",
         "name":"ITF Men"
      },
      "third_level_category":{
         "id":"sr:competition:17630",
         "name":"ITF Portugal F2, Men Singles"
      },
      "competitors":[
         {
            "id":"sr:competitor:103625",
            "name":"GEERTS, MICHAEL",
            "qualifier":"home"
         },
         {
            "id":"sr:competitor:153644",
            "name":"MONTEIRO, JOAO",
            "qualifier":"away"
         }
      ],
      "venue":{
         "id":"sr:venue:27383",
         "name":"Court 1"
      },
      "contents":[
         {
            "id":"av:content:121364",
            "content_type":{
               "id":"av:content_type:6",
               "name":"Full match"
            },
            "name":null,
            "is_main":true,
            "streams":[
               {
                  "id":"av:stream:107200",
                  "product":{
                     "id":"av:product:1",
                     "name":"Live Channel Retail"
                  },
                  "distribution":{
                     "id":"av:distribution:1",
                     "name":"Live"
                  },
                  "start_time":"2019-02-28T11:40:00+00:00",
                  "end_time":null,
                  "stream_status":{
                     "id":"av:stream_status:2",
                     "name":"On Air Broadcast"
                  },
                  "languages":[],
                  "geo_restrictions":[
                     {
                        "device_category":{
                           "id":"av:device_category:1",
                           "name":"Desktop"
                        },
                        "country_iso_alpha2_codes":"AT,NO"
                     }
                  ]
               },
               {
                  "id":"av:stream:107202",
                  "product":{
                     "id":"av:product:2",
                     "name":"Live Channel Online"
                  },
                  "distribution":{
                     "id":"av:distribution:1",
                     "name":"Live"
                  },
                  "start_time":"2019-02-28T11:40:00+00:00",
                  "end_time":null,
                  "stream_status":{
                     "id":"av:stream_status:2",
                     "name":"On Air Broadcast"
                  },
                  "languages":[],
                  "geo_restrictions":[
                     {
                        "device_category":{
                           "id":"av:device_category:1",
                           "name":"Desktop"
                        },
                        "country_iso_alpha2_codes":"AT,NO,US",
                        "country_subdivision_iso_codes": {
                           "US": "US-NJ,US-NV"
                        }
                     },
                     {
                        "device_category":{
                           "id":"av:device_category:2",
                           "name":"Mobile"
                        },
                        "country_iso_alpha2_codes":"AT,NO,US",
                        "country_subdivision_iso_codes": {
                           "US": "US-NJ,US-NV"
                        }
                     }
                  ]
               }
            ]
         }
      ]
   }
]
```

Description of the properties

Property	**Type**	**Can be null/empty?**	**Description**
“id”	String	No	Event ID. ID scheme is always “av:event:<integer>”.
“client_event_id”	String	Yes	Client event ID. This property is present only if a mapping exists between the Sportradar match ID and the client event ID. Does not use any ID scheme. The client event ID is returned as it is.
“sport_event_id”	String	Yes	Sportradar sport event ID. For now, ID scheme is always “sr:match:<integer>”. Some other ID schemes may be available in the future.
“start_time”	String	No	Start time of the event in ISO-8601 format. Uses the UTC offset by default.
“end_time”	String	Yes	End time of the event in ISO-8601 format. Uses the UTC offset by default.

“event_status”		Object	No	Status of the event. See the section "Description of statuses" at the end of this document for more information.
	“id”	String	No	Event status ID. ID scheme is always “av:event_status:<integer>”. Use GET /event-statuses endpoint to get the list of possible values.
	“name”	String	No	Name of the event status.
“first_level_category”		Object	No	First level category of the event.
	“id”	String	No	First level category ID. Two ID schemes are possible: - “sr:sport:<integer>”: the property will contain the Sportradar sport ID. - “av:category:<integer>”: the property will contain the ID of the first level category (for Sportradar AudioVisual-only events).
	“name”	String	No	Name of the first level category.
“second_level_category”		Object	No	Second level category of the event.
	“id”	String	No	Second level category ID. Two ID schemes are possible: - “sr:category:<integer>”: the property will contain the Sportradar category ID. - “av:category:<integer>”: the property will contain the ID of the second level category (for Sportradar AudioVisual-only events).
	“name”	String	No	Name of the second level category.
“third_level_category”		Object	No	Third level category of the event.
	“id”	String	No	Third level category ID. Two ID schemes are possible: - “sr:tournament:<integer>”: the property will contain the Sportradar tournament ID. - “av:category:<integer>”: the property will contain the ID of the third level category (for Sportradar AudioVisual-only events).
	“name”	String	No	Name of the third level category.
“competitors”		Array	Yes	Array of “competitor” objects.
	“id”	String	No	Competitor ID. Three ID schemes are possible: - “sr:competitor:<integer>”: the property will contain the Sportradar competitor ID. - “sr:simpleteam:<integer>”: the property will contain the Sportradar simple team ID (only if the Sportradar competitor ID is not available). - “av:competitor:<integer>”: the property will contain the ID of the competitor (for Sportradar AudioVisual-only events).
	“name”	String	No	Name of the competitor.
	“qualifier”	String	No	Defines is the competitor is “home” or “away”.

“venue”				Object	Yes	Venue of the event.
	“id”			String	No	Venue ID. Two ID schemes are possible: • “sr:venue:<integer>”: the property will contain the Sportradar venue ID. • “av:venue:<integer>”: the property will contain the ID of the venue (for Sportradar AudioVisual-only events).
	“name”			String	No	Name of the venue.
“contents”				Array	No	Array of “content” objects.
	“id”			String	No	Content ID. ID scheme is always “av:content:<integer>”.
	“content_type”			Object	No	Type of the content.
		“id”		String	No	Content type ID. ID scheme is always “av:content_type:<integer>”. Use GET /content-types endpoint to get the list of possible values.
		“name”		String	No	Name of the content type.
	“name”			String	Yes	Name of the content.
	“is_main”			Boolean	No	True if the content is the main content of the event. For example, for a football match, the main content would be the video of the full duration of the match.
	“streams”			Array	No	Array of “stream” objects.
		“id”		String	No	Stream ID. ID scheme is always “av:stream:<integer>”. This is the ID to be sent when using the GET /streams/<stream_id>/<stream_type> endpoint.
		“product”		Object	No	AudioVisual product of the stream.
			“id”	String	No	Product ID. ID scheme is always “av:product:<integer>”. Use GET /products endpoint to get the list of possible values.
			“name”	String	No	Name of the product.
		“distribution”		Object	No	Distribution of the stream.
			“id”	String	No	Distribution ID. ID scheme is always “av:distribution:<integer>”. Use GET /distributions endpoint to get the list of possible values.
			“name”	String	No	Name of the distribution.
		“start_time”		String	No	Start time of the stream in ISO-8601 format. Uses the UTC offset by default.
		“end_time”		String	Yes	End time of the stream in ISO-8601 format. Uses the UTC offset by default.

		“stream_status”			Object	No	Status of the stream. See the section "Description of statuses" at the end of this document for more information.
			“id”		String	No	Stream status ID. ID scheme is always “av:stream_status:<integer>”. Use GET /stream-statuses endpoint to get the list of possible values.
			“name”		String	No	Name of the stream status.
		“languages”			Array	Yes	Array of “language” objects. Languages available as audio for the stream. If this array is empty, it means that only one default language will be available as audio for the stream.
			“id”		String	No	Sportradar language ID. ID scheme is always “sr:language:<integer>”.
			“name”		String	No	Name of the language.
		“geo_restrictions”			Array	No	Array of “geo restriction” objects.
			“device_categories”		Object	No	Device category for which the geo restriction applies.
				“id”	String	No	Device category ID. ID scheme is always “av:device_category:<integer>”. Use GET /device-categories endpoint to get the list of possible values.
				“name”	String	No	Name of the device category.
			“country_iso_alpha2_codes”		String	No	String of coma-separated country ISO alpha-2 codes. Countries allowed for the given stream and device category.
			“country_subdivision_iso_codes”		Object	Yes	Lists of coma-separated country subdivision ISO codes, grouped by country (where the key is the country ISO alpha-2 code). Country subdivisions allowed for the given stream and device category. If an allowed country has no subdivisions listed, it means that geo-restrictions are not handled on subdivision level for that country. Therefore all the subdivisions of that country are allowed. Please contact us to know which countries have their geo-restrictions handled on subdivision level. If none of the allowed countries have their geo-restrictions handled on subdivision level, this property will be omitted. This property is not supported in the current version of the API but support will come soon.

GET /events/<event_id>

HTTP method: GET
Path: /events/<event_id>

This endpoint returns the same response as the GET /events endpoint, but just for one event. The structure is the same except that the root element is an object instead of an array. Please see the documentation of GET /events endpoint for more information (note that the GET parameters from the GET /events endpoint do not apply to this endpoint).

Parameters in the path of the URL

Name	**Type**	**Optional?**	**Description**
“event_id”	String	No	Event ID. The ID scheme “av:event:<integer>” must be used.

GET /streams/<stream_id>/<stream_type>

HTTP method: GET
Path: /streams/<stream_id>/<stream_type>

This endpoint returns the URL of a stream by stream ID for a given stream type.

Parameters in the path of the URL

Name	Type	Optional?	Description
“stream_id”	String	No	Stream ID as provided in the GET /events endpoint. The ID scheme “av:stream:<integer>” must be used.
“stream_type”	String	No	Type of stream. Possible values: • “dash-manifest”: to get the DASH manifest URL. Available only for the products Live Channel Retail Flex, Live Channel Online and Third Party and only for a booking on the device category “Desktop” or “Mobile”. • “cmaf-manifest”: to get the CMAF manifest URL. Available only for the products Live Channel Retail Flex, Live Channel Online and Third Party and only for a booking on the device category “Desktop” or “Mobile”. • “hls”: to get the HLS stream URL. Available only for the products Live Channel Online and Third Party and only for a booking on the device category “Desktop” or “Mobile”. • “hls-manifest”: to get the HLS manifest URL. Available only for the products Live Channel Online and Third Party and only for a booking on the device category “Desktop” or “Mobile”. • “rtmp”: to get the RTMP stream URL. Available only for the products Live Channel Retail Flex, Live Channel Online and Third Party and only for a booking on the device category “Desktop”.

Parameters as query parameters in the URL

Name	Type	Optional?	Description
“device_category”	String	Yes	ID of the device category of the end-user who is going to play the stream. The ID scheme “av:device_category:<integer>” must be used. Use GET /device-categories endpoint to get the list of possible options. Since the geo restrictions on a stream can be different by device category, it is important to pass this parameter so that the API can perform a correct geo location check based on the end-user IP passed in the “X-Real-IP” HTTP header. If this parameter is not passed, a default value will be used, depending on the requested stream type: • “dash-manifest”: “Desktop”. • “cmaf-manifest”: “Desktop”. • “hls-manifest” / “hls”: “Mobile”. • “rtmp”: “Desktop”.
“language”	String	Yes	Sportradar language ID as provided in the GET /events endpoint. The ID scheme “sr:language:<integer>” must be used. Will return the stream URL, if there is one, for the given language.
“max_bitrate”	Integer	Yes	Maximum bitrate of the stream in bps. If the given value is lower than the lowest available bitrate, the URL of the stream with the lowest bitrate will be returned.
“label”	String	Yes	Unique label identifier used to divide traffic from the main account. This parameter will be forwarded and included (as a “label” query parameter) in the returned stream URL. Allowed values are restricted. Please contact us if you want to add support for a new value for your account.
"csid"	String	Yes	Client side ID. Unique and consistent ID for the end user. For example a hash of a user name. The string passed should be anonymized, and should be consistent with end user IDs provided to other Sportradar systems. This parameter will be forwarded and included (as a “csid” query parameter) in the returned stream URL.

Parameters as HTTP headers

Name	Type	Optional?	Description
“X-Real-IP”	String	No	IP of the end user which is intended to play the stream.
“Geo-Position”	String	Yes	Comma-separated string containing the latitude and longitude values specifying the location of the end-user, in WGS84 format. For example: 40.714224,-73.961452. Required only for streaming in the USA, Canada and Mexico, see section “Special requirements for streaming in the USA, Canada and Mexico” above.
“Geo-Position-Timestamp”	Integer	Yes	Unix timestamp of the latitude/longitude coordinates retrieval. Required only for streaming in the USA, Canada and Mexico, see section “Special requirements for streaming in the USA, Canada and Mexico” above.

Example of response in JSON:

```
{
   "url": "https://sportradar-lco.akamaized.net/24770/121364/dash/desktop
/tennis_itf_men_itf_portugal_f2_men_singles_17517635.mpd?
cid=24770&mid=17517635&ecid=121364&pid=2&dtid=1&sid=538435715117&hdnts=ip
=92.62.38.2~exp=1551450105~acl=%2F24770%2F121364%2Fdash%2Fdesktop%2F%
2A~hmac=03185324934d24d5c581af8f6ed1d7775eb92ac7467d2563915c708283f99ae6"
}
```

For “RTMP” streams, the response contains an additional property for the stream name:

```
{
   "url": "rtmp://cp508682.live.edgefcs.net/live/",
   "stream_name": "avup_single_81@s752670?
auth=da_cbdZdob6bqbbaeaYdzbXbFb.adaFdwb1-bCD.60-E-xAwAEoJorBoCFEa-
o8sakdlcm5&aifp=v1&cid=24770&mid=17517635&ecid=121364&pid=2&dtid=1&sid=28
7080384538"
}
```

Important note regarding RTMP stream URLs

Responses for RTMP stream URLs are special because the stream URL is divided into two different properties: **“url”** and **“stream_name”**. Depending on the video player you use, you will have to keep these two properties separated or you will have to join them into one single “full” URL, the latter being the most common case. If you need to join the **“url”** and **“stream_name”** properties into one single “full” URL, you have to make sure to always include the actual stream name before the query parameters because, depending on the streaming server used to deliver the stream, the query parameters can be either in the **“url”** property or in the **“stream_name”** property.

Example with the query parameters in the **“stream_name”** property: let’s **“url”** be **“rtmp://my.domain.com/path/”** and **“stream_name”** be **“my_stream_name?param1=value1”**. The full URL to put in the player should then be **“rtmp://my.domain.com/path/my_stream_name?param1=value1”**.

Example with the query parameters in the **“url”** property: let’s **“url”** be **“rtmp://my.otherdomain.com/otherpath/?otherparam1=othervalue1”** and **“stream_name”** be **“my_other_stream_name”**. The full URL to put in the player should then be **“rtmp://my.otherdomain.com/otherpath/my_other_stream_name?otherparam1=othervalue1”**.

GET /contents/<content_id>/product/<product_id>/distribution/<distribution_id>/stream/<stream_type>

HTTP method: GET
Path: /contents/<content_id>/product/<product_id>/distribution/<distribution_id>/stream/<stream_type>

This endpoint returns the URL of a stream by content ID, product ID and distribution ID for a given stream type.

Parameters in the path of the URL

Name	Type	Optional?	Description
“content_id”	String	No	Content ID as provided in the GET /events endpoint. The ID scheme “av:content:<integer>” must be used.
“product_id”	String	No	Product ID. The ID scheme “av:product:<integer>” must be used. Use GET /products endpoint to get the list of possible options.
“distribution_id”	String	No	Distribution ID. The ID scheme “av:distribution:<integer>” must be used. Use GET /products endpoint to get the list of possible options.
“stream_type”	String	No	Same as for GET /streams/<stream_id>/<stream_type> endpoint.

Parameters as query parameters in the URL

Same as for GET /streams/<stream_id>/<stream_type> endpoint.

Parameters as HTTP headers

Same as for GET /streams/<stream_id>/<stream_type> endpoint.

Response

Same as for GET /streams/<stream_id>/<stream_type> endpoint.

GET /event-statuses

HTTP method: GET
Path: /event-statuses

This endpoint returns an array with the list of possible event statuses.
See the section "Description of statuses" at the end of this document for more information.

GET /content-types

HTTP method: GET
Path: /content-types

This endpoint returns an array with the list of possible content types.

GET /products

HTTP method: GET
Path: /products

This endpoint returns an array with the list of possible products.

GET /distributions

HTTP method: GET
Path: /distributions

This endpoint returns an array with the list of possible distributions.

GET /stream-statuses

HTTP method: GET
Path: /stream-statuses

This endpoint returns an array with the list of possible stream statuses.
See the section "Description of statuses" at the end of this document for more information.

GET /device-categories

HTTP method: GET
Path: /device-categories

This endpoint returns an array with the list of possible device categories.

Description of statuses

Event statuses

Name	Description
Confirmed	The coverage of the event has been confirmed. Please check the status and the expected start time of the required stream to find out when Sportradar will make it available.
Cancelled	The event has been cancelled. Therefore, it will not be available for streaming.
Unknown	There is currently no public information on the event status. It has neither been planned nor cancelled.

Stream statuses

Please note that the stream status can change several times even during a live event. Streams might be available again shortly after an error has occurred.

Name	Description
Approved	The distribution of the stream has been confirmed. Live streams will be made available by Sportradar prior to the expected start time.
On Air Ready	The distribution of the stream has been prepared in Sportradar internal systems and will be publicly accessible in a short time. The playback URL of the stream cannot be retrieved at the moment.
On Air Broadcast	The distribution of the stream is available and publicly accessible. The playback URL of the stream can be retrieved.
Error	Due to an error, the distribution of the stream is currently interrupted.
Delivered	The distribution of the stream has ended and is not available any more.

Error responses

In case of errors, a response with the right HTTP code (4xx or 5xx) will be returned by the API and the body will contain an internal error code as well as a message with the description of the error.

Code	HTTP status	Message	Endpoints	Description
1000	404	The requested resource could not be found.	–	When there is no route matching the request.
1001	405	The requested method is not allowed. Allowed methods are: <allowedMethods>.	All	When the HTTP method is not allowed.
1002	500	The request failed due to an unexpected error.	All	When an unexpected error occurred.
2000	401	‘Authorization’ HTTP header is required.	All	When no “Authorization” HTTP header is passed.
2001	401	Invalid ‘Authorization’ HTTP header format (expected: Bearer <token>).	All	When the “Authorization” HTTP header format is not correct.
2002	401	Invalid token.	All	When the token is not valid.

2003	401	Invalid client.	All	When the client is not valid.
3000	400	Error with GET parameters. ‘from’ and ‘to’ parameters must be provided together or not provided at all.	GET /events	When “from” GET parameter is passed without “to” or vice-versa.
3001	400	Error with GET parameters. Bad format for ‘from’ or ‘to’ parameter. Expected ISO-8601 format (e.g. <nowInUTC>). Got: <sentParameter>.	GET /events	When at least one of “from” or “to” GET parameters does not have the ISO-8601 format.
3002	400	Error with GET parameters. The ‘to’ date /time must be greater than the ‘from’ date /time.	GET /events	When the date/time of the “to” GET parameter is not greater than the date/time of the “from” GET parameter.
3003	400	Error with GET parameters. Bad format for ‘<parameterName>’. Expected: <rightFormat>. Got: <sentParameter>.	All	When a GET parameter which is supposed to be passed with a certain format does not have that format.
3004	400	Error with parameters in the path. Bad format for ‘<parameterName>’. Expected: <rightFormat>. Got: <sentParameter>.	All	When a parameter in the path which is supposed to be passed with a certain format does not have that format.
3005	400	Error with parameters in the path. Invalid stream type.	Endpoints to get stream URL	When the stream type passed in the path is not one of the possible values.
3006	404	Stream not available.	Endpoints to get stream URL	When there is no stream available for the requested stream ID, the requested stream type and the given GET parameters.
3007	404	The requested stream type is not available for the requested stream.	Endpoints to get stream URL	When the stream type is not supported for the requested stream ID.
3008	403	Device category not booked.	Endpoints to get stream URL	When the stream was booked but not for the requested device category.
3009	400	Error with HTTP headers. IP must be sent in the ‘X-Real-IP’ HTTP header.	Endpoints to get stream URL	When the “X-Real-IP” HTTP header was not sent.
3010	400	Error with HTTP headers. The value passed in the ‘X-Real-IP’ HTTP header is not a valid IP.	Endpoints to get stream URL	When the IP sent in the “X-Real-IP” HTTP header is not a valid IPv4 or IPv6 IP.
3011	403	The country <countryCode> associated to the IP <ip> is not authorized for the requested stream.	Endpoints to get stream URL	When the country associated to the IP passed in the “X-Real-IP” HTTP header is not authorized for the requested stream and stream type.
3012	404	Event not available.	/events/<event_id>	When there is no event available for the requested event ID.
3014	400	Error with GET parameters. The device category passed in the “device_category” parameter is not supported for the requested stream type.	Endpoints to get stream URL	When the device category passed in the “device_category” GET parameter is not supported for the requested stream type.
3015	403	The IP <ip> is not authorized for the requested stream.	Endpoints to get stream URL	When the IP sent in the "X-Real-IP" HTTP header is not authorized.
3023	403	Stream URL generation not allowed.	Endpoints to get stream URL	When the stream is not allowed.

Back to top