C:\Users\CEINFO\AppData\Local\Microsoft\Windows\INetCache\Content.Word\MMILogo.png

Auto-suggest API

Project Atlas



November 22, 2020

Deepak

deepak.chawla@mapmyindia.co.in

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| **VERSION DETAILS** | | | | | | | | |
| DOCUMENT SUBMITTED BY: | | | Deepak | | REFERENCE DOCUMENT NAME: | | |  |
| DESCRIPTION | | | | | | | | |
| The document provides insights on to the upcoming project Atlas – Auto-Suggest API signature. It’s parameters, response codes and response structure. | | | | | | | | |
| **S. No.** | **Version / Revision no.** | **Date** | | **Revised Sections** | | **By** | **Remarks** | |
| 1 | - | 22/11/2020 | | Document created | | Deepak | Finalized API signature | |

Table of Contents

[Disclaimer 2](#_Toc493681993)

[Request Headers 3](#_Toc493682000)

[Security Type: 3](#_Toc493682001)

[Response Type 3](#_Toc493682002)

[Response Codes {as HTTP response code} 3](#_Toc493682003)

[Response Messages (as HTTP response message) 3](#_Toc493682004)

[URL 4](#_Toc493681994)

[Method 4](#_Toc493681995)

[Request Parameters 4](#_Toc493681996)

[a. Mandatory Parameters: 4](#_Toc493681997)

[b. Optional Parameters: 4](#_Toc493681998)

[Internal Parameters: 4](#_Toc493681999)

Claim Request Parameters...............................................................................................................................................4

[Response Parameters 6](#_Toc493682005)

[Sample Response 7](#_Toc493682006)

Claim [Sample Response](#_Toc493682006) **...........................................................................................................................................**8

# Disclaimer

1. The document contains sensitive information on parameters and responses that can be accessed only by MapmyIndia Products and not by any clients outside MapmyIndia. Kindly take prior permissions before sharing with any personnel outside MapmyIndia.

1. The document contains information on the overall API signatures and a few parameter functionalities may or may not be implemented as of this date but will be in the not too distant future. The parameters will be highlighted with turquoise colour.
2. The document contains restricted parameters as mentioned in point 1. These parameters will only be accepted if the key has access to the resource. Use this for providing more advanced type of API calls from inhouse products while the other 3rd or 2nd parties will not be able to send these parameters unless their key has access to it.

# **Autosuggest API**

## The Autosuggest API helps users to complete queries faster by adding intelligent search capabilities to your web or mobile app. This API returns a list of results as well as suggested queries as the user types in the search field.

MapmyIndia Autosuggest API Platform provides following advance features:

* POD
* Filter

### **POD**

It takes in the place type code which helps in restricting the results to certain chosen type. Parameter used is "pod".

Below mentioned are the codes for the pod:

**SLC**: SubLocality, **LC**:Locality, **CITY**: City, **VLG**:Village, **SDIST**: SubDistrict, **DIST**: District, **STATE**: State, **SSLC**: SubSubLocatlity.

### **Filter**

"filter" parameter helps you restrict the result by mentioning a bounded area, certain eloc (6 digit code to any poi, locality, city, etc.) or Pincode, below mentioned are the types:

* **pin\***: It takes pincode/postal code of an area {e.g. filter=pin:110055}
* **bounds**: lat1, lng1; lat2, lng2 (latitude, longitude) {e.g. filter=bounds: 28.598882, 77.212407; 28.467375, 77.353513}
* **cop**: {eloc} (string) {e.g. filter=cop:YMCZ0J}

### **Operators**

To send multiple keywords in a request, we’ve defined a couple of operators that can help the developers wrap their applications around this functionality. Below are the operators:

* The **;** Operator: This operator is an **OR** operator. Defining multiple keywords separated with a **;** would provide results that satisfies either of the keywords.
* The **$** Operator: This operator is an **AND** operator. Defining multiple keywords separated with a **$** would provide results that satisfy all the provided keywords. (default).
* The **,** operator: This operator is a seperator for using two or more **key:value** pairs and it has functionality of **AND($)**

To use these operators, simple just send in the keywords parameter like below:  
&keywords=coffee;tea$sea food;alcohol  
The above nearby search would yield in results that either provide coffee or tea but must provide either sea food or alcohol.

### **Request Headers**

The Atlas API leverages OAuth 2.0 based security. Hence, the developer would need to send a request for access token using their **client\_id** and **client\_secret** to the OAuth API. Once validated from the OAuth API, the access\_token and the token\_type need to be sent as Authorization header with the value: “{token\_type} {access\_token}”.

1. **Authorization**: “{token\_type} {access\_token}”.

### **Security Type:**

Atlas OAuth 2.0 based security using AES 256 and SHA-1.

### **Response Type**

* JSON: if the user passed in “/json”.
* XML: if the user passed in “/xml”

### **Response Codes {as HTTP response code}**

**Success:**

1. 200: To denote a successful API call.   
2. 204: To denote the API was a success but no results we’re found.

**Client-Side Issues:**

3. 400: Bad Request, User made an error while creating a valid request.   
4. 401: Unauthorized, Developer’s key is not allowed to send a request with internal parameters   
5. 403: Forbidden, Developer’s key has hit its daily/hourly limit

**Server-Side Issues:**

6. 500: Internal Server Error, the request caused an error in our systems.  
7. 503: Service Unavailable, during our maintenance break or server downtimes.

### **Response Messages (as HTTP response message)**

1. 200: Success.   
2. 204: No matches we’re found for the provided query.   
3. 400: Something’s just not right with the request.   
4. 401: Access Denied.   
5. 403: Services for this key has been suspended due to daily/hourly transactions limit.   
6. 500: Something went wrong.   
7. 503: Maintenance Break.

### **URL**

<https://atlas.mapmyindia.com/api/places/search>/json?

### **Method**

GET

### **Request Parameters**

The parameters are colour coded based on their requirements. The “Red” one’s are mandatory, “black” one’s are optional and the “**green**” one’s are claimbased.

**Please Note:** The Restricted parameters are also optional.

## **Mandatory Parameters:**

1. **query** (string) e.g. Shoes, Coffee, Versace, Gucci, H&M, Adidas, Starbucks, B130 {POI, House Number, keyword, tag}

## **Optional Parameters:**

1. **location** {string (latitude[double],longitude[double])} e.g. location=28.454,77.435
2. **zoom** (double): takes the zoom level of the current scope of the map (min: 4, max: 18).
3. **region** (string): it takes in the country code. LKA, IND, BTN, BGD, NPL for Sri-Lanka, India, Bhutan, Bangladesh, Nepal respectively. Default is India (IND)
4. **sort** (valueLess): Overrides the existing sorting for the API to that of the search priority. This is a temporary parameter and it’s not recommended to use it for production systems. This parameter is bound to change.

### c. **Internal Parameters:**

1. **pod** (string): Provides zoom level interaction with API results for specific view modes and it takes in the place type code which helps in restricting the results to certain chosen type.  
   Below mentioned are the codes for the pod –

* **SLC**: Sublocality
* **LC**: Locality
* **CITY**: City
* **VLG**: Village
* **SDIST**: Subdistrict
* **DIST**: District
* **STATE**: State
* **SSLC**: Subsublocality

1. **itemCount (integer): provides a configurable parameter to set number of results (default: 10, min: 1, max: 20)**
2. **bridge** (valueLess): provides a parameter to configure search suggestions in the API response. It’ll allow the API to provide link to a suggested nearby API search.
3. **explain** (valueless): provides insights on the results as to why have received the response you’ve reecived.

### **Claim Request Parameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Claims Available** |  |  |  |  |
|  | **Remarks** | **Claimed Type** | **Possible Values** | **isPublic** |
| **defaultEnabled** | integer This lets us enable / disable default ANT search when **searchFor** parameter is given. Contains two values 0, 1. By default, a user has access to MMI data in search. To prevent the user to access the MMI data, set the value of this claim to "0". Removing this claim or setting the value to 1 will allow users to access MMI data. **Note** : In autosuggest, without claim, it is not resulting EVs in the response | Valued | 0,1 | N |
| **searchFor** | string For searching within specific datasets: like Honda or EV charging stations; depends on **defaultEnabled** | Valued | EV, Honda | N |
| **region** | string | Valued | lka,mmr,npl,btn,bgd,ind | Y |
| **smartBehavior** | The claim will act as a Boolean to enable the Autosuggest API to switch to nearby search. If no location is provided and the corona keyword is given, the search will exhibit as normal autosuggest behavior. No smart behavior will be exhibited. | valueless |  | Y |
| **allowedCategories(list)** | This claim will have a list of categories allowed to a user. Any given input category that is not in this list will be ignored and hence not searched. | Valued | Category codes | Y |
| **mapCenter** | If the user provides both parameters, **mapCenter**(Bengaluru) and **location**(current location, say MMI), and searches for **coffee**. The **suggestedLocations** will contain results from Bengaluru and in **suggestedSearches**, the suggestion **"coffee near me"**will have the location reference of the user's location. | Valued |  | Y |
| **isElocSearch** | Search can be done by passing a single eLoc and can be used to enable and disable eLoc based queries. By default, the eLoc based search is disabled for all the users. | valueless |  | Y |
| **outputFieldsList[list]** | claim can be provisioned from Anchor to control the output fields and Contains the list of fields to be displayed in the output. The fields can be selected from a pre-populated set only. | Valued |  | Y |
| **isPartnerSearch[list]** | A marker claim to switch on/off the partner search for a user. | Valued | maruti ev | Y |
| **partnerIds[List]** | This claim can be used to set the partnerIds to be allowed to a user. The values in this claim can be selected from a pre-populated set of values. | Valued |  |  |
| **tokenizeAddress** |  | Valueless | --- | Y |
|  |  |  |  |  |
| **Claims Proposed** | **Remarks** | **Possible Claims** | |  |
|  |  | **Claimed Type** | **Possible Values** | **isPublic** |
| **query** | string | Not applicable | Not applicable | Y |
| **location** | string; **e.g.** location=28.454,77.435 | Valued | true,false | Y |
| **zoom** | integer (min: 4, max: 18) | Valued | true,false | Y |
| **pod** | String Possible values:  SLC: Sublocality LC: Locality CITY: City VLG: Village SDIST: Subdistrict DIST: District STATE: State SSLC: Subsublocality | Valued | SLC,LC,CITY,VLG,SDIST,DIST,STATE,SSLC | Y |
| **filter** | String **filter=pin**:{PIN} {e.g. filter=pin:110055} **filter=bounds**: lat1, lng1; lat2, lng2 (North West, South East) {e.g. filter=bounds: 28.598882, 77.212407; 28.467375, 77.353513} **filter=cop**:{eloc} (string) {e.g. filter=cop:YMCZ0J} | Valued | pin,bounds,eLoc | Y |
| **bridge** | bridges autosuggest with nearby API to provide the suggestedSearches object in response | valueless | true,false | N |
| **explain** | Provides explanation why the result was provided. | valueless | true,false | N |

### **Response Parameters**

1. **suggestedLocations** ([object])
2. **type** (string): type of location POI or Country or City
3. **typeX** (string):
4. **placeAdress** (string): Address of the location
5. **latitude** (string): Latitude of the location
6. **longitude** (string): longitude of the location
7. **eloc (string):** Place Id of the location 6-char alphanumeric
8. **entryLatitude** (number($double)): latitude of the entrance of the location
9. **entryLongitude** (number($double)): longitude of the entrance of the location
10. **placeName** (string): Name of the location
11. **alternateName** (string):
12. **keywords** (object): contains the category code to which the POI result(if applicable) belongs to
13. **addressTokens** (object):

* **houseNumber** (string): house number of the location.
* **houseName** (string): house name of the location.
* **poi** (string): name of the POI (if applicable)
* **street** (string): name of the street. (if applicable)
* **subSubLocality** (string): the sub-sub-locality to which the location belongs. (if applicable)
* **subLocality** (string): the sub-locality to which the location belongs. (if applicable)
* **locality** (string): the locality to which the location belongs. (if applicable)
* **village** (string): the village to which the location belongs. (if applicable)
* **subDistrict** (string): the sub-district to which the location belongs. (if applicable)
* **district** (string): the district to which the location belongs. (if applicable)
* **city** (string): the city to which the location belongs. (if applicable)
* **state** (string): the state to which the location belongs. (if applicable)
* **pincode** (string): the PIN code to which the location belongs. (if applicable)
* **P** (integer): **For internal use only**
* **orderIndex** (integer): The order where this result should be placed
* **score (number($double):** For internal use only
* **suggester** (string): For internal use only
* **richInfo** (string):

1. **suggestedSearches** ([object])
2. **keyword** (string): what the user typed in.
3. **Identifier** (string): what did the API use for it to qualify it as a suggested search request
4. **location** (string): the name of the location to which the nearby will run in context to.
5. **hyperlink** (string): the ready-made link for the nearby API pre-initialized with all default parameters and location with code to search for.

### 

### **Sample Response**

**Input URL:** https://atlas.mapmyindia.com/api/places/search/json?access\_token=<>&query=Golden Heights Mall, Rajaji Nagar, Bengaluru&location=12.9828, 77.5598

**Response:**

{

"suggestedLocations": [

{

"type": "POI",

"typeX": 7,

"placeAddress": "No 1/2, 4th Main Block, Dr Rajkumar Road, Manjunath Nagar, Rajaji Nagar, Bengaluru, Karnataka, 560010",

"latitude": 12.982834,

"longitude": 77.5598560000001,

"eLoc": "Q9KRT7",

"entryLatitude": 12.9832440000001,

"entryLongitude": 77.560167,

"placeName": "Golden Heights",

"alternateName": "Golden Heights Mall,Westgate Value Mall",

"keywords": [

"SHPMAL"

],

"addressTokens": {},

"p": 984,

"orderIndex": 1,

"score": 3114.1915290172956,

"suggester": "alternateName",

"richInfo": {}

}

### **Claim Sample Response**

**smartBehavior**:- if allowed autosuggest will be switched in nearby mode for the given categories in allowedCategories claim.

**Input URL:** [https://atlasweb.mapmyindia.in/api/v4.4/places/search/json? access\_token=1aa0e659-d16a-45f4-b3c2-f88fb27d4f9a&location=28.5507160000001,77.2689280000001&query=corona testing](https://atlasweb.mapmyindia.in/api/v4.4/places/search/json?%20access_token=1aa0e659-d16a-45f4-b3c2-f88fb27d4f9a&location=28.5507160000001,77.2689280000001&query=corona%20testing%20)

**Response:**

{

    "suggestedLocations": [

        {

            "type": "POI",

            "typeX": 7,

            "placeAddress": "Sukhdev Vihar Road, Sukhdev Vihar, New Delhi, Delhi, 110025",

            "latitude": 28.554093,

            "longitude": 77.2741350000001,

            "eLoc": "5FD0BP",

            "entryLatitude": 28.5550260000001,

            "entryLongitude": 77.2738540000001,

            "placeName": "CSIR IGI",

            "alternateName": "",

            "keywords": [

                "SHPCOM"

            ],

            "addressTokens": {},

            "p": 8060,

            "orderIndex": 1,

            "score": 0.0,

            "suggester": null,

            "richInfo": {}

        }

**searchFor**:- For searching within specific datasets: like Honda or EV charging stations; depends on **defaultEnabled**. defaultEabled contain two value 0 and 1. Here is an example of searchFor parameter with defaultEnabled contains value 1.

**Input URL:**  https://atlas.mapmyindia.com/api/places/search/json?access\_token=<>&query=Hyundai charging&location=13.115541, 77.607842&searchFor=

**Response:**

{

"suggestedLocations": [

{

"type": "POI",

"typeX": 7,

"placeAddress": "No 65/1, Venkatala Village Yelahanka, KIAL Road, Uttarahalli Hobli, Bengaluru, Karnataka, 560063",

"latitude": 13.115393,

"longitude": 77.608121,

"eLoc": "DUOJI7",

"entryLatitude": 13.115629,

"entryLongitude": 77.607714,

"placeName": "Hyundai Electric Vehicle Charging Station",

"alternateName": "Hyundai EV Charging Station",

"keywords": [

"TRNECS"

],

"addressTokens": {},

"p": 12000,

"orderIndex": 1,

"score": 275.65111318320015,

"suggester": "alternateName"

}

|  |
| --- |
| **mapCenter**:- If the user provides both parameters, **mapCenter**(Bengaluru) and **location**(current location, say MMI), and searches for **coffee**. The **suggestedLocations** will contain results from Bengaluru and in **suggestedSearches**, the suggestion **"coffee near me"**will have the location reference of the user's location.  **Input URL:**  <https://atlas.mapmyindia.com/api/places/search/json?bridge&query=coffee&mapCentre=12.9734040000001,77.613703&location=28.5507160000001,77.2689280000001>  **Response:**  {      "suggestedLocations": [          {              "type": "POI",              "typeX": 7,              "placeAddress": "Mahatma Gandhi Main Road, Bengaluru, Karnataka, 560025",              "latitude": 12.973803,              "longitude": 77.6127520000001,              "eLoc": "DDS88B",              "entryLatitude": 12.9739250000001,              "entryLongitude": 77.6126850000001,              "placeName": "Coffee Shop",              "alternateName": "",              "keywords": [                  "FODCOF"              ],              "addressTokens": {},              "p": 11634,              "orderIndex": 1,              "score": 1402.1833112096274,              "suggester": "placeName",              "richInfo": {}          } |

**outputFieldsList**:- list of output fields that can be displayed to the user.

**Input URL:** <https://atlasweb.mapmyindia.in/api/v4.4/places/search/json?query=mapmyindia&location=28.5507160000001,77.2689280000001>

**Response:**

{

    "suggestedLocations": [

        {

            "placeAddress": "237, Okhla Industrial Estate Phase 3, Near Modi Mill, New Delhi, Delhi, 110020",

            "eLoc": "MMI000",

            "placeName": "MapmyIndia Head Office New Delhi",

            "orderIndex": 1

        },

        {

            "placeAddress": "Block B Auto Market, Sector 16, Noida, Uttar Pradesh, 201301",

            "eLoc": "W52NU3",

            "placeName": "MapmyIndia Navigator Dealer",

            "orderIndex": 2

        }

**partnerIds**:- This claim can be used to set the partnerIds to be allowed to a user. The parameter named isPartnerSearch is of data type list.The values in this claim can be selected from a pre-populated set of values. For the example PartnerIds is 2. The list of PartnerIds is :

* 1 -- MMI
* 2 : MSIL EV
* 3 : Highway Delight
* 4: Web SDK

**Input URL:** https://atlas.mapmyindia.com/api/places/search/json?access\_token=636165d7-862a-4c67-b400-47aaec6b0ec5&query=maruti ev&location=28.49728368,77.07214415

**Response:**

|  |
| --- |
| { |
| "suggestedLocations": [ |
| { |
| "type": "POI", |
| "typeX": 7, |
| "placeAddress": "Old Palam Gurgaon Road", |
| "latitude": 28.497283676540597, |
| "longitude": 77.07214415073395, |
| "eLoc": "AGRTQV", |
| "entryLatitude": 0, |
| "entryLongitude": 0, |
| "placeName": "Ensto AC GUR0005", |
| "alternateName": "", |
| "keywords": [ |
| "TRNECS" |
| ], |
| "addressTokens": {}, |
| "p": 12000, |
| "orderIndex": 1, |
| "score": 0, |
| "suggester": null |
| } |