

بسم الله الرحمن الرحيم

راهنمای developer مازول whois

تیر ۹۷

فهرست مطالب

۳	۱-۱ نگارش ها
۳	۲-۱ توضیحات اولیه
۴	۳-۱ کلاس whois
۴	۱-۳-۱ متد get_whois_with_zone()
۴	۴-۱ کلاس whois_api
۴	۱-۴-۱ متد get_result()
۵	۵-۱ کلاس parser
۵	۱-۵-۱ متد get_main_structure()
۵	۲-۵-۱ متد get_emails()
۶	۳-۵-۱ متد get_phones()
۶	۴-۵-۱ متد get_names()
۶	۵-۵-۱ خروجی نهایی parser
۱۰	۶-۱ متد whois_reverse_by_name
۱۰	۷-۱ متد whois_reverse_by_email
۱۱	۸-۱ متد whois_history
۱۱	۹-۱ متد whois_get_balance
۱۱	۱۰-۱ متد reverse_ip_lookup
۱۲	۱۱-۱ محتوای موجود در socket request
۱۳	۱۲-۱ جدول خطاها
۱۳	۱۳-۱ جدول method_id

فصل ۱

۱-۱ نگارش ها

توضیحات	نسخه	تاریخ
نسخه اولیه document آماده شد.	۱.۰	اسفند ۹۶
<ul style="list-style-type: none"> • اضافه شدن name به عنوان entity در خروجی • تغییر و اصلاح property ها 	1,1	فروردین ۹۷
<ul style="list-style-type: none"> • به روز رسانی خروجی ها • استفاده از api برای domain های غیر از .ir • ارایه progressbar 	۱.۲	تیر ۹۷
<ul style="list-style-type: none"> • اضافه کردن متود reverse_ip_lookup 		آذر ۹۷

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

۲-۱ توضیحات اولیه

این ماژول جهت استخراج اطلاعات مربوط به whois یک domain به صورت parse شده نوشته شده است. این ماژول به دو طریق به درخواست ها پاسخ می دهد. ۱- استفاده از api آماده ۲- استفاده از نسخه قدیمی ماژول. در حال حاضر تمامی درخواست ها از طریق api پاسخ داده می شوند. آدرس این api آماده www.whoxy.com می باشد که به عنوان ورودی دامنه و api_key را دریافت می کند و در خروجی یک فایل json ارایه می دهد. این api از دامنه های .ir پشتیبانی نمی کند بنابراین درخواست های .ir به نسخه قدیمی whois تحویل داده می شود که به صورت عادی اطلاعات domain را استخراج می کرد. با وجود استخراج اطلاعات whois از دو منبع مختلف. در این نسخه از ماژول خروجی هر دو یکسان است که در بخش های بعدی نمونه آن موجود است. در نسخه ی عادی whois دو کد اصلی است به نام های whois و parser می باشد. کلاس مربوط به کد whois جهت به دست آوردن اطلاعات whois یک دامنه در این ماژول قرار گرفته و کلاس مربوط به parser جهت parse کردن اطلاعات استخراج شده توسط کلاس whois به شکل entity_property نوشته شده است. در نهایت خروجی این ماژول که لیستی از اطلاعات مختلف در قالب entity_property هستند، برای کاربر ارسال می شود. در نسخه api یک کلاس whois_api وجود دارد که دامنه و api_key و نهایتاً پس از دریافت اطلاعات از whoxy.com خروجی را به صورت مشابه با نسخه عادی whois، parse می کند. در قسمت های بعدی به طور مفصل راجع به هر کلاس و متد های آنها صحبت خواهد شد. همچنین یک نمونه خروجی کامل مربوط به این ماژول برای دامنه "time.ir" در این [لینک](#) موجود است و همچنین یک نمونه خروجی کامل مربوط به این ماژول برای دامنه "time.com" که مربوط به api می باشد در این [لینک](#) موجود است.

نکته: در صورتی که api نتواند به درخواست پاسخ دهد، درخواست به ورژن قدیمی این متد داده می شود. (مانند دامنه های .ir)

۳-۱-۳-۱ whois کلاس

همانطور که در قسمت توضیحات گفته شد ، این کلاس جهت استخراج اطلاعات whois یک domain که ir است نوشته شده است. این کلاس دارای یک متد به نام () get_whois_with_zone بوده که اطلاعات مربوط به whois دامنه را به دست آورده و در اختیار کلاس parser قرار می دهد. در قسمت بعدی راجع به تنها متد این کلاس در این مازول توضیح خواهیم داد.

۱-۳-۱-۱ متد get_whois_with_zone()

این متد جهت استخراج اطلاعات یک domain به کار می رود. ورودی را که به صورت domain است دریافت کرده و اطلاعات مربوط به whois این domain را که کلی و parse نشده است بر می گرداند. جدول مربوط به ورودی و یک نمونه خروجی این مازول در زیر قرا گرفته است. (نسخه قدیمی مازول)

ورودی	توضیحات
domain	دومین دریافت شده به عنوان ورودی
Method_id	شناسه ی تابع مورد نظر را مشخص می کند و شناسه این متد برابر ۳ است.

جدول مربوط به ورودی متد () get_whois_with_zone

```
["% This is the IRNIC Whois server v1.6.2.\n% Available on web at http://whois.nic.ir/\n% Find the terms and conditions of use on http://www.nic.ir/\n% \n% This server uses UTF-8 as the encoding for requests and responses.\n\n% NOTE: This output has been filtered.\n\n% Information related to 'time.ir'\n\nndomain:\t\ttime.ir\nnascii:\t\ttime.ir\nnremarks:\t\t(Domain Holder) Hamidreza Taherian\nnremarks:\t\t(Domain Holder Address) No 48.9, Hosseini Rad St, Valiasr Ave. Tehran, IRAN., Tehran, Tehran, IR\nnholder-c:\t\t76-irnic\nnadmin-c:\t\t76-irnic\nntech-c:\t\t76-irnic\nnserver:\t\t2.parsihost.com\nnserver:\t\t1.parsihost.com\nnlast-updated:\t\t2017-03-16\nnexpire-date:\t\t2022-03-04\nnsource:\t\tIRNIC # Filtered\n\nnnic-hdl:\t\t76-irnic\nnperson:\t\tHamidreza Taherian\nne-mail:\t\tthamid@radcom.co\nnaddress:\t\tNo 48.9, Hosseini Rad St, Valiasr Ave. Tehran, IRAN., Tehran, Tehran, IR\nnphone:\t\t88926505\nnfax-no:\t\t88926986\nnsource:\t\tIRNIC # Filtered\n\n"]
```

نمونه خروجی متد مربوط به دامنه "time.ir"

۴-۱-۴-۱ whois_api کلاس

همانطور که در قسمت توضیحات گفته شد ، این کلاس جهت استخراج اطلاعات whois یک domain که غیر از ir است نوشته شده است. این کلاس دارای یک متد به نام () get_result بوده که اطلاعات مربوط به whois دامنه را به دست آورده و در اختیار کلاس parser قرار می دهد. در قسمت بعدی راجع به تنها متد این کلاس در این مازول توضیح خواهیم داد. نکته: در صورتی که api نتواند به درخواست پاسخ دهد، درخواست به ورژن قدیمی این متد داده می شود. (مانند دامنه های ir)

۱-۴-۱-۱ متد get_result()

این متد جهت استخراج اطلاعات یک domain به کار می رود. ورودی را که به صورت domain است دریافت کرده و اطلاعات مربوط به whois این domain را از api مربوط به سایت whoxy.com است می گیرد و به صورت json بر می گرداند. اما از آن جایی که خروجی مورد نظر از این مازول با خروجی این api متفاوت است. خروجی تحویل parser داده می شود تا به شکل خواسته شده در بیاید. جدول مربوط به ورودی و یک نمونه خروجی این مازول در زیر قرا گرفته است.

ورودی	توضیحات
domain	دامنه دریافت شده به عنوان ورودی
api_key	کلید api که باید معتبر باشد.
Method_id	شناسه‌ی تاع مورد نظر را مشخص می کند و شناسه این متد برابر ۳ است.

جدول مربوط به ورودی متد () get_result

```
{
  "status": 1,
  "domain_name": "itieee.info",
  "query_time": "2018-07-15 08:03:46",
  "whois_server": "whois.afiliat.net",
  "domain_registered": "yes",
  "create_date": "2017-12-04",
  "update_date": "2018-03-12",
  "expiry_date": "2018-12-04",
  "domain_registrar": {...}
}
```

قسمتی از خروجی متد مربوط به دامنه "itieee.info"

۱-۵-۵ کلاس parser

همانطور که در قسمت توضیحات گفته شد، این کلاس جهت parse کردن اطلاعات مربوط به خروجی کلاس whois به کار می رود. این کلاس دارای سه متد بوده و محتوای whois را دریافت کرده و با استفاده از این سه متد اطلاعات را در بخش های مجزا و parse شده به عنوان خروجی بر می گرداند. در بخش بعدی راجع به کار متدهای آن و همچنین ورودی و خروجی آنها صحبت خواهیم کرد. دقت شود کلاس parser برای هر دو قسم whois و whois_api یکسان است.

۱-۵-۱ متد get_main_structure()

این متد محتوای whois استخراج شده را به عنوان ورودی می گیرد و خروجی را به شکل entity_property در قالب json بر می گرداند. Data ای که در خروجی این ماژول قرار دارد به نام unstructured_data می شناسیم. ورودی این ماژول یا محتوای whois استخراج شده است که در بخش های قبلی [نمونه](#) آن را دیدیم. پارامترهای خروجی این متد در زیر قابل مشاهده است.

۱-۵-۲ متد get_emails()

این متد محتوای whois استخراج شده را به عنوان ورودی می گیرد و خروجی آن که لیستی از ایمیل ها ست را به شکل entity_property و در قالب json بر می گرداند. ورودی این ماژول همان محتوای whois استخراج شده است که در بخش های قبلی نمونه آن را دیدیم. توضیحات در مورد entity_property ها در این [لینک](#) موجود است.

۳-۵-۱ متد get_phones()

این متد محتوای whois استخراج شده را به عنوان ورودی می گیرد و خروجی آن که لیستی از تلفن ها ست را به شکل entity_property و در قالب json بر می گرداند. ورودی این مازول همان محتوای whois استخراج شده است که در بخش های قبلی نمونه آن را دیدیم. توضیحات در مورد entity_property ها در این [لینک](#) موجود است.

۴-۵-۱ متد get_names()

این متد محتوای whois استخراج شده را به عنوان ورودی می گیرد و خروجی آن که لیستی از نام ها ست را به شکل entity_property و در قالب json بر می گرداند. ورودی این مازول همان محتوای whois استخراج شده است که در بخش های قبلی نمونه آن را دیدیم. توضیحات در مورد entity_property ها در این [لینک](#) موجود است.

۵-۵-۱ خروجی نهایی parser

خروجی نهایی ماژول whois_api پس از استخراج قسمت های مختلف و پارس کردن به شکل زیر خواهد بود. (خروجی whois نیز دقیقاً به شکل زیر است).

method	Output_parameters
Whois_api.get_result()	<pre>{ "properties": [{ "dns": "Name Servers", "type": 3 }, { "registrant_name": "name", "type": 11 }, { "admin_name": "name", "type": 11 }, { "tech_name": "name", "type": 11 }, { "registration_date": "date", "type": 0 }, { "expiration_date": "date", "type": 0 }], "special_properties": [{ "should_be_fixed": false, "type": 0 }, { "has_dns": true,</pre>

```

    "type": 0
  },
  {
    "valid": false,
    "type": 0
  }
],
"results": [
  {
    "type": 0,
    "data": {
      "registrant": {
        "city": "Tehran",
        "country": "IR",
        "name": "name",
        "fax": "12345678",
        "street": "No 48.9",
        "phone": "12345678",
        "email": "info@domain.com"
      },
      "admin": {
        "city": "Tehran",
        "country": "IR",
        "name": "name",
        "fax": "12345678",
        "street": "No 48.9",
        "phone": "12345678",
        "email": "info@domain.com",
        "type": "whois_contact_name"
      },
      "tech": {
        "city": "Tehran",
        "country": "IR",
        "name": "name",
        "fax": "12345678",
        "street": "No 48.9",
        "phone": "12345678",
        "email": "info@domain.com",
        "type": "whois_contact_name"
      }
    },
    "ref": {
      "task": "whois",
      "whois_for": "domain.com",
      "whois_from": ""
    }
  },
  {
    "data": "+98.22222222",
    "type": 4,
    "properties": [
      {
        "location": "",
        "type": 11
      },
      {
        "owner": "",
        "type": 11
      }
    ]
  }
]

```

```

    }
  ],
  "special_properties": [
    {
      "operator": "",
      "type": 0
    },
    {
      "phone_type": "",
      "type": 0
    },
    {
      "is_valid": true,
      "type": 0
    },
    {
      "country_code": "",
      "type": 0
    }
  ],
  "ref": {
    "task": "whois",
    "whois_for": "domain.com",
    "label": "tech_name",
    "whois_from": ""
  }
},
{
  "data": "info@domain.com",
  "type": 2,
  "is_valid": true,
  "properties": [
    {
      "local_address": "",
      "type": 5
    },
    {
      "domain_name": "",
      "type": 12
    },
    {
      "owner": "",
      "type": 11
    },
    {
      "organization": "",
      "type": 11
    }
  ],
  "special_properties": [
    {
      "is_valid": true,
      "type": 0
    }
  ],
  "ref": {
    "task": "whois",
    "whois_for": "domain.com",

```



```

"label": "tech_name",
"whois_from": ""
},
{
"data": "name",
"type": 11,
"properties": [
{
"fax": "",
"type": 4
},
{
"phone": "",
"type": 4
},
{
"email": "",
"type": 2
},
{
"city": "",
"type": 11
},
{
"country": "",
"type": 11
},
{
"street": "",
"type": 8
}
],
"special_properties": [
{
"is_username": true,
"type": 0
},
{
"is_domain_name": true,
"type": 0
},
{
"is_public_name": true,
"type": 0
},
{
"is_account_name": true,
"type": 0
}
],
"ref": {
"task": "whois",
"whois_for": "domain.com",
"label": "tech_name",
"whois_from": ""
}
}

```

۶-۱ متد whois_reverse_by_name

این متد دو آرگومان به نام `api_key` و `name` را دریافت می کند و خروجی که شامل کلیه اطلاعات `whois` ثبت شده با اسم ارسال شده را در خروجی برمی گرداند. این متد از `api` استفاده می کند بنابراین ارسال `api_key` معتبر اجباری می باشد. از آن جایی که در `api` مربوط به `whoxy`، دامنه های `ir` وجود ندارد، بنابراین با ارسال نام هایی که دامنه های `ir` ثبت کرده اند به این متد، خروجی دریافت نخواهد شد.

ورودی	توضیحات
method_id	شناسه تابع، برای این تابع برابر ۲ است
Api_key	یک <code>api_key</code> معتبر سایت <code>whoxy</code>
name	اسم مورد جست و جو در <code>whois</code>

جدول مربوط به ورودی متد `Reverse_whois_by_name`

نمونه خروجی این متد در [لینک](#) زیر موجود است.

۷-۱ متد whois_reverse_by_email

این متد دو آرگومان به نام `api_key` و `email` را دریافت می کند و خروجی که شامل کلیه اطلاعات `whois` ثبت شده با `email` ارسال شده را در خروجی برمی گرداند. این متد از `api` استفاده می کند بنابراین ارسال `api_key` معتبر اجباری می باشد. از آن جایی که در `api` مربوط به `whoxy`، دامنه های `ir` وجود ندارد، بنابراین با ارسال `email` هایی که دامنه های `ir` ثبت کرده اند به این متد، خروجی دریافت نخواهد شد.

ورودی	توضیحات
method_id	شناسه تابع، برای این تابع برابر ۱ است
Api_key	یک <code>api_key</code> معتبر سایت <code>whoxy</code>
email	<code>email</code> مورد جست و جو در <code>whois</code>

جدول مربوط به ورودی متد `Reverse_whois_by_email`

نمونه خروجی این متد در [لینک](#) زیر موجود است.

۸-۱ متد whois_history

این متد دو آرگومان به نام api_key و domain را دریافت می کند و در خروجی لیستی از تغییرات اعمال شده در اطلاعات whois این دامنه را بر می گرداند. این متد از api استفاده می کند بنابراین ارسال api_key معتبر اجباری می باشد. از آن جایی که در api مربوط به whoxy، دامنه های ir وجود ندارد، بنابراین با ارسال دامنه های ir. به این متد، خروجی دریافت نخواهد شد.

ورودی	توضیحات
method_id	شناسه تابع، برای این تابع برابر ۴ است
Api_key	یک api_key معتبر سایت whoxy
domain	domain مورد جست و جو در whois

جدول مربوط به ورودی متد whois_history

نمونه خروجی این متد در [لینک](#) زیر موجود است.

۹-۱ متد whois_get_balance

این متد یک آرگومان به نام api_key را دریافت می کند و خروجی که شامل میزان اعتبار باقی مانده در api می باشد. این متد از api استفاده می کند بنابراین ارسال api_key معتبر اجباری می باشد.

ورودی	توضیحات
method_id	شناسه تابع، برای این تابع برابر ۰ است
Api_key	یک api_key معتبر سایت whoxy

جدول مربوط به ورودی متد whois_get_balance

نمونه خروجی این متد در [لینک](#) زیر موجود است.

۱۰-۱ متد reverse_ip_lookup

این متد دو پارامتر به نام api_key و host و domain_count_flag و max_results را دریافت می کند و خروجی که شامل کلیه دامنه های ثبت شده برای host مورد نظر می باشد را در خروجی بر می گرداند. این متد از api استفاده می کند بنابراین ارسال api_key معتبر اجباری می باشد.

ورودی	توضیحات
method_id	شناسه تابع، برای این تابع برابر ۵ است

api_key	یک api_key معتبر سایت dnsinfo
host	ip مورد نظر برای جستجو می‌باشد.
domain_count_flag	مقدار آن برابر با True یا False می‌باشد. در صورت True بودن فقط تعداد دامنه‌های ثبت شده را برمی‌گرداند.
max_results	حداکثر تعداد نتایجی که مورد نظر می‌باشد.

جدول مربوط به ورودی متد reverse_ip_lookup

نمونه خروجی این متد در [لینک](#) زیر موجود است.

۱-۱۱ محتوای موجود در socket request

برای استفاده از ماژول parser باید یک درخواست از طریق socket به فریم وورک 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 می‌گیرد. پارامتر data هم همان داده ارسالی است که شامل پارامترهای ورودی ماژول whois است.

```
{
  "data":{"domain": "time.ir"}
  "type": 0
}
```

نمونه body یک socket request برای دامنه ir.

```
{
  "data":{"domain": "time.com", "api_key": "key " }
  "type": 0
}
```

نمونه body یک socket request برای دامنه com.

در تصاویر نمونه بالا داریم:

data : داده ارسالی برای استفاده در ماژول که شامل پارامترهای ورودی ماژول هست.

type : نوع ورودی ماژول را مشخص می‌کند که می‌تواند از نوع unstructured,path,file و یا انواع دیگر باشد و باید id ورودی موردنظر در این قسمت قرار گیرد.

Api_key: این پارامتر api_key لازم جهت استفاده از سرویس api مربوط به whois است. مقداردهی این پارامتر در صورتی که درخواست whois برای دامنه غیر از ir. باشد. الزامی است و در صورت عدم مقداردهی ماژول به شما خطا بر می گرداند. (توجه شود در صورتی که برای دامنه ir. درخواستی داده می شود. نیازی به ارسال api_key, نمی باشد.

۱۲-۱ جدول خطاها

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

توضیحات	نام	کد
زمانی که فرمت داده ورودی صحیح نباشد.	InvalidInputError	۱۰۲
زمانی رخ می دهد که در ماژول عادی whois نتواند برای دامنه داده شده یک whois مناسب پیدا کند.	InvalidDataError	۱۰۵
زمانی که مقدار api_key ارسالی نامعتبر باشد.	WrongApiKey	۱۰۴
زمانی که نتیجه ای برای درخواست داده شده یافت نشود.	ResultNotFoundError	۱۱۴
مشکلی در دسترسی به شبکه وجود دارد.	NetworkError	۱۰۷
مشکل غیر منتظره ای درون ماژول رخ داده است.	InternalModuleError	۱۰۱

۱۳-۱ جدول method_id

id	method
۰	Get_balance()
۱	Reverse_whois_by_email()
۲	Reverse_whois_by_name()
۳	Whois_doma_info()
۴	Whois_history()
۵	reverse_ip_lookup()