

## Case Sensitivity

Stats API calls are case sensitive, and incorrect capitalization may result in unintended, incorrect, or empty returns. The Stats API uses the Lower Camel Case naming convention in which several words are joined together, the first letter of the string is lower case, and the following words have a capitalized first letter.

For example, the call <https://statsapi.mlb.com/api/v1/divisions?sportId=1&divisionId=200> will return information on the AL West.

```
"divisions": [
  {
    "id": 200,
    "name": "American League West",
    "nameShort": "AL West",
    "link": "/api/v1/divisions/200",
    "abbreviation": "ALW",
    "league": {
      "id": 103,
      "link": "/api/v1/league/103"
    },
    "sport": {
      "id": 1,
      "link": "/api/v1/sports/1"
    },
    "hasWildcard": false
  }
]
```

However, the seemingly identical call <https://statsapi.mlb.com/api/v1/divisions?sportId=1&divisionID=200> will return information for all MLB Divisions as if no specific divisionId were requested. The reason why this call differs from the first is divisionID=200, only "I" should be capitalized. The endpoint for division Id can only be called accurately by divisionId and not divisionID.

Although disregarding case sensitivity generates a different call than expected, it does not result in an error response. If only a section of the call is wrong, the Stats API ignores it and returns a default call with all sections that are correct. For example, by adding a correct leagueId to the incorrect "divisionID" call <https://statsapi.mlb.com/api/v1/divisions?divisionID=200&sportId=1&leagueId=103>, the Stats API still returns division

# Parameters

## Optional / Required Parameters

A parameter is further information added to the base/endpoint combination when making more complex or specific requests. There are two types of parameters: **Optional** and **Required**. The Stats API docs provide guidelines for what parameters are required and optional for each endpoint.

For example, in the following `Game-WinProbability` endpoint, the `gamePk` parameter is required since, without it, there is no information to return. `Timecode` is an optional parameter that allows the user to take an individual snapshot of data rather than the entire game's worth, but it is not required since the full game would be returned if excluded.



### TIP

- <https://statsapi.mlb.com/api/v1/game/531060/winProbability> - Full Game
- [https://statsapi.mlb.com/api/v1/game/531060/winProbability?timecode=20180803\\_182458](https://statsapi.mlb.com/api/v1/game/531060/winProbability?timecode=20180803_182458) - Snapshot

If **required** does not appear next to a parameter then it is optional.

## Query vs Path Parameters

Stats API calls can consist of query and/or path parameters. Path parameters are indicated by brackets, such as `{gamePk}`. Most of the path parameters in the Stats API are **required**, while a majority of query parameters are optional.

Below is an example of `Query` and `Path` parameter in use. While the `GamePk` is **required** as a path parameter to receive a return, the query parameter is only needed if calling a specific `timecode`. Without the timecode, the call will still be valid and return the information for the complete game.

Path parameter	Query parameter	Call with path and query parameters
game_pk: 531060	timecode: 20180803_182458	<a href="https://statsapi.mlb.com/api/v1/game/531060/boxscore?timecode=20180803_182458">https://statsapi.mlb.com/api/v1/game/531060/boxscore?timecode=20180803_182458</a>

# Requests

Requests are the processes of acquiring access and querying a back-end database. The Stats API retrieves data based on the user's request like a patron placing an order, or "request," at a restaurant. Users place requests via URL, and upon a valid request, the Stats API returns data in JSON format.

## Base URL

To begin, most Stats API calls, including basic game information and Statcast data, will flow through the base url:

<https://statsapi.mlb.com/api/v1/>. The GUMBO play-by-play feed is accessible via <https://statsapi.mlb.com/api/v1.1/>. However, in order to complete a valid request, an endpoint must be added to the base call.

To demonstrate a valid request, here is an example using the Statcast feed and the gameTypes endpoint:

- Stats API base call - <https://statsapi.mlb.com/api/v1/>
- Stats API endpoint - gameTypes

## Example

Example Valid Request - The endpoint above determines the specific information returned, while the base call determines from where information is retrieved. Valid request URLs are used in browsers for one-off information referencing, or coded into local programming software to automate information retrievals and ensure continuously updated user databases.

## Rate Limiting

The Stats API was designed to sustain a large volume of requests; however, Requests will be limited to 25 per second, and only those requests that exceed that rate will return a 429 response. Limits are on a per-second basis, and further requests can be made in the subsequent seconds even after the limit has been reached previously. 429 responses are reserved for rate limit exceptions for the Stats API.

information for the corresponding league based on its id. For further information about the correct capitalization of API calls, please consult the relevant Configs section to confirm proper capitalization of each endpoint.

# Fields, Limits, and Offsets

Depending on the specific request, the Stats API may return a large amount of data. For certain endpoints, users can manipulate the Stats API return through the use of three parameters: `fields`, `limits`, and `offsets`.

## Fields

Users may only want or need certain fields generated in the return. The Stats API includes the `fields` parameter to reduce payload and streamline the return to only include a user's desired fields.

The `fields` parameter is found in all Stats API endpoints. This parameter consolidates bulky requests into returns with specific information by processing data from the top level node (parent) down to attribute. For example, the return for the stats endpoint begins with the JSON stats object containing all child node objects, arrays, & strings. Therefore, the `fields` parameter in the call must contain stats. The best practice is to start the `fields` parameter with the parent node stats and proceed with further attributes in the child node in order to prevent confusion. For example, the 2022 MLB Players call can be specified to only return `fullName` and `id` across all entries by using the `fields` parameter:  
<https://statsapi.mlb.com/api/v1/sports/1/players?season=2022&fields=people,id,fullName>

In order to include more than one attribute in a return, include the parent-to-child node paths for all attributes. However, multiple field attributes that share a common parent node do not need to duplicate parent nodes in the call, and require only one specification in the `fields` parameter. For example, since `id` and `fullName` share common parent node people the parent node should only be called once as follows:  
<http://statsapi.mlb.com/api/v1/sports/1/players?season=2019&fields=people,id,fullName>

## Limit

Some endpoints' return is limited to a certain number of items per call in an effort to keep the Stats API return manageable without failing or timing out. Endpoints with this limitation, such as analytics/game and analytics/guids, are flagged as such in the Stats API documentation. In order to obtain a more comprehensive set of data, users can make multiple calls using the limit and offset parameters.

The `limit` parameter is used to return a subset of records from a given endpoint. To return only 10 games from the Statcast last updated endpoint, pass in a `limit` of 10. <http://statsapi.mlb.com/api/v1/analytics/game?limit=10>.

## Offset

While the `fields` parameter reduces the payload of a return, the `offset` parameter can be used in conjunction with the `limit` parameter to return a specific subset of data. The `offset` parameter returns  $i+1$  as the first record in a return. For example, without passing in a timestamp, the updated Statcast games endpoint <http://statsapi.mlb.com/api/v1/analytics/game> defaults to returning only the 1000 most recently updated games. In order to obtain a longer list of updated games from this endpoint, an `offset` must be passed. By setting the `offset` to 1000 <http://statsapi.mlb.com/api/v1/analytics/game?offset=1000> the Stats API call will return the next 1000 games (1001 to 2000).

## Limit & Offset

The best practice for data pagination is to use both `limit` and `offset` together <http://statsapi.mlb.com/api/v1/analytics/game?limit=100&offset=1000>. By adding a `limit`, users can cap the return from the API to properly fit to their parser while an `offset` will confirm that the user is returning all data available.

## Default Responses

When generating API calls within the Stats API, consumers must be aware of default responses. Although a call may be incorrect, there are instances where Stats API does not return an error but rather a fallback default response. In the User Guide section titled Case Sensitivity, it is mentioned that a simple case error could lead to a default call. At times, it may be difficult to recognize a default return due to the length or structure of the structure. In most cases, an invalid call only returns the information from a call using the required parameters for each relevant endpoint. The Stats API docs outline the required parameters for each endpoint, and is a useful resource to identify cases of a default return.



**TIP**

Arizona Diamondbacks Active roster for the 2000 season

- Correct call: <https://statsapi.mlb.com/api/v1/teams/109/roster?rosterType=Active>
- Error Call Resulting In Fallback: <https://statsapi.mlb.com/api/v1/teams/109/roster?rosterType=Active&season=20000>

In the above example, an extra "0" is added to the end of the season parameter value. Upon submitting this call, the return defaults to the current year's roster (the default call for this endpoint) since "20000" is an invalid year. The user should ensure the return is providing the intended information rather than expecting an error to populate upon submitting an invalid call.

The exceptions (returns that produce an error message rather than a default return) are cases that include invalid input within the required parameters. Since the required parameters generate the default return, an invalid input within those parameters will render the entire call invalid and produce an error.



**TIP**

Example: teams endpoint with required parameters

- Correct call: <https://statsapi.mlb.com/api/v1/teams/147/affiliates>
- Error Call: <https://statsapi.mlb.com/api/v1/teams/affiliates>

# Stats API Rate Limiting

## Rate Limiting

MLB enforces rate limiting for the Stats API to ensure high performance and prevent abuse. Requests are limited to 25 per second, and only those requests that exceed that rate will return a 429 response. Limits are on a per-second basis, and further requests can be made in the subsequent seconds even after the limit has been reached previously. 429 responses are reserved for rate limit exceptions for the Stats API.

Accounts may be temporarily blocked by MLB Information Security if consumers repetitively request new access tokens by incorrectly implementing OAuth2. As a reminder, access tokens are cached for 60 minutes, and new tokens should only be requested when the existing token expires. Refresh tokens do not expire unless they are unused for 14 days.

Please reach out to [VideoStatsSupport@mlb.com](mailto:VideoStatsSupport@mlb.com) with any questions or concerns.