CII数据源接口说明

ds(datasource)

# Response

成功的格式：{"success":true,【请求返回数据】}

失败的格式：{"success":false,"error":"error msg"}

故下文仅列出成功格式。

# 数据源信息列表

Method:get

Url:http://ip:port/**cii/ds/getAllDS**

Response:{"success":true,”results”:[{obj},{obj},{obj}...]}

obj={“ds\_id”:””,“ds\_name”:””,”ds\_type”:””,”ds\_json”:{“k”:”v”}}

说明：ds\_json里的k,v为各源定义的配置内容。

建议：对于不同type的json格式不一致问题，页面可考虑分组grid展示；

# 数据源映射信息

Method:post

Url:http://ip:port/**cii/ds/getMpDS**

Data:{“ds\_type”:””}

Response:{"success":true,”results”:[{obj},{obj},{obj}...]}

obj: {"ds\_id": "","ds\_name": ""}

说明：页面的select数据源(展示ds\_name值,存储ds\_id值)

# 数据源名称重复检查

同一type的name不允许重复

####################################

Method:post

Url:http://ip:port/**cii/ds/filterName**

Data:{“ds\_type”:””,“ds\_name”:””}

Response:{"success":true,”exist”:true|false}

说明：exist:已经存在(true),不存在(false)

# 数据源连接测试

#######暂不可用########

Method:post

Url:http://ip:port/**cii/ds/connect**

Data:{“ds\_type”:””,“ds\_json”:{“k”:”v”}}

Response:{"success":true}

说明：ds\_json里的k,v为各源定义的配置内容。

# 创建修改数据源信息

Method:post

Url:http://ip:port/**cii/ds/storeSource**

Data:{“ds\_id”:””,“ds\_name”:””,”ds\_type”:””,”ds\_json”:{“k”:”v”}}

Response:{"success":true,”ds\_id”:””}

说明：创建数据源时ds\_id为空；

ds\_json里的k,v为各源定义的配置内容

# 数据源定义

## Kafka

{ “kafka\_url”:””, # kafka地址

”zk\_url”:””, # zookeeper地址

”jmx\_url”:”” # kafka jmx地址

}

# 附录

## Mysql表

CREATE TABLE `ciisource` (

`ds\_id` char(14) NOT NULL,

`ds\_name` varchar(30) NOT NULL COMMENT '数据源名称',

`ds\_type` tinyint(1) NOT NULL COMMENT '数据源类型，kafka(0)',

`ds\_json` varchar(500) NOT NULL COMMENT '数据源内容',

PRIMARY KEY (`ds\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;