



OPC Router 中 MongoDB

使用指南

北京麦斯时代信息技术有限公司

2019年9月23日

目录

1. 需求背景.....	2
2. MongoDB 简介	2
3. 配置步骤.....	2
3.1. OPC UA	2
3.2. MongoDB	4
3.3. Connections	5
3.4. 运行结果.....	8

1. 需求背景

- 使用 KEPServerEX 与 PLC 通讯，读取 PLC 数据，然后使用 OPC Router 中的 OPC UA Plug-in 与 KEPServerEX 连接
- 使用 OPC Router 中的 MongoDB Plug-in，将 KEPServerEX 中读取的 PLC 数据送到 MongoDB 数据库。



2. MongoDB 简介

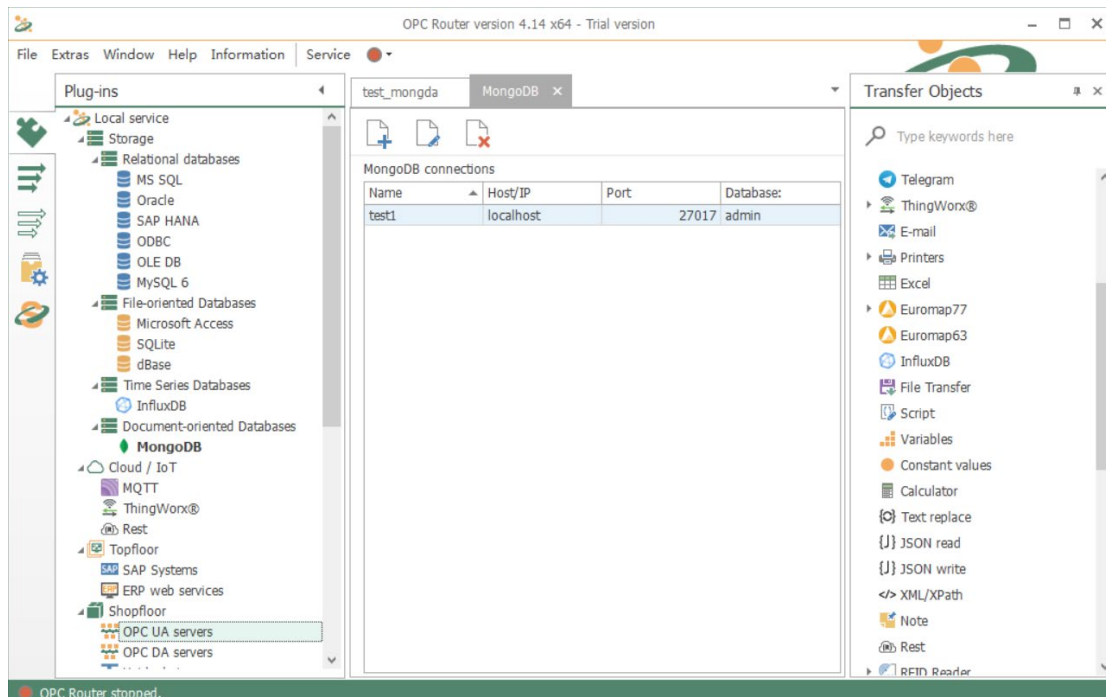
MongoDB 数据库是最常用的 NoSQL 数据库之一，属于面向文档的数据库，能够非常高效地处理大量 JSON 文档。MongoDB 的另一个重要功能是能够轻松复制到多台服务器，这使得数据库的可用性非常高。

作为一个面向文档的数据库，MongoDB 非常适合于分布式使用和处理大量 JSON 文档。在最新版本的 OPC Router v4.14 中新增加了 MongoDB 插件，可以很容易地将工业 4.0 的基础架构与 MongoDB 连接起来。

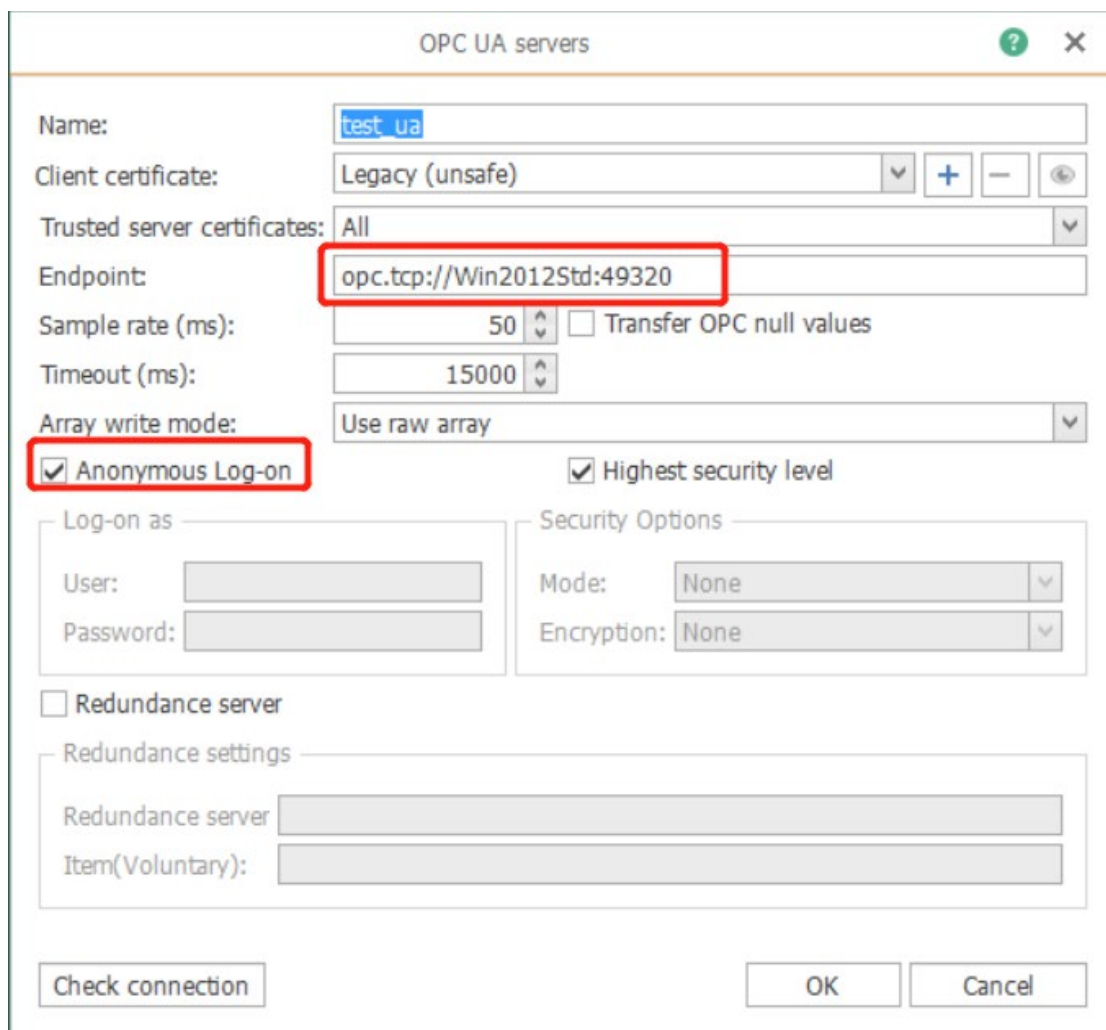
3. 配置步骤

3.1. OPC UA

点击 Windows 桌面 OPC Router 4 Configuration 图标，运行 OPC Router。

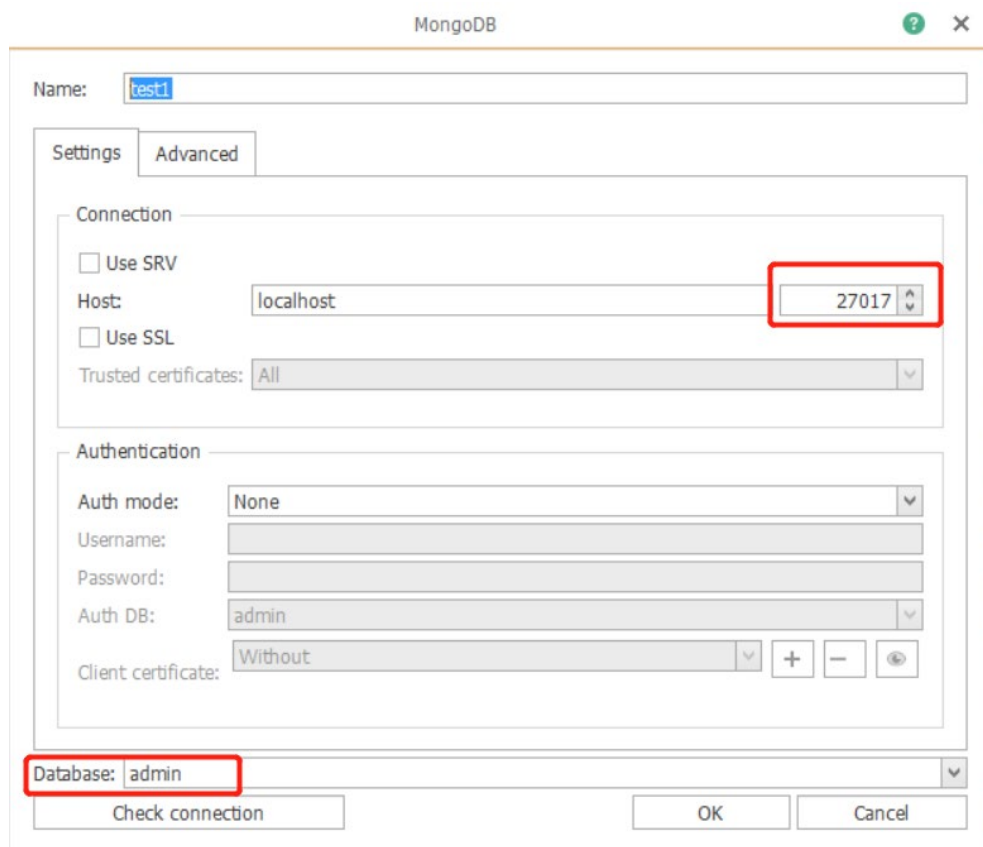
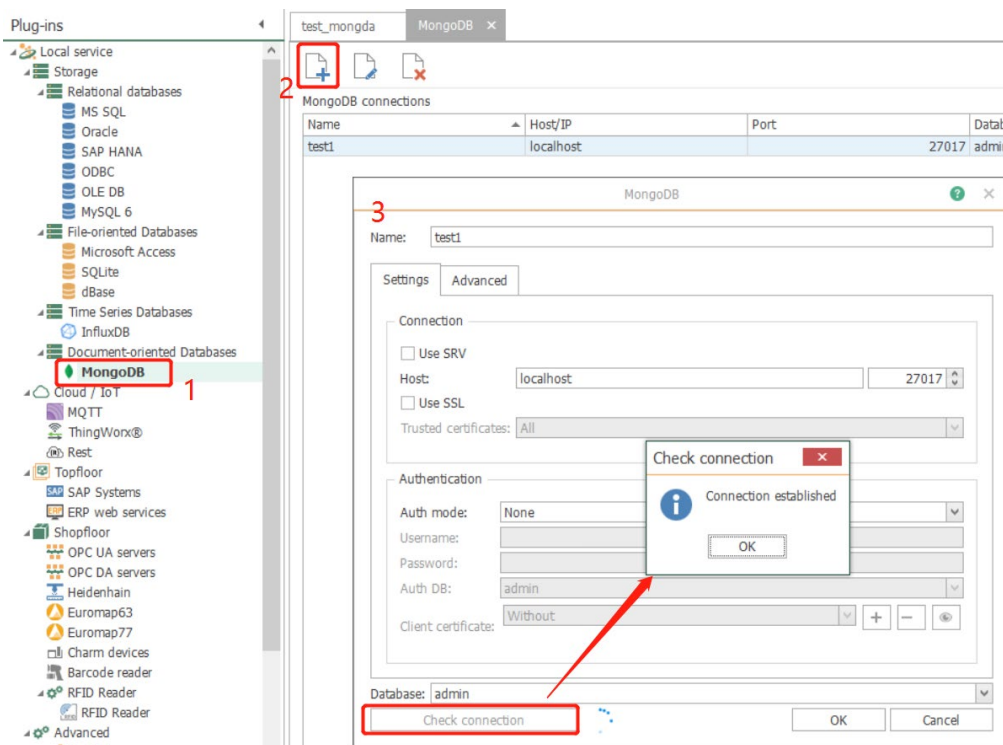


点击左侧 Plug-ins-> Shopfloor -> OPC UA Servers, 配置建立与 KEPServerEX 进行通讯。

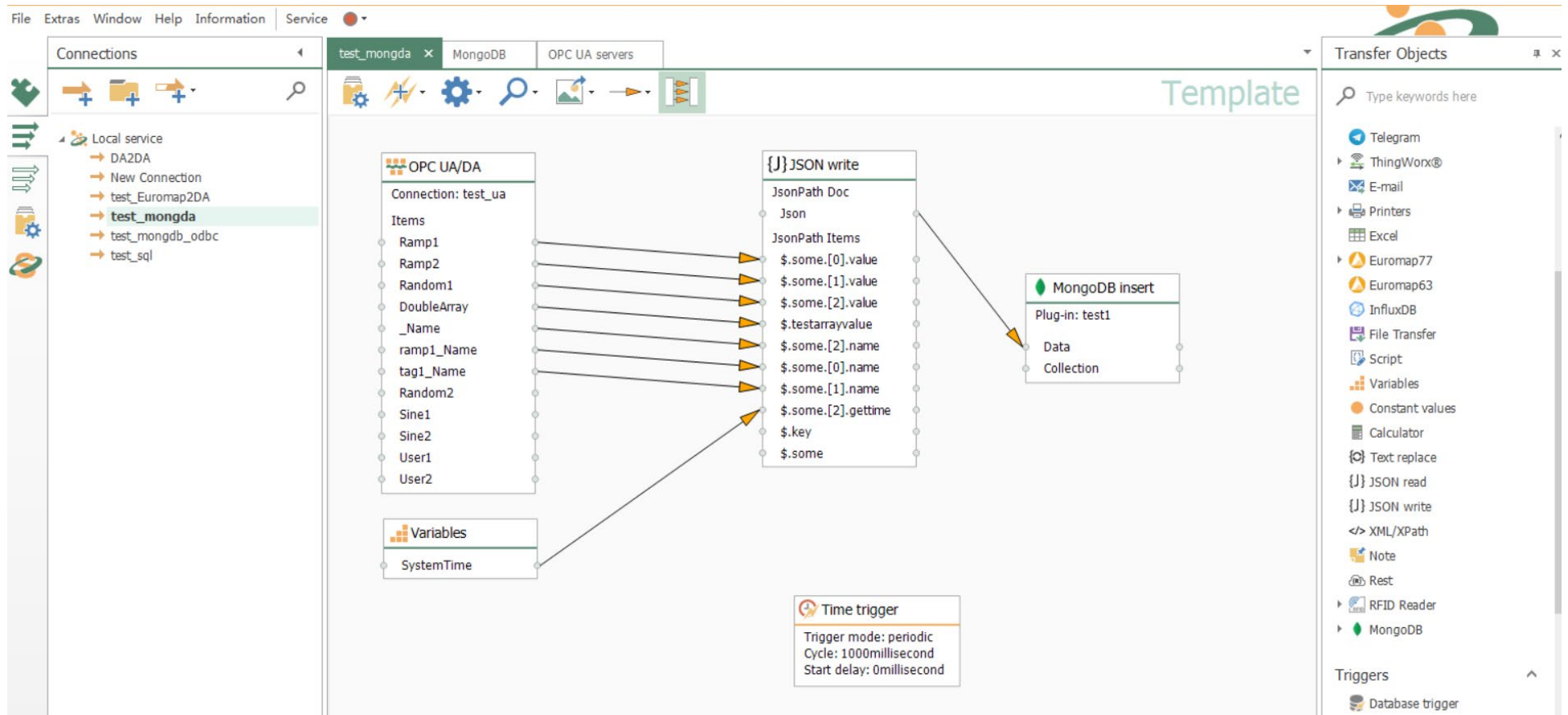


3.2. MongoDB

点击左侧 Plug-ins-> Document-oriented Databases -> MongoDB, 配置建立与 MongoDB 数据库通讯。

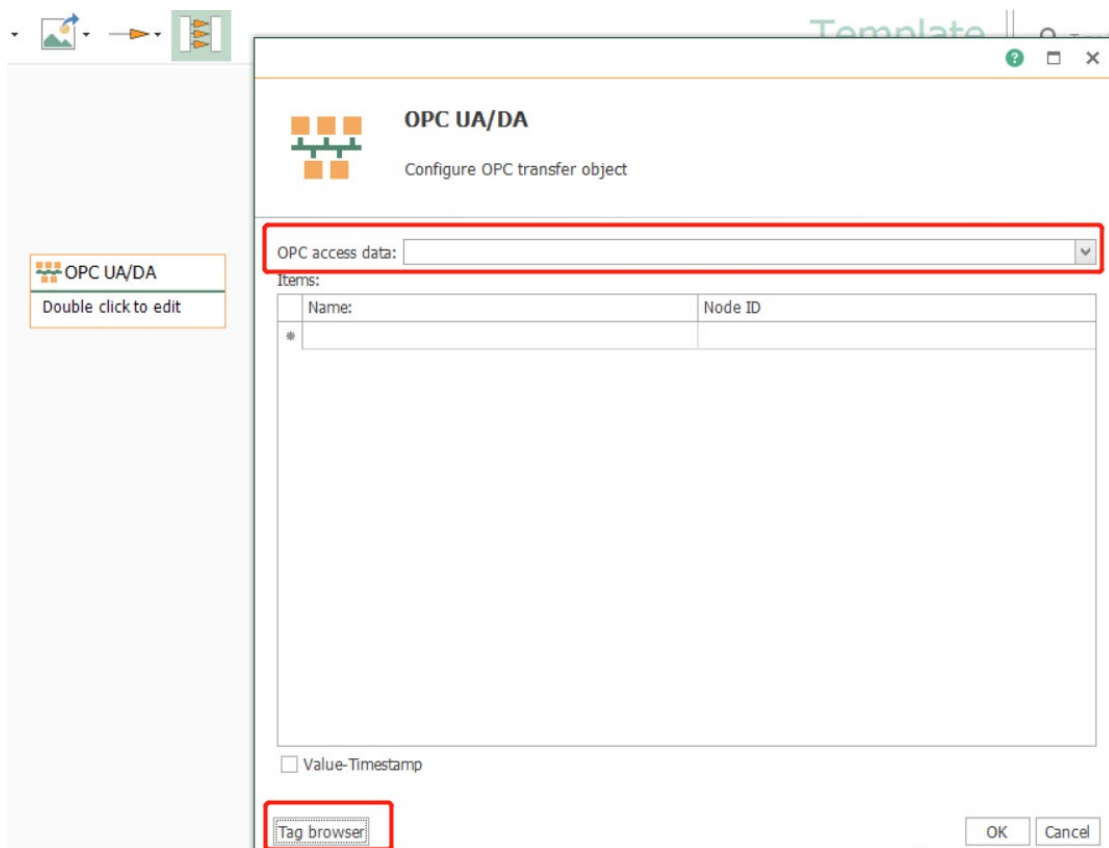


3.3. Connections



从右侧 Transfer Objects 里选择相关对象拖放到中间的界面里。

- 1) 首先, 点中 Data sources and destinations-> OPC UA/DA, 拖放到中间空白地方。



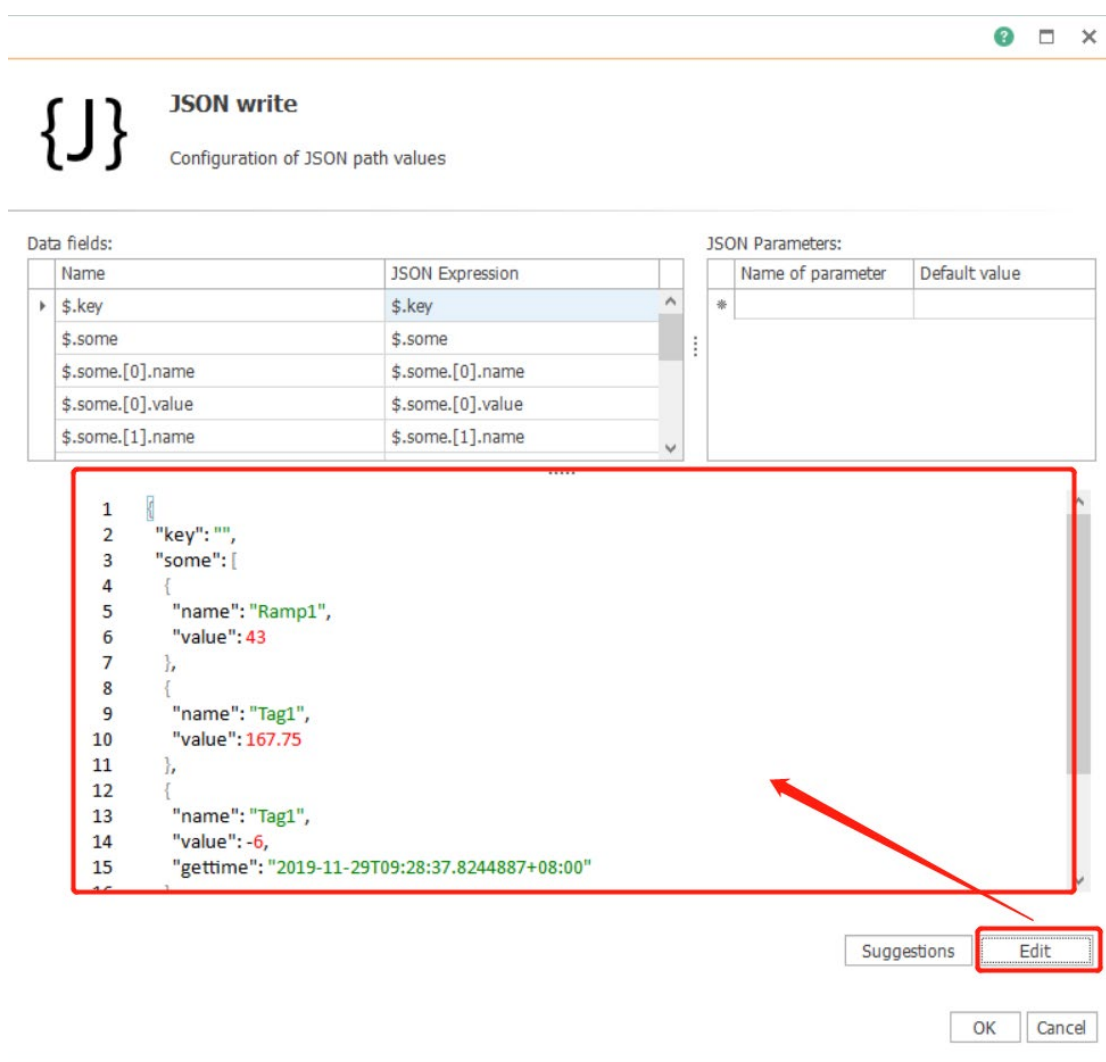
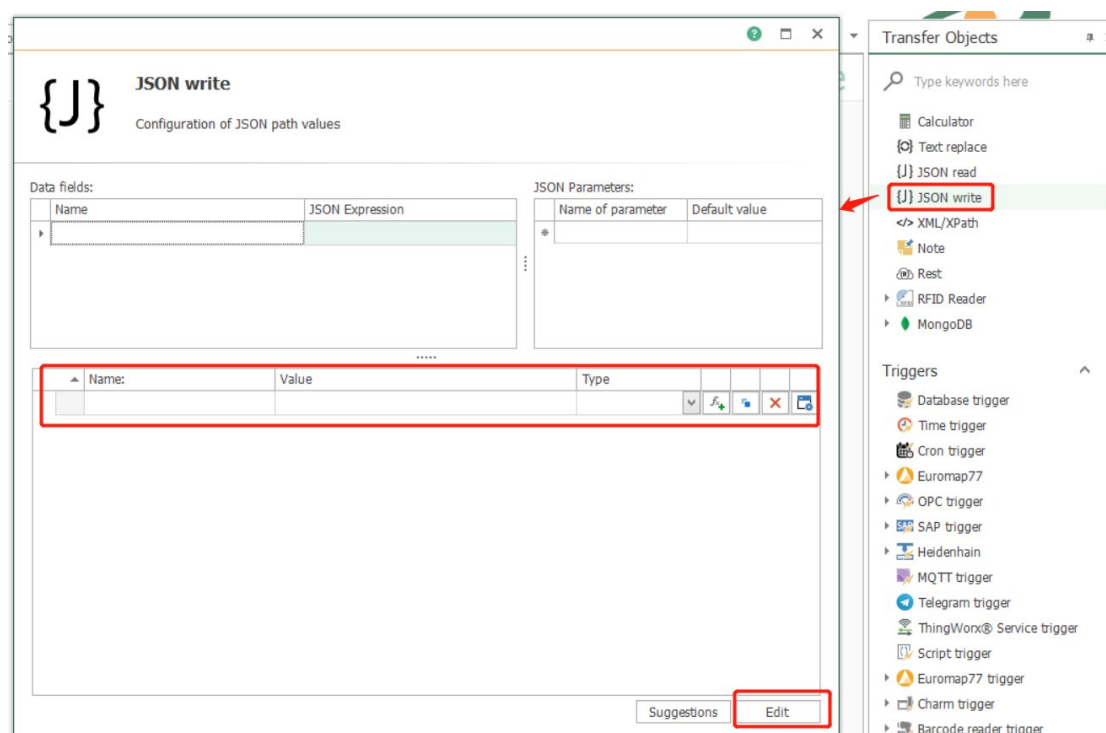
从 OPC access data 下拉列表里选择前面定义好的 OPC UA SERVER。

点击左下角 Tag browser, 从 OPC UA server 中选择标签。

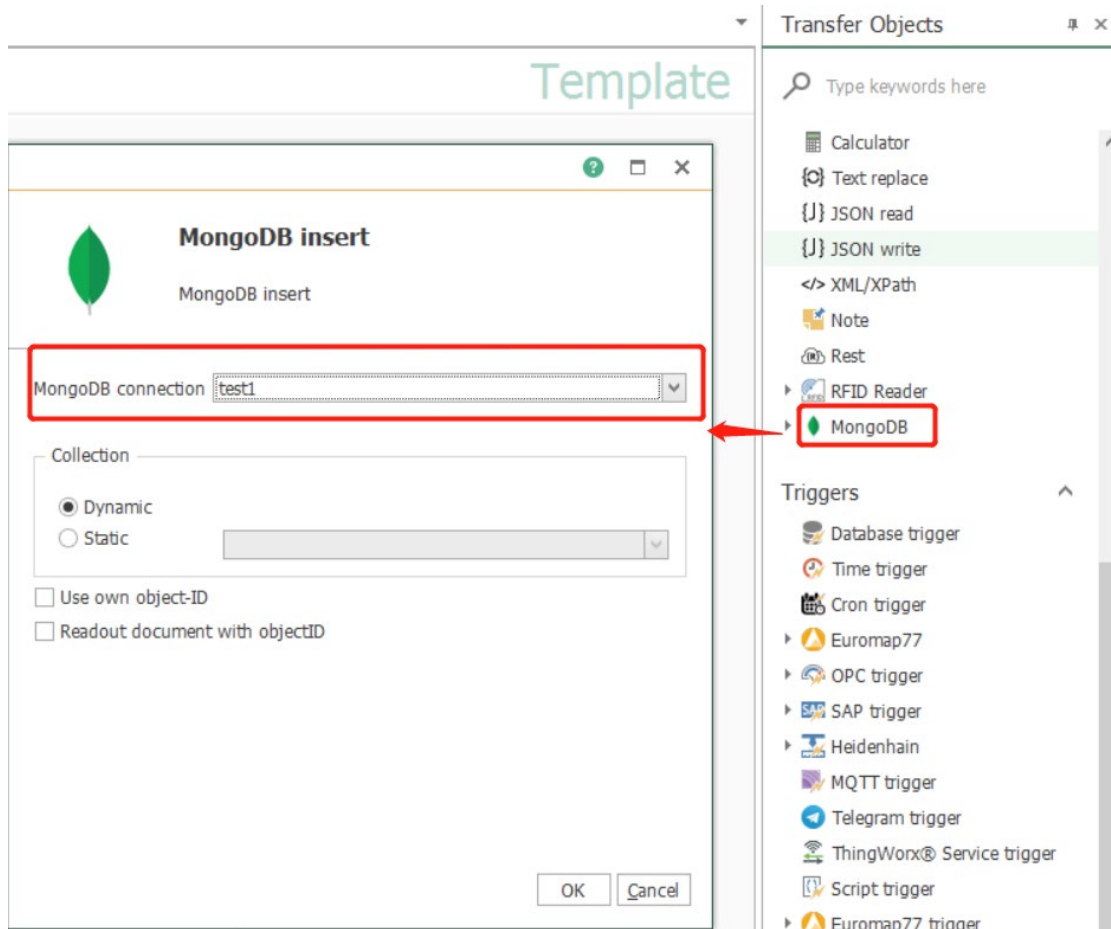
- 2) 点中 Data sources and destinations->JSON write

配置 JSON 中 name, value.

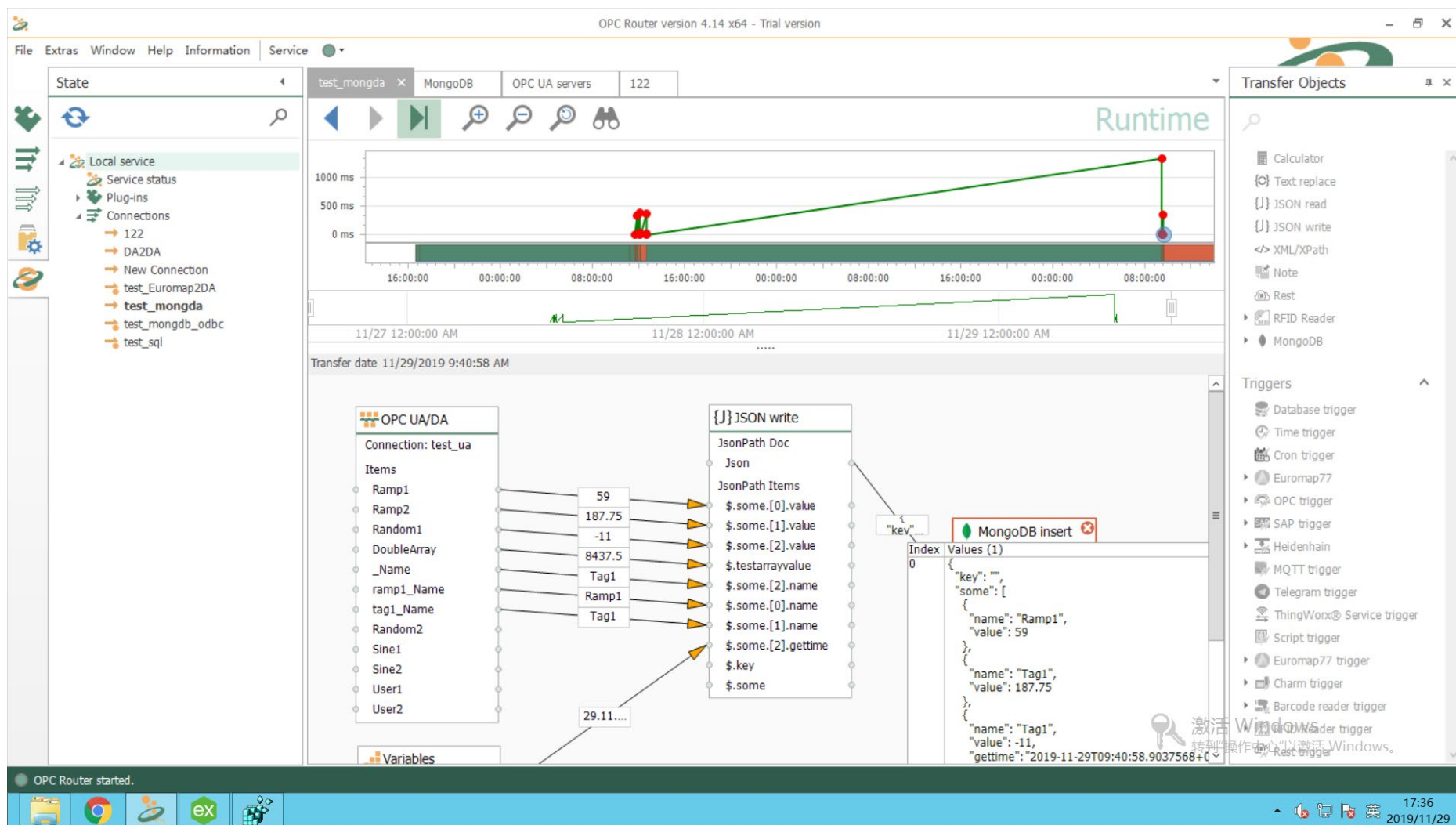
可以右小角 Edit, 直接在编辑器里写 json。



3) 点中 Data sources and destinations->MongoDB



3.4. 运行结果



OPC Router version 4.14 x64 - Trial version

File Extras Window Help Information Service

test_mongda MongoDB OPC UA servers 122

Runtime

State

- Local service
 - Service status
 - Plug-ins
 - Connections
 - 122
 - DA2DA
 - New Connection
 - test_Euromap2DA
 - test_mongda
 - test_mongodb_odbc
 - test_sql

Transfer date 11/29/2019 9:40:58 AM

OPC UA/DA

Connection: test_ua

Items

- Ramp1
- Ramp2
- Random1
- DoubleArray
- _Name
- ramp1_Name
- tag1_Name
- Random2
- Sine1
- Sine2
- User1
- User2

Variables

{[]} JSON write

JsonPath Doc

Json

JsonPath Items

- \$.some[0].value
- \$.some[1].value
- \$.some[2].value
- \$.testarrayvalue
- \$.some[2].name
- \$.some[0].name
- \$.some[1].name
- \$.some[2].gettime
- \$.key
- \$.some

MongoDB insert

Index Values (1)

```

{
  "key": "",
  "some": [
    {
      "name": "Ramp1",
      "value": 59
    },
    {
      "name": "Tag1",
      "value": 187.75
    },
    {
      "name": "Tag1",
      "value": -11,
      "gettime": "2019-11-29T09:40:58.9037568+08:00"
    }
  ]
}
  
```

29.11....

11/27 12:00:00 AM 11/28 12:00:00 AM 11/29 12:00:00 AM

1000 ms 500 ms 0 ms

16:00:00 00:00:00 08:00:00 16:00:00 00:00:00 08:00:00 16:00:00 00:00:00 08:00:00

Transfer Objects

- Calculator
- Text replace
- JSON read
- JSON write
- XML/XPath
- Note
- Rest
- RFID Reader
- MongoDB

Triggers

- Database trigger
- Time trigger
- Cron trigger
- Euromap77
- OPC trigger
- SAP trigger
- Heidenhain
- MQTT trigger
- Telegram trigger
- ThingWorx® Service trigger
- Script trigger
- Euromap77 trigger
- Charm trigger
- Barcode reader trigger
- Barcode reader trigger
- Rest trigger

OPC Router started.

17:36 2019/11/29

附：Router 软件架构

