

domain-tool REST API 接口说明

<http://www.boluogan.com>

版本	修改人	发布日期	修改内容
0.1	老林	2015/1/8/	初始版本

目录

简介	2
默认根路径	2
REST API 说明	3
全部根后缀.....	3
可以解析结果的根后缀	3
单个域名查询	4
域名批量查询接口	5
websocket 接口调用说明	5
代表 websocket 客户端的 JS 全局变量.....	5
接口的连接和订阅	5
发送批量查询请求	6
断开接口连接	6
websocket 客户端程序设计（建议）	6
批量查询的返回数据说明.....	7

保存查询结果.....	7
查看历史查询结果.....	8

1.简介

接口程序基于 spring-boot 开发，用 Maven 构建，提供 restful 风格的 api 接口。
项目代码在 <https://github.com/boluogan/domain/tree/master/domain-tool>

2.默认根路径

<http://localhost:18080/>

元素	说明
接口地址	tlds/root/all
调用类型	GET
参数	无参数

返回	Json 数据，格式如下： ["se","as","edu","va","dm","re","in","fo","br","pl","ie", "vg","jobs","gl","th","com","cn","coop","ua","lk","sk", "cd","org","ru","sh","cx","at","bg","pt","me","na","cc", "it","si","tf","ee","jp","my","tm","ch","bz","il","is", "ly","am","nl","tc","tl","tk","eu","info","li","ag","be", "au","pro","kr","no","ws","name","hu","int","cat","tv", "mx","la","sg","museum","lt","fr","ca","vc","ae","us", "ck","mc","af","nu","aero","sa","hm","iq","mobi","nf", "dk","lv","mil","cl","ms","sc","ac","gd","tw","biz", "uk","tr","lu","ro","st","mu","net","de","sm","fi", "nz","hk","tp","to","ki","gs","gov","cz","asia"]
----	---

3.REST API 说明

3.1 全部根后缀

本接口列出全部可以查询的域名根后缀

3.2 可以解析结果的根后缀

元素	说明
接口地址	tlds/root/parseable
调用类型	GET
参数	无参数

返回	Json 数据，格式如下： <pre>["uk", "com", "hk", "us", "net", "in", "biz", "info", "cn", "tw", "tv", "name", "me", "edu", "cc", "cd"]</pre>
----	--

这些后缀的域名，其 whois 查询结果字符串可以转成 DomainWhoisInfo 和 WhoisRecord 对象。

3.3 单个域名查询

查询单个域名的 whois 信息。

元素	说明
接口地址	whois/\${domainname}
调用类型	GET
参数	域名（不带 www，如:boluogan.com）
返回	Json 数据，格式如下： <pre>{ "status": "SUCCESS", "message": { "sid": 1, "domainname": "abc.hk", "registerStatus": "已注册", "createDate": "2010-09-03 00:00:00", "expirationDate": "2015-09-03 00:00:00", "registrant": null, "registrantEmail": "domain@mail.im", "registrar": "Hong Kong Domain Name Registration Company Limited" } }</pre>

3.4 域名批量查询接口

这是一个 websocket 接口，需要连接 websocket，订阅消息，然后在需要查询的时候发送消息到 websocket 接口，等待接口把查询信息推送回来。

具体调用可以参考示例页面：

<http://localhost:18080/scan.html>

3.4.1 websocket 接口调用说明

在示例程序中，用到了 sockjs-0.3.4.js 和 storm.js 开源框架。

3.4.2 代表 websocket 客户端的 JS 全局变量

```
var stompClient = null;
```

3.4.3 接口的连接和订阅

```
function connect() {

    var socket = new SockJS('/domainscan');
    stompClient = Stomp.over(socket);
    stompClient.connect({}, function(frame) {
        setConnected(true);

        console.log('Connected: ' + frame);
        stompClient.subscribe('/topic/bulkquery',
function(responseData){

                //var                responseDataBody                =
JSON.parse(responseData.body);
```

```
        //showMessage(responseData.body);

    });

});

}
```

3.4.4 发送批量查询请求

```
stompClient.send("/app/domainscan", {},
JSON.stringify([{"sid":1,"domainname":"boluogan.com"}, {"sid":2,"domainname":
"pingguogan.com"}]));
```

3.4.5 断开接口连接

```
function disconnect() {
    stompClient.disconnect();
    //setConnected(false);
    console.log("Disconnected");
}
```

3.4.6 websocket 客户端程序设计（建议）

建议查询页面在加载完毕之后，就连接 websocket。

批量查询事件的出发按钮可以在非连接状态之下设置为不可连接。

批量查询的返回数据说明

发送批量请求之后，推送返回 json 数据：

```
{"status":"BEGIN","message":"开始查询..."}
```

每查询完一个域名，讲查询到的 whois 信息推送回来：

```
{"status":"QUERYING","message":{"sid":1,"domainname":"boluogan.com",
"registerStatus":"已注册","createDate":"2014-06-11
00:00:00","expirationDate":"2015-06-11
00:00:00","registrant":null,"registrantEmail":null,"registrar":"ENAME
TECHNOLOGY CO., LTD."}}
```

查询结束后，返回(status 为 Done，表示查询结束。)：

```
{"status":"DONE","message":"批量查询结束."}
```

3.5 保存查询结果

为此次域名查询指定一个名称，并将查询结果保存。

元素	说明
接口地址	savequeries
调用类型	POST

参数	<p>Json 数据，数据格式参考：</p> <pre>{ "name": "以 daikuan 为开头的三拼 COM 查询", "data": [{ "sid": 1, "domainname": "abc.hk", "registerStatus": "已注册", "createDate": "2010-09-03 00:00:00", "expirationDate": "2015-09-03 00:00:00", "registrant": null, "registrantEmail": "domain@mail.im", "registrar": "Hong Kong Domain Name Registration Company Limited" }] }</pre>
返回	<p>Json 数据，格式如下：</p> <pre>{ "status": "SUCCESS", "message": "处理成功" }</pre>

3.6 查看历史查询记录

列出已保存的历史查询记录（3.5 一节所说接口所保存的全部数据）。

元素	说明
----	----

接口地址	queries
调用类型	GET
参数	无参数
返回	<p>Json 数据，格式如下：</p> <pre>[{"name": "以 daikuan 为开头的三拼 COM 查询", "url": "/datas/201412/20141217203508.json"}]</pre>