

بسم الله الرحمن الرحيم

راهنمای web مازول IP2Location

مرداد ۹۷

## فصل ۱

### ۱-۱ نگارش

در جدول زیر تاریخچه نگارش های مختلف ماژول آورده شده است.

تاریخ	نسخه	توضیحات
تیر ۹۷	۱,۰	نسخه اولیه document آماده شد.

### ۲-۱ توضیحات اولیه

این ماژول جهت دریافت مشخصات یک IP از طریق ارسال یک درخواست get به api مورد نظر نوشته شده است. این ماژول دارای دو متد اصلی به نام های get\_ip\_info و ip\_is\_proxy می باشد. توضیحات بیشتر در مورد متود های این ماژول در زیر آورده شده است. کد مربوط به این ماژول ۱۱۵ می باشد. [جدول ماژول ها](#).

### ۳-۱ متد get\_ip\_info()

این متد دو آرگومان ip و API\_KEY را به عنوان ورودی دریافت می کند و یک دیکشنری از مشخصات ip وارد شده را به عنوان خروجی برمی گرداند. پارامترهای ورودی و خروجی و همچنین نمونه خروجی این متد در زیر قابل مشاهده است.

ورودی	توضیحات
Method_id	شماره تابع. مقدار این پارامتر برای این متد برابر ۱ می باشد
ip	ip مورد نظر جهت دریافت مشخصات آن
Api_key	جهت اتصال به سرویس مورد نظر نیاز به API_KEY آن می باشد.

جدول مربوط به ورودی متد get\_ip\_info ()

method	Output_parameters
get_ip_info()	<pre>{   "properties": [     {       "country_name": "country_name",       "type": 11     },     {       "city_name": "city_name",       "type": 11     },     {       "latitude": "latitude",       "type": 0     },     {       "longitude": "longitude",       "type": 0     }   ],   "special_properties": [     {       "region_name": "region_name",</pre>

```

"type": 11
},
{
"zip_code": "zip_code",
"type": 0
},
{
"time_zone": "time_zone",
"type": 0
},
{
"isp": "isp",
"type": 11
},
{
"domain": "hetzner.de",
"type": 12
},
{
"idd_code": "idd_code",
"type": 0
},
{
"area_code": "area_code", "type": 0
},
{
"mobile_brand": "mobile_brand",
"type": 11
},
{
"elevation": "elevation",
"type": 0
},
{
"usage_type_desc": "usage_type + Please Read Document",
"type": 0
},
{
"mcc": "mcc",
"type": 0
},
{
"mnc": "mnc",
"type": 0
},
{
"net_speed": "net_speed",
"type": 0
}
],
"results": [
{
"geo_coordinates": " latitude, longitude ",
"type": 14
}
]
}

```

جدول مربوط به پارامترهای خروجی متد get\_ip\_info()

نمونه خروجی کامل این ماژول در این [لینک](#) موجود است.

#### ۴-۱ متد ip\_is\_proxy()

این متد دو آرگومان ip و API\_KEY را به عنوان ورودی دریافت می کند و یک دیکشنری از مشخصات ip وارد شده را به عنوان خروجی برمی گرداند. پارامترهای ورودی و خروجی و همچنین نمونه خروجی این متد در زیر قابل مشاهده است.

ورودی	توضیحات
Method_id	شماره تابع. مقدار این پارامتر برای این متد برابر ۲ می باشد
ip	ip مورد نظر جهت دریافت مشخصات آن
api_key	جهت اتصال به سرویس مورد نظر نیاز به API_KEY آن می باشد.

جدول مربوط به ورودی متد ip\_is\_proxy()

method	Output_parameters
ip_is_proxy()	<pre>{   "properties": [],   "special_properties": [     {       "country_name": "countryName",       "type": 11     },     {       "region_name": "regionName",       "type": 11     },     {       "city_name": "cityName",       "type": 11     },     {       "isp": "isp",       "type": 11     },     {       "proxy_type_desc": "proxyType + Please Read Document",       "type": 0     },     {       "is_proxy": "isProxy",       "type": 0     }   ],   "results": [] }</pre>

جدول مربوط به پارامترهای خروجی متد ip\_is\_proxy()

نمونه خروجی کامل این ماژول در این [لینک](#) موجود است.

#### ۵-۱ متد ip\_info\_balance()

این متد یک آرگومان API\_KEY را به عنوان ورودی دریافت می کند و میزان بالانس API مربوط به ip\_info را در خروجی می دهد. پارامترهای ورودی و خروجی و همچنین نمونه خروجی این متد در زیر قابل مشاهده است.

ورودی	توضیحات
Method_id	شماره تابع. مقدار این پارامتر برای این متد برابر ۳ می باشد
api_key	جهت اتصال به سرویس مورد نظر نیاز به API_KEY آن می باشد.

جدول مربوط به ورودی متد ip\_info\_balance()

method	Output_parameters
ip_is_proxy()	{ "results": [{"ip_info_balance": 0, "type": 0}] }

جدول مربوط به پارامترهای خروجی متد ip\_info\_balance()

#### ۶-۱ متد ip\_is\_proxy\_balance()

این متد یک آرگومان API\_KEY را به عنوان ورودی دریافت می کند و میزان بالانس API مربوط به ip\_is\_proxy را در خروجی می دهد. پارامترهای ورودی و خروجی و همچنین نمونه خروجی این متد در زیر قابل مشاهده است.

ورودی	توضیحات
Method_id	شماره تابع. مقدار این پارامتر برای این متد برابر ۴ می باشد
api_key	جهت اتصال به سرویس مورد نظر نیاز به API_KEY آن می باشد.

جدول مربوط به ورودی متد ip\_is\_proxy\_balance()

method	Output_parameters
ip_is_proxy()	{ "results": [{"ip_is_proxy_balance": "0", "type": 0}] }

جدول مربوط به پارامترهای خروجی متد ip\_is\_proxy\_balance()

#### ۷-۱ محتوای موجود در درخواست get

برای استفاده از این ماژول می بایست یک درخواست از نوع get جهت دریافت اطلاعات به api مورد نظر به صورت زیر ارسال شود. بدنه درخواست ارسالی شامل پارامترهای data و type است. پارامتر type مشخص کننده نوع ورودی ماژول (unstructue,file, domain,path,...) است که id مربوط به انواع مختلف ورودی ها و توضیحات مربوطه که در این [لینک](#) موجود است، در این فیلد قرار می گیرد. پارامتر data هم همان داده ارسالی است که شامل پارامترهای ورودی ماژول می باشد.

```
{
  "data":{"IP":" 185.81.98.126", "API_KEY":" C571380781"},
  "type":0
}
```

نمونه body یک socket request

در تصویر نمونه بالا داریم:

Data: داده ارسالی برای استفاده در ماژول که شامل IP و API\_KEY می باشد.  
Type: نوع ورودی ماژول را مشخص می کند که می تواند از نوع unstructured,path,file و یا انواع دیگر باشد و باید id ورودی موردنظر در این قسمت قرار گیرد.

#### ۸-۱- جدول خطاها

جدول زیر شامل خطاهایی است که ماژول IP2Location در پاسخ به کلاینت بر می گرداند. خطا ها با شماره مشخصی نشان داده می شوند.

توضیحات	نام	کد
زمانی که فرمت داده ورودی اشتباه باشد.	InvalidInputError	102
زمانی که کلید API اشتباه باشد.	WrongApikeyError	104
زمانی که شبکه دچار تداخل شود.	NetworkError	107
زمانی که نتیجه ای برگشت داده نشود.	resultNotFoundError	114