

图数据库 KonisGraph

数据导入



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

数据导入

最近更新时间：2023-05-16 11:17:34

您可以将图的顶点和边数据导入图数据库 KonisGraph 进行存储、管理和 [图数据可视化查询](#) 等。本文为您介绍如何将图数据导入 KonisGraph 实例。

概述

KonisGraph-Importer 是图数据库 KonisGraph 的数据导入组件，能将多种数据源的数据转换为图的顶点（Vertex）和边（Edge），并批量导入到图数据库中。

目前支持的数据源包括：本地磁盘 CSV 格式文件。

数据文件准备

CSV 文件是带分隔符的文本文件。第一行通常是标题，记录了每一列的名称。其余的每行代表一条记录，会转换为一个顶点或一条边。行的每一列对应一个字段，会转化为顶点或边的 ID 及属性等。

当前 KonisGraph-Importer 通过在映射文件中指定标题，因此数据源文件如果是带有标题行的，需要将其去掉。

图库元数据准备

在开始导入点边数据之前，首先要对已有的数据和对应的 [图模型](#) 有一个清晰的构想，然后通过可视化页面或者 console 插入元数据，就完成 [图模型的构建](#)。用户需要关注的元数据包括：

- 属性：导入的点包含哪些属性，导入的边包含哪些属性。
- 点的元数据：点表示了一个实体对象，构建时需要了解点应该包含除 ID 外的哪些属性。
- 边的元数据：边表示了实例之间的关系，定义是需要了解关系的主体和客体，以及关系还包含哪些属性。

⚠ 注意

导入点的数据可能包含隐含的 ID 属性，添加属性时需注意。

数据导入任务

准备好数据源文件、输入源映射文件及构建好图模型后，就可以将数据导入到图数据库中，导入任务由用户通过工具在命令行发起。

KonisGraph-Importer 参数说明

| 参数 | 描述信息 |
|--------------|------------------------------|
| -addr <addr> | KonisGraph 图数据库内网地址:Import端口 |

| | |
|--|---|
| <code>-im <csv file, json file></code> | 上传并导入数据，如 <code>-im konws.csv knows.json</code> 。前者 <code>knows.csv</code> 表示数据文件，后者 <code>knows.json</code> 表示映射文件 |
| <code>-lo <json file></code> | 导入数据，如 <code>knows.json</code> |
| <code>-ls</code> | 显示已上传文件列表 |
| <code>-p <password></code> | KonisGraph 图数据库访问密码 |
| <code>-u <user></code> | KonisGraph 图数据库访问用户 |
| <code>-rm <csv file></code> | 删除已上传的文件 |
| <code>-up <csv file></code> | 上传文件 |

数据导入示例

以 [tinkerpop 示例](#) 来说明如何进行数据导入。

1. 获取 KonisGraph-Importer

下载最新版本的 KonisGraph-Importer 包：

```
wget https://documentation-down-1306723546.cos.ap-nanjing.myqcloud.com/importer-client-bin.tar.gz

tar zxvf importer-client-bin.tar.gz
```

解压后会得到下述目录结构：

```
importer-client
| - bin
| - conf
| - lib
| - logs
```

在 `conf` 目录下提供了一个用于测试的数据源文件及映射文件。

2. 创建图模型

图模型，即图中点/边/属性的模型，创建图模型有两种方式：图可视化的图库管理、gremlin console。

2.1 通过 gremlin console 创建

通过 gremlin console 操作图数据库创建图模型的属性元数据。

示例中包含 name 、 age 、 lang 、 weight 等4个显式属性以及 id 隐式属性，添加属性数据：

```
s.addP("id", "T_LONG", "0")
s.addP("name", "T_STRING", "")
s.addP("age", "T_LONG", "0")
s.addP("lang", "T_STRING", "")
s.addP("weight", "T_DOUBLE", "0.0")
```

示例中包含两类实例：person 和 software。其中 person 包含额外的 name 和 age 属性，software 包含额外的 name 和 lang 属性。添加点元数据：

```
s.addV("person", "id", ["name", "age"])
s.addV("software", "id", ["name", "lang"])
```

示例包含两类关系：knows 和 created。knows 表示 person 与 person 之间关系，其主体和客体都是 person；created 表示 person 和 software 的之间的关系，主体是 person，客体是 software。添加边元数据：

```
s.addE("knows", "person", "person", ["weight"])
s.addE("created", "person", "software", ["weight"])
```

2.2 通过图库管理创建

通过图可视化的图库管理，创建图模型，参考 [图库管理](#)。

3. 准备数据源文件

3.1 顶点数据

顶点数据由一行一行数据组成，下面以 CSV 格式进行说明。

- person 顶点数据，命名为 person.csv

```
1,marko,29
2,vadas,27
4,josh,32
6,peter,35
```

- software 顶点数据，命名为 software.csv

```
3,lop,java
```

```
5,ripple,java
```

3.2 边数据

边数据文件由一行一行数据组成，每一行表示一条边。其中部分列作为源顶点和目标顶点的 ID，其他列作为边属性。

- knows 边数据，命名为 knows.csv

```
1,2,0.5
1,4,1.0
```

- created 边数据，命名为 created.csv

```
1,3,0.4
4,3,0.4
6,3,0.2
4,5,1.0
```

3.3 编写映射文件

映射文件用于描述如何将输入源数据与图的顶点类型或边类型建立映射文件，以 json 格式组织。

- person 顶点映射文件，命名为 person.json

```
{
  "type": "VERTEX",
  "graph": "default",
  "label": "person",
  "data_source": {
    "source_type": "CSV",
    "csv_source": {
      "filename": "person.csv",
      "header": [
        "id",
        "name",
        "age"
      ],
      "delimiter": ","
    }
  }
}
```

- software 顶点映射文件，命名为 software.json

```
{
  "type": "VERTEX",
  "graph": "default",
  "label": "software",
  "data_source": {
    "source_type": "CSV",
    "csv_source": {
      "filename": "software.csv",
      "header": [
        "id",
        "name",
        "lang"
      ],
      "delimiter": ","
    }
  }
}
```

- knows 边映射文件，命名为 knows.json

```
{
  "type": "EDGE",
  "graph": "default",
  "label": "knows",
  "data_source": {
    "source_type": "CSV",
    "csv_source": {
      "filename": "knows.csv",
      "header": [
        "SID",
        "OID",
        "weight"
      ],
      "delimiter": ","
    }
  }
}
```

- created 边映射文件，命名为 created.json

```
{
  "type": "EDGE",
  "graph": "default",
  "label": "created",
  "data_source": {
    "source_type": "CSV",
    "csv_source": {
      "filename": "created.csv",
      "header": [
        "SID",
        "OID",
        "weight"
      ],
      "delimiter": ","
    }
  }
}
```

3.4 执行命令导入

```
sh bin/start.sh -u steven -p rainbow-dash -im person.csv person.json -
addr localhost:8180
sh bin/start.sh -u steven -p rainbow-dash -im software.csv software.json
-addr localhost:8180
sh bin/start.sh -u steven -p rainbow-dash -im knows.csv knows.json -addr
localhost:8180
sh bin/start.sh -u steven -p rainbow-dash -im created.csv created.json -
addr localhost:8180
```