

# 数据挖掘-服务

在进行机器学习实验的过程中，为了简化和加速模型的构建、训练和部署，使用自动化机器学习功能更快地识别合适的算法并优化超参数。

- 服务 workflow 示例
- 部署服务
  - 概述
  - 部署方法
    - 数据输入
    - 部署方式
    - 应用服务
  - 示例
- 服务监控
- 服务调用示例

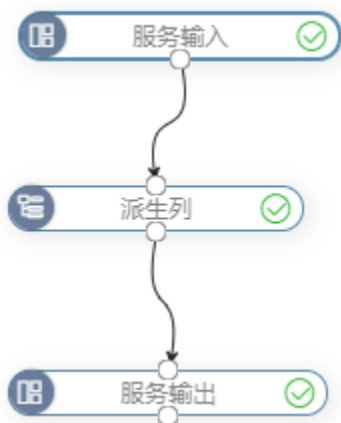
## 服务 workflow 示例

服务 workflow 是将数据挖掘以服务的方式进行发布。

要求：输入层必须是“服务输入”节点，输出层必须是“服务输出”节点。

通过部署服务后，通常用于数据预测的应用。“服务输入”的处理层可以实现数据来源于其它接口。

下图是一个简单的服务 workflow 示例：



在这个 workflow 中，实现的是对输入的数据进行派生列处理后输出。

## 部署服务

### 概述

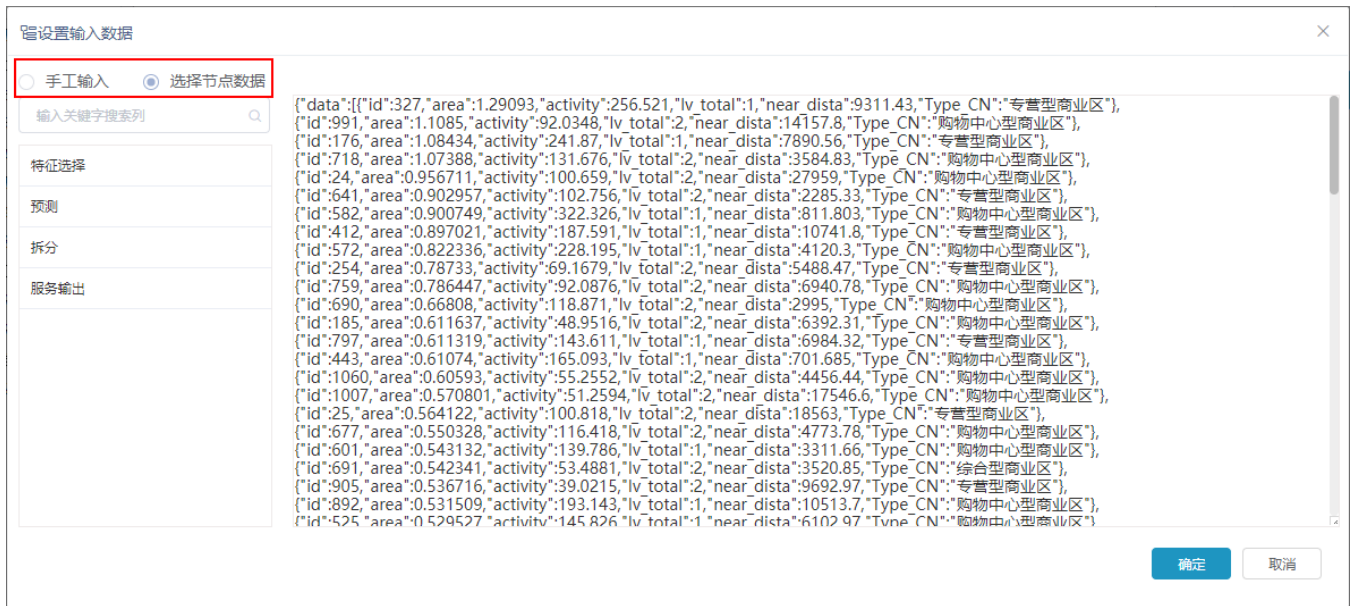
服务部署指的是当用户用实验训练好了一个预测模型，并将此预测模型部署成一个 web 的服务。当前发布的 web 服务就可以作为预测模型应用于类似场景的预测，当用户要预测一个新样本，只需要把样本特征信息用 restful api 传给服务，服务就可以依据训练好的预测模型返回预测结果。

前提条件：服务 workflow 完成搭建且执行成功。workflow 中每个节点旁边显示绿色的勾即表示该节点是执行成功的。

### 部署方法

#### 数据输入

服务输入中的数据支持“手工输入”和“选择节点数据”两种方式：



- 手工输入：是手工输入 json 列数据。
- 选择节点数据：是选择运行通过后的某节点输出结果数据。

## 部署方式

部署服务包括以下三种方式：

名称	说明
新增服务	输入服务名称新增一个服务： <div data-bbox="219 1060 1068 1514" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> </div> <p>一个服务实验可以生成多个服务。</p>

更新服务 选择一个原服务更新其实验节点：



灰度部署 灰度部署用于使多版本模型并行运行，并保持多版本模型运行的结果，通过结果比较灰度测试的模型版本的准确性和稳定性。



- 灰度测试：又名A/B测试、灰度发布，一种在黑白之间发布平滑过渡的方式。可以对其执行A/B测试，一些用户继续使用产品功能A，一些用户开始使用产品功能B，如果用户不反对B，则逐渐扩大范围并迁移所有用户到B来。  
灰度测试可以确保整个系统的稳定性，并且可以在初始灰度级找到并调整问题以确保其影响度。
- 灰度部署是基于某个服务(主服务/A服务)，创建一个新的服务（B服务），此服务与主服务按照“加权采样”进行分流处理。  
例如：A服务流量设置60，则B服务流量为40，将会有40%的概率调用B服务，将会有60%的概率执行A服务（主服务）。生成的B服务名称是以A服务名称加此时的时间戳，一个服务只能产生一个B服务。

在灰度部署中，已经生成过灰度服务的的服务将不能被选择。



- 关联服务：基于当前服务再创建一个关联的服务；
- 新增服务流量/关联服务流量：用于分流处理中分配服务的流量，流量之和为100%，即设置新增服务流量值为50，将会有50%的流量分配给新增服务，50%的流量分配给关联服务；

生成的服务将会在服务管理中显示，如图灰度部署生成两个服务，上面的为A服务，下面的为B服务。

便捷简单的对服务进行监控，随时了解服务运营状态

输入你想要搜索的内容

刷新

全部 4 应用 4 异常 0 已下线 0

名称	描述	修改时间	状态	流量控制	常用操作
部署服务		2021-08-30 10:41:32	应用	30% <i>1</i>	🔍 📄 🗑️ 🔄
部署服务-16302914899...		2021-08-30 10:41:32	应用	70% <i>1</i>	
服务-部署		2021-08-27 16:35:00	应用		
fw1		2021-08-27 16:30:39	应用		

其中“流量控制”列显示的是对应服务的流量，点击对应服务的流量值，可进行对应A/B服务的流量配置。

### 流量控制[部署服务] ✕

当前服务名称 部署服务

当前服务流量  ⬆️ ⬆️ Ⓞ️

关联服务名称 部署服务-1630291489985

关联服务流量  ⬆️ ⬆️ Ⓞ️

确定(O)
取消(C)

## 应用服务

目前服务工作流的应用通常为：对于已创建好的分类模型、回归模型、聚类模型，我们将新的数据输入后，通过模型计算预测出结果。

选择部署服务方式或成功部署服务后，可进行服务的应用。

应用入口：在“服务监控”界面的服务列表中，双击“名称”列中的服务，进入到“服务配置”界面，再单击 **服务测试** 页签，显示“服务测试”界面：

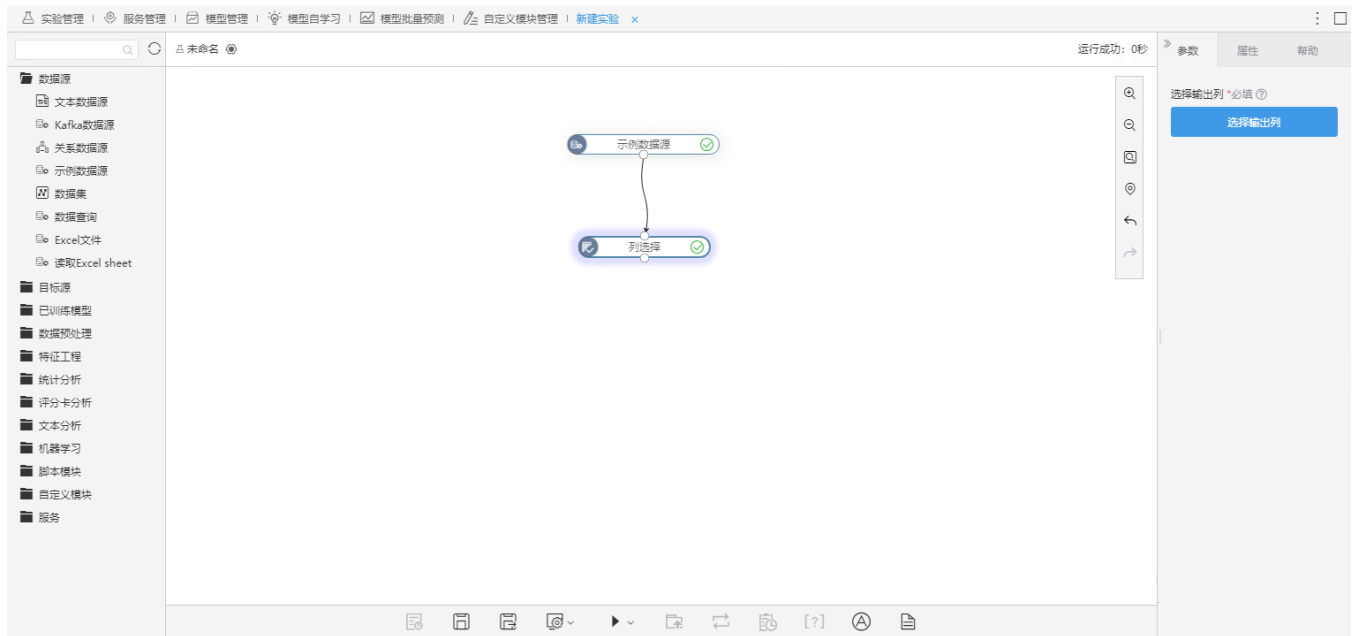


在“测试数据”文本框中按照示例填写数据，单击 **测试** 按钮，这些数据进入到当前服务部署的模型中执行后显示测试结果。

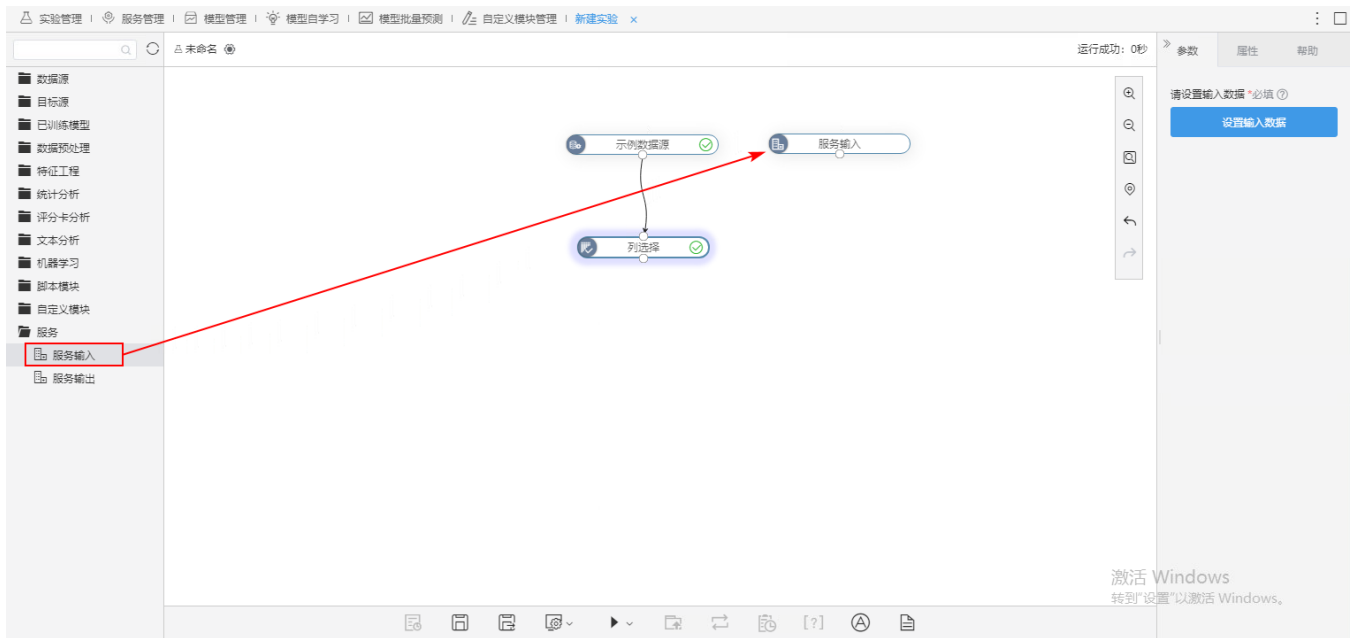
**i** 对于灰度部署，测试数据将按照流量分配选择A服务、B服务中的一列进行运行，返回最后的测试结果，测试结果即算法选择服务输出的结果。

## 示例

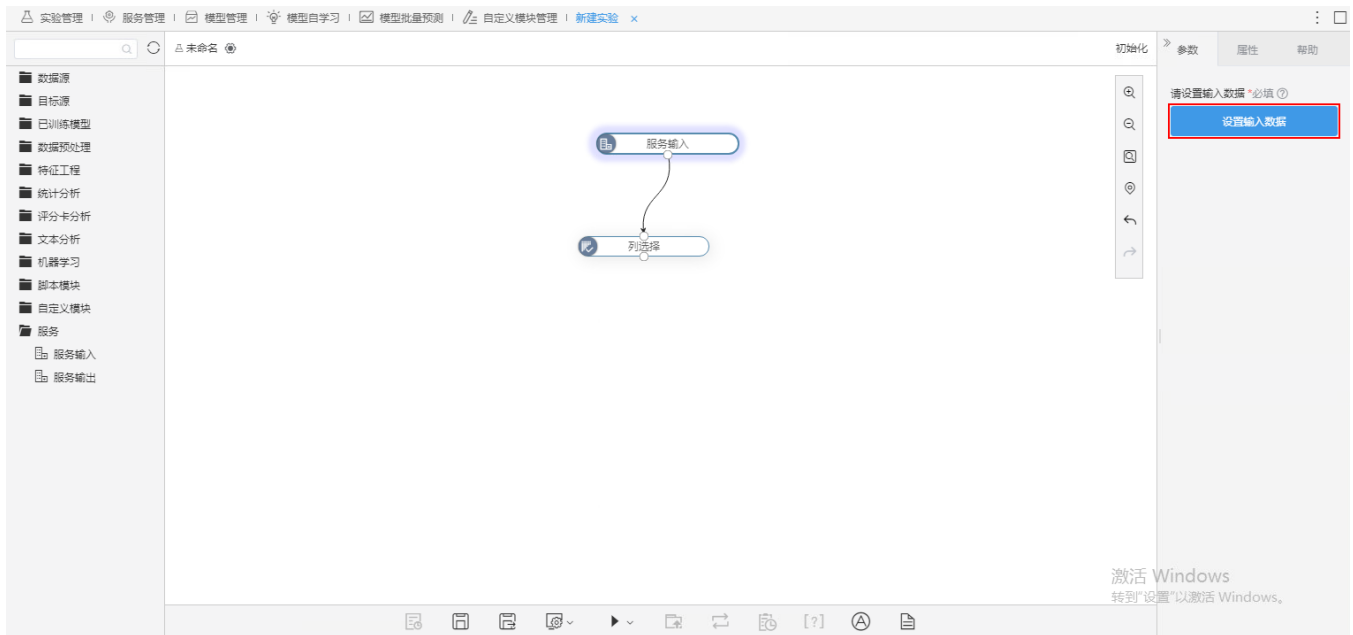
1、通过数据源或数据预处理定义实验模型，并运行该实验模型。



2、拖拽“服务输入”节点到画布，并选择节点，该节点的数据结果是用于服务发布的节点。



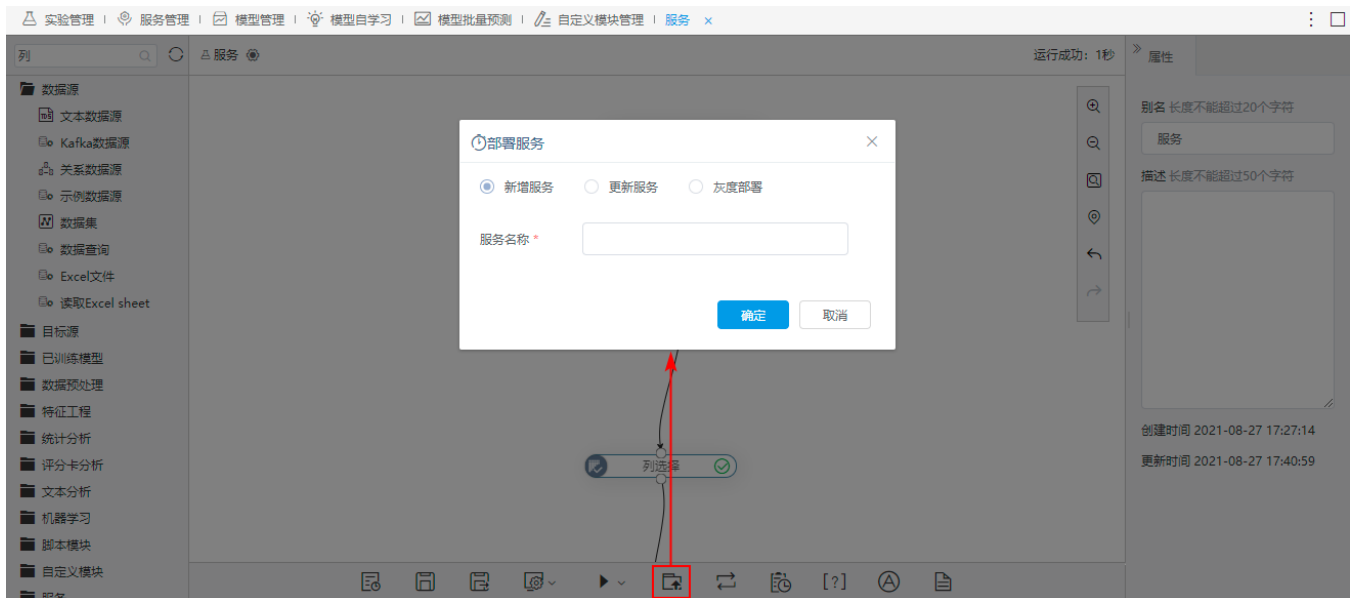
3、将“服务输入”节点在实验模型中替换所选的节点，并将所选节点前的所有节点删除，点击 **设置输入数据** 按钮。



输入数据默认为“选择节点数据”。



4、拖拽“服务输出”节点到流程节点最末，流程执行通过后单击工具栏的 **部署服务** 按钮。



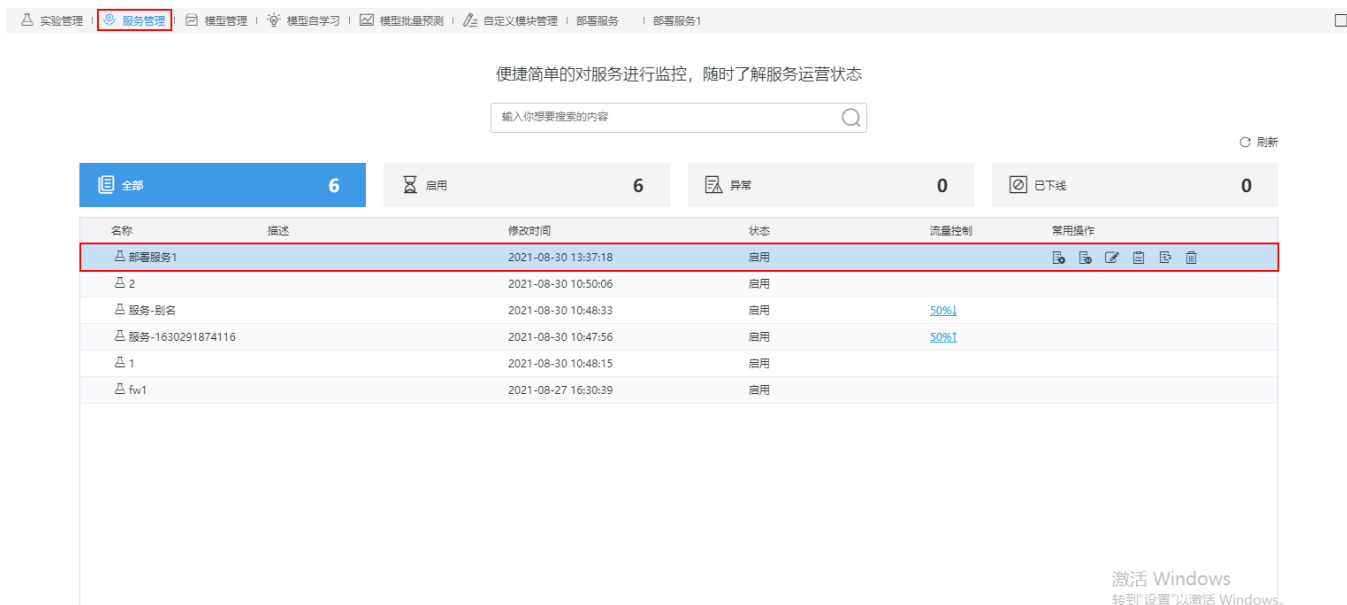
5、选择新增服务并输入服务名称。



6、选择“服务测试”页面，点击 **测试** 按钮对服务进行应用。



7、选择服务管理界面，点击 **刷新** 按钮，可对新建的服务进行启用、下线、编辑等操作，详情请参考 [服务监控](#)。

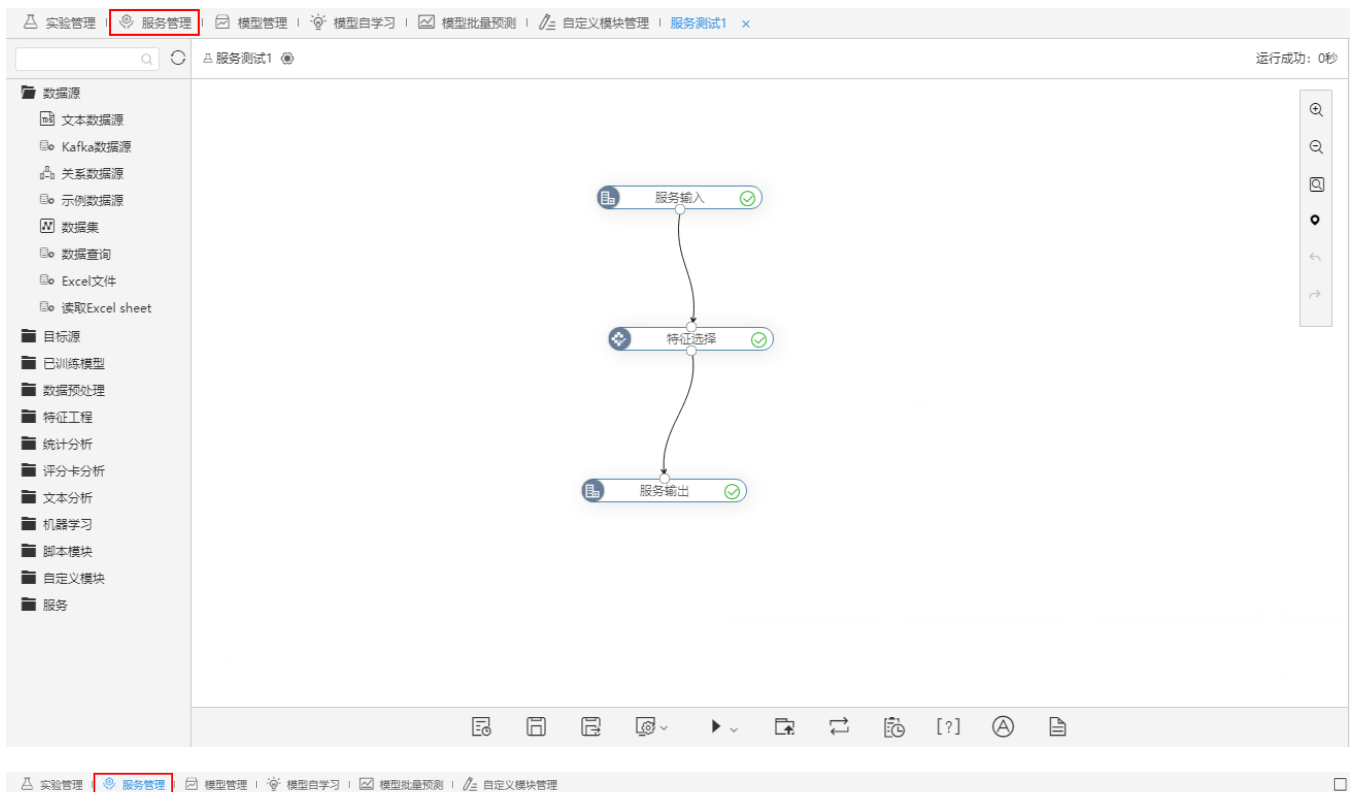


## 服务监控

服务监控是指对所有已经搭建的服务工作流程进行管理。

功能入口：在“数据挖掘”界面中选择 **服务管理** 页签，显示“服务管理”界面，如下图所示：





便捷简单的对服务进行监控，随时了解服务运营状态





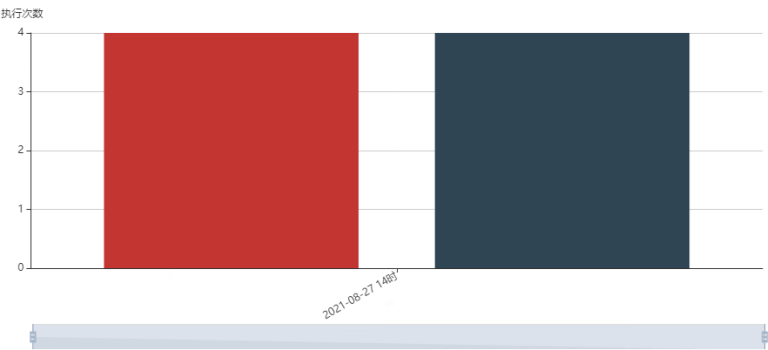


该界面分为如下几个区：


- 功能区：用于手动刷新列表和搜索。

操作	图标	说明
下载SDK		用于将服务提供给其他系统调用。
手动刷新		手动刷新服务列表。
搜索服务		服务流程名称关键字模糊匹配搜索结果。

- 状态栏：显示所有服务流程不同状态的统计数据。
- 列表区：显示服务流程列表，支持如下操作：

操作	图标	说明
服务启用		用于启用当前服务流程。 若当前服务呈启用状态，该按钮置灰。
服务下线		用于设置当前服务下线。 若当前服务呈下线状态，该按钮置灰。
编辑		用于编辑当前服务，进入“服务配置”界面（如下图所示），支持修改服务的别名和描述。 <div data-bbox="349 554 1300 879" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>服务配置 服务测试</p> <p>服务ID: I8a8aefa3017b86ba86ba455c017b958fde730041</p> <p>服务名称: 部署服务1</p> <p>服务别名: <input type="text" value="部署服务1"/></p> <p>内部调用地址: https://10.10.11.25:8901/api/v1/services/I8a8aefa3017b86ba86ba455c017b958fde730041</p> <p>实例ID: I8a8aefa3017b86ba86ba455c017b86d49d180008</p> <p>实例ID: I8a8aefa3017b86ba86ba455c017b86e0199w000a</p> <p>服务描述: <input type="text"/></p> <p><input type="button" value="保存"/></p> </div>
服务调用统计		用于对某个时间段服务调用执行记录进行统计。 <div data-bbox="349 942 1300 1436" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>服务调用统计[名称: 部署服务]</p> <p>查询方式: 按小时 <input type="text" value="2021-05-29 00:00:00"/> ~ <input type="text" value="2021-08-27 23:59:59"/></p> <p>服务统计 <span style="color: red;">■</span> 成功 <span style="color: darkblue;">■</span> 失败 <span style="float: right;">折线图 柱图</span></p>  <p style="text-align: center;">2021-08-27 14:00</p> <p style="text-align: right;"><input type="button" value="确定(O)"/> <input type="button" value="取消(C)"/></p> </div> <ul style="list-style-type: none"> <li>查询方式：按照月份、日期、小时三个时间段查询；</li> <li>服务统计：通过折线图、柱图展现统计的某个时间段成功、失败的条数。</li> </ul>

<p>服务调用记录</p> 	<p>记录了服务调用的服务器名、开始时间、结束时间、执行时间、运行结果、错误日志等信息。可用于监控模型的运行时长、稳定性等。</p> <div data-bbox="349 205 1307 640" style="border: 1px solid #ccc; padding: 5px;"> <p>服务调用记录[名称: 部署服务]</p> <table border="1"> <thead> <tr> <th>服务器名</th> <th>开始时间</th> <th>结束时间</th> <th>执行时间</th> <th>运行结果</th> <th>错误日志</th> </tr> </thead> <tbody> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:48:53</td> <td>2021-08-27 14:48:53</td> <td>0秒</td> <td>执行成功</td> <td>---</td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:48:51</td> <td>2021-08-27 14:48:51</td> <td>0秒</td> <td>执行成功</td> <td>---</td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:41:18</td> <td>2021-08-27 14:41:18</td> <td>0秒</td> <td>失败</td> <td><a href="#">错误日志</a></td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:40:53</td> <td>2021-08-27 14:40:53</td> <td>0秒</td> <td>失败</td> <td><a href="#">错误日志</a></td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:40:50</td> <td>2021-08-27 14:40:50</td> <td>0秒</td> <td>失败</td> <td><a href="#">错误日志</a></td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:39:39</td> <td>2021-08-27 14:39:39</td> <td>0秒</td> <td>失败</td> <td><a href="#">错误日志</a></td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:39:14</td> <td>2021-08-27 14:39:14</td> <td>0秒</td> <td>执行成功</td> <td>---</td> </tr> <tr> <td>host-10-10-111-36</td> <td>2021-08-27 14:39:03</td> <td>2021-08-27 14:39:04</td> <td>1秒</td> <td>执行成功</td> <td>---</td> </tr> </tbody> </table> <p style="text-align: right;">共 8 条 10条/页 &lt; &gt; 前往 1 / 1页</p> <p style="text-align: right;"><a href="#">确定(O)</a> <a href="#">取消(C)</a></p> </div> <p>错误日志: 在资源有限的情况下, 用于排查某个模型服务卡死。只有执行失败的服务记录才会有错误日志, 点击错误日志即可下载日志到本地查看。</p>	服务器名	开始时间	结束时间	执行时间	运行结果	错误日志	host-10-10-111-36	2021-08-27 14:48:53	2021-08-27 14:48:53	0秒	执行成功	---	host-10-10-111-36	2021-08-27 14:48:51	2021-08-27 14:48:51	0秒	执行成功	---	host-10-10-111-36	2021-08-27 14:41:18	2021-08-27 14:41:18	0秒	失败	<a href="#">错误日志</a>	host-10-10-111-36	2021-08-27 14:40:53	2021-08-27 14:40:53	0秒	失败	<a href="#">错误日志</a>	host-10-10-111-36	2021-08-27 14:40:50	2021-08-27 14:40:50	0秒	失败	<a href="#">错误日志</a>	host-10-10-111-36	2021-08-27 14:39:39	2021-08-27 14:39:39	0秒	失败	<a href="#">错误日志</a>	host-10-10-111-36	2021-08-27 14:39:14	2021-08-27 14:39:14	0秒	执行成功	---	host-10-10-111-36	2021-08-27 14:39:03	2021-08-27 14:39:04	1秒	执行成功	---
服务器名	开始时间	结束时间	执行时间	运行结果	错误日志																																																		
host-10-10-111-36	2021-08-27 14:48:53	2021-08-27 14:48:53	0秒	执行成功	---																																																		
host-10-10-111-36	2021-08-27 14:48:51	2021-08-27 14:48:51	0秒	执行成功	---																																																		
host-10-10-111-36	2021-08-27 14:41:18	2021-08-27 14:41:18	0秒	失败	<a href="#">错误日志</a>																																																		
host-10-10-111-36	2021-08-27 14:40:53	2021-08-27 14:40:53	0秒	失败	<a href="#">错误日志</a>																																																		
host-10-10-111-36	2021-08-27 14:40:50	2021-08-27 14:40:50	0秒	失败	<a href="#">错误日志</a>																																																		
host-10-10-111-36	2021-08-27 14:39:39	2021-08-27 14:39:39	0秒	失败	<a href="#">错误日志</a>																																																		
host-10-10-111-36	2021-08-27 14:39:14	2021-08-27 14:39:14	0秒	执行成功	---																																																		
host-10-10-111-36	2021-08-27 14:39:03	2021-08-27 14:39:04	1秒	执行成功	---																																																		
<p>删除</p> 	<p>用于删除当前服务流程。</p>																																																						

 服务监控只能使用ClickHouse高速缓存库, 如果更改了错误的高速缓存库, 需要重启服务引擎后才能使用。

## 服务调用示例

调用流程说明:

1. 示例附件 ([ServiceInvokeDemo.zip](#)) 解压后, 得到Demo目录结构如下:



2. 各目录说明

- 1) doc——文档目录, demo说明文档在该目录下。
- 2) example——服务示例目录, 自带一个可运行的服务Dag示例, 可以在平台中进行流程导入, 并部署成服务。
- 3) java——java语言编写的demo工程, 该项目为maven工程, 可以使用开发工具以maven项目的形式进行导入, maven环境自行安装。
- 4) python——python语言编写的demo工程, 开发工具以python项目的形式进行导入, python环境自行安装(推荐使用anaconda3和python3.6及其以上版本)。

3. 调用执行

- 1) java语言, 成功导入工程后, 在ServiceInvokeDemo.java文件中根据实际调用的服务环境, 修改对应的ip、端口、服务ID和输入数据, 然后运行。运行成功返回如图所示:

```
Result: {"data": [{"a":1.0,"b":"a","c":1.0}, {"a":2.0,"b":"b","c":3.1}], "type":"dataTable"}
```

备注: 该图片返回数据是自带服务Dag示例的返回数据。如果不是使用该服务示例, 可能返回数据有一定差异。

- 2) python语言, 成功导入工程后, 在ServiceInvokeDemo.py文件中根据实际调用的服务环境, 修改对应的ip、端口、服务ID和输入数据, 然后运行。运行成功返回如图所示:

```
{"data": [{"a":1.0,"b":"a","c":1.0}, {"a":2.0,"b":"b","c":3.1}], "type":"dataTable"}
```

备注: 该图片返回数据是自带服务Dag示例的返回数据。如果不是使用该服务示例, 可能返回数据有一定差异。

