# K-Darwin网站 Rest API文档

## GET /list

获取可用的文件列表

输入参数

output: 输出格式，可选值json和html

type：输出文件类型，可选值automate和model，分别代表动态过程文件和静态知识图文件，分别对应文件扩展名.txt和.json

输出

如果output=html，返回的是一个html页面，里面是一个html ul。

如果output=json，返回json，格式如下：

[

[

"\u795e\u7ecf\u7f51\u7edc",

"%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C"

],

[

"\u6211\u4eec\u4e1c\u65b9?3\*\*\*f",

"%E6%88%91%E4%BB%AC%E4%B8%9C%E6%96%B9%3F3%2A%2A%2Af"

]

]

其中，元素的第一个值为utf-8编码的文件名，第二个值为html编码的文件名。

## GET /services

获取可用的搜索服务列表

输入参数：无

输出：json字符串，格式如下：

{

"services": [

{

"desc": "\u4f7f\u7528\u767e\u5ea6\u767e ",

"id": "baike",

"img": "/img/baidu.png",

"name": "\u767e\u79d1",

"select": true

},

{

"desc": "\u641c\u7d22smart ref\u7684\u96f6\u4ef6\u5e93",

"id": "smartref",

"img": "/img/coins.png",

"name": "Smart Ref",

"select": true

}

]

}

desc为描述，id为服务的id，img为服务的icon的url，name为服务的显示名称，select为是否默认启用该服务。

## GET /model/load/{id}

加载静态知识图文件

输入：

id为文件名（经过html编码，不包括.json部分）

输出：json，格式如下：

{

"nodes": [

{

"index": 0,

"fixed": true,

"name": "\u66fc\u5fb7\u62c9",

"url": "",

"distance\_rank": 0,

"type": "query",

"id": "query\_\u66fc\u5fb7\u62c9"

},

{

"index": 1,

"name": "\u66fc\u5fb7 ",

"url": "http://www.chinanews.com/gj/2013/12-15/5620733.shtml?f=baidu",

"distance\_rank": 0,

"type": "referData",

"id": "referData\_\u66fc\u5fb7\u62c9 "

}

],

"links": [

{

"source": "query\_\u66fc\u5fb7\u62c9",

"target": "referData\_\u66fc\u5fb7\u62c9 "

},

{

"source": "query\_\u66fc\u5fb7\u62c9",

"target": "referData\_\u5feb\u8baf:\u5357\u975e\u603b "

},

]

}

nodes部分是节点的列表，每个节点的属性如下：

index: 数字，从0开始的数组下标

fixed： 布尔，是否在知识图上固定

name： 字符串，显示名称

url： 字符串，部分知识点会有链接

distance\_rank：数字，表示到其他节点的距离（聚类中使用）

type： 字符串，节点类型，用于节点的染色。常见type有referData、doc和relationship。

id： 字符串，节点的唯一标识。id是连线部分要使用的。

links部分是连线的列表，每个连线的属性如下：

source： 数字或字符串，对应连线起始节点的index或id。

target： 数字或字符串，对应连线终止节点的index或id。

## POST /model?id={save\_file\_name}

保存静态知识图

输入：

id：要保存的文件名（html编码，不包括.json部分）

POST form部分，是一个json格式的字符串，格式同GET /model/load/{id}的输出。

输出：格式为json

如果保存成功，返回{"message":"ok"}

如果保存失败，返回{"error":"#{error\_message\_info}"}

调用示例

使用javascript和jquery，可以这样调用该服务：

$.post("/model?id=" + save\_file\_name, res, function(d) {

if (d.error != null) {

notify("保存出现如下错误:" + d.error);

return;

}

notify("已保存");

return;

});

## GET /play/{scr}

加载动态过程文件

输入

scr：要加载的名称，不包括.txt部分，html编码

输出：

json字符串，顶层是一个列表，列表元素为GET /model/{id}中的json结构。

调用示例

var scr = encodeURIComponent("name1");

$.getJSON("/play/" + scr, function(d) {

});

## POST /automate

提交信息垂钓的请求，根据当前的知识图，结合输入中的各种停止条件

输入：form data为json格式，顶层为一个字典，其中有如下的key:

nodes: 列表，当前知识图中的所有节点

links： 列表，当前知识图中的所有连接

blacklist: 列表，当前知识图中曾经被删除的节点的id（不再探索的节点）

out\_fname: 字符串，要保存的文件名

max\_total\_node\_num: 数字，最大节点数

max\_single\_node\_num： 数字，单节点最大连接数

timeout\_seconds： 数字，以秒为单位的超时

max\_depth： 数字，最大深度（从下标为0的节点开始算）

示例：

var dic = {

"nodes": nodes,

"links": links,

"blacklist": [],

};

var \_ref = "max\_total\_node\_num max\_single\_node\_num timeout\_seconds max\_depth out\_fname".split(" ");

for (\_i = 0, \_len = \_ref.length; \_i < \_len; \_i++) {

var p = \_ref[\_i];

dic[p] = window[p][1].value;

}

$.post("/automate", JSON.stringify(dic), function(d) {

if (d.error != null) {

notify(("宏 " + dic.out\_fname + " 运行出现如下错误 ") + d.error);

} else {

notify("宏 " + dic.out\_fname + " 已完成运行");

}

});

## POST /search

搜索文档请求

输入：form data为json格式字符串，结构如下：

{

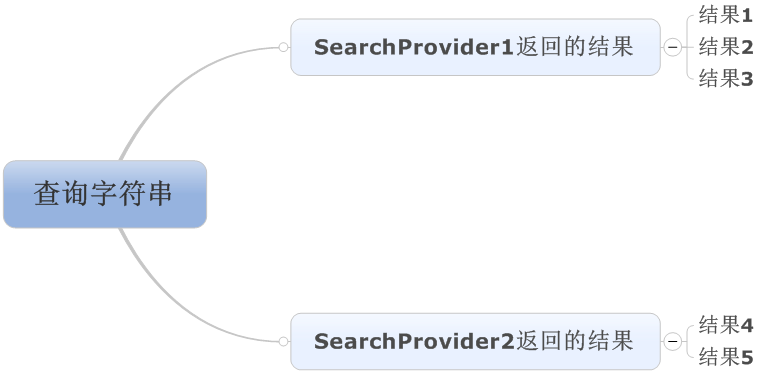
'keys': key,

'services': services

}

其中keys为关键字，services为字符串列表，每个字符串为搜索服务的id。具体有哪些搜索服务可用，参见GET /services。

输出：为json字符串，格式同GET /model/id，其中的节点和连线的拓扑结构如下：



## POST /explore

探索知识点的请求

输入：form data为json格式字符串，结构如下：

{

keys: “波音777”,

return\_id: “baike\_波音777”，

url: “http://www.a.com/b/c”

}

keys为要探索的节点的名称（也就是知识点的名称），return­\_id为节点的id(用于结果返回时找到正确的节点)，url为可选，优先级高于keys。

输出：为json字符串，格式如下：

{

“baike\_波音777”: [

node1,

node2,

node3

]

}

其中键和输入中的return\_id对应，列表元素（node1等等）的格式，参见GET /model/{id}。

示例

var data = {

keys: “波音777”,

return\_id: “baike\_波音777”

};

if (d.url != null) {

data.url = d.url;

}

$.post("/explore/", JSON.stringify(data), function(data){

//数据处理

}, 'json');