

数据接入介绍

目 录

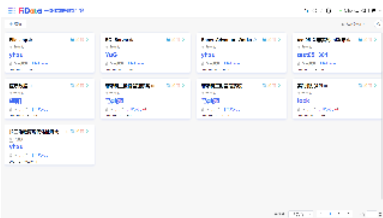
- 1. 概述
- 2. 界面概览
- 3. 数据接入
 - 3.1 创建非实时应用
 - 3.1.1 创建MySQL、SQL Server、Oracle驱动类型的应用
 - 3.1.2 创建ftp驱动类型的应用
 - 3.1.3 创建API驱动类型的应用
 - 3.1.4 使用非实时API
 - 3.2 创建实时应用
 - 3.2.1 创建RestfulAPI驱动类型的应用
- 4. 查看应用日志

1. 概述

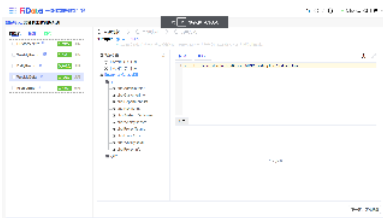
数据接入功能提供可视化的界面，方便用户进行数据的接入，支持SQL Server、Oracle、MySQL、Excel、CSV等多种数据源，数据支持以全量、增量、业务主键覆盖等方式将数据同步到本平台中

2. 界面概览

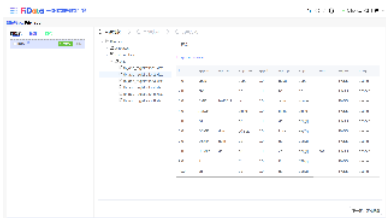
入口



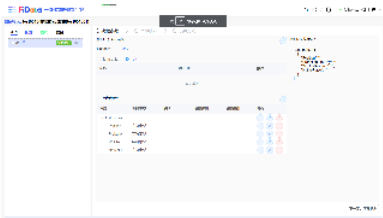
物理表



FTP页面



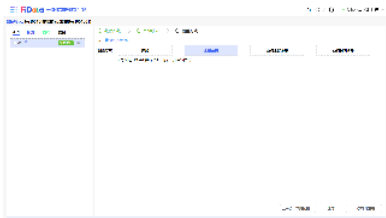
API页面



字段映射



覆盖方式



3. 数据接入

3.1 创建非实时应用

3.1.1 创建MySQL、SQL Server、Oracle驱动类型的应用

(1) **创建应用**： 点击添加，输入应用基本信息、联系人、驱动类型、服务器配置后，点击测试连接，连接通过，即可创建应用



应用名称：中英文名称皆可

应用简称：一般是英文名称，类似数据库的架构名称

应用负责人：一般是该系统的IT负责人，方便后续在数据接入的过程中，如果发现上游系统无法正常访问时，能够第一时间得到相应帮助

负责人邮箱：同应用负责人作用一样

FiData

一体化数据赋能平台

feihongz 欢迎回来

数据集成: 添加应用

基本

数据源

应用类型:

非实时应用

实时应用

主要以平台主动抽取的方式从上游系统中拿到数据。对于数据的时效性要求不高的系统采用此种方式

基本信息:

* 应用名称

* 应用简称

应用描述

联系人:

* 应用负责人

* 联系人一般填写该系统的IT负责人信息，方便后续在数据接入的过程中，如果发现上游系统无法正常访问时能够第一时间得到相应的帮助

* 负责人邮箱

下一步: 数据源

驱动类型：本平台支持的几种上游系统，如mysql、sqlserver、oracle、ftp文件系统、非实时API调用、RestfulAPI实时推送

服务器配置：本平台同步数据时，用于连接上游系统的服务器基本信息

FiData 一体化数据赋能平台

🔔 ⚙️ ? 🗣️ feihongz 欢迎回来

数据集成: 添加应用

基本 > 数据源

驱动类型

mysql

mysql

sqlserver

oracle

ftp

api

选择驱动类型

服务器配置:

* 服务器地址

ftp

* 服务器地址说明

* 端口

* 端口说明

* 数据库

* 数据库说明

* 连接字符串

jdbc:mysql://

* 字符串说明

* 连接账号

* 账号说明

* 连接密码

* 填写密码说明

测试连接

上一步

保存

(2) 创建物理表

1.新增一张物理表 → 2.选择新增的未命名表 → 3.在页面的SQL编辑器中输入业务SQL → 4.进入字段映射配置页面；

FiData 一体化数据赋能平台

🔔 ⚙️ ? 🗣️ 赵飞鸿 欢迎回来

数据接入: test0524

物理表

新增 发布

store

发布成功

删除

person

发布成功

删除

dd

发布成功

删除

cs

发布成功

删除

address_0706

发布成功

删除

未命名

未发布

删除

2

来源对象

字段映射

覆盖方式

来源类型: 自定义脚本

未能实现从该应用对接表或者视图，自定义脚本支持自定义的SQL脚本实现从上游对接数据

系统变量

上次同步开始时间

上次同步结束时间

AdventureWorks

表

dbo.DatabaseLog

dbo.ErrorLog

dbo.AWBuildVersion

dbo.CALENDAR_INFO

dbo.test

dbo.dim_time_table

HumanResources.EmployeeDe

HumanResources.Department

HumanResources.Employee

HumanResources.Shift

HumanResources.EmployeePa

HumanResources.JobCandidat

Person.Person

Person.Password

Person.PersonPhone

Person.PhoneNumberType

Person.Address

执行 停止

SELECT * FROM Person.Address 3

SQL查询的数据

结果

AddressID	Address...	Address...	City	StatePr...	PostalC...	Spatial...	rowguid	Modifie...
1	1970 Na...		Bothell	79	98011	E61000...	9AADC...	2007-12...
2	9833 Mt...		Bothell	79	98011	E61000...	32A54B...	2008-11...
3	7484 Ro...		Bothell	79	98011	E61000...	4C5069...	2013-03...
4	9539 Gl...		Bothell	79	98011	E61000...	E5946C...	2008-02...
5	1226 Sh...		Bothell	79	98011	E61000...	FBAFF9...	2008-12...
6	1399 Fir...		Bothell	79	98011	E61000...	FEBF81...	2009-02...

4

下一步: 字段映射

5.对字段进行配置 → 6.进入覆盖方式页面

数据类型：

- 文本：
 - 存放文本类型数据
 - 设置长度
- 整型：
 - 存放数值类型，无小数类型
- 时间戳：
 - 存放时间类型数据
- 浮点数：
 - 存放小数类型数据
 - 设置精度
- 字符串：
 - 存放字符串类型数据
 - 设置长度

FiData 一体化数据赋能平台

数据接入: test0524

物理表 新增 发布

store

发布成功

删除

person

发布成功

删除

dd

发布成功

删除

cs

发布成功

删除

address_0706

发布成功

删除

未命名

未发布

删除

来源对象 字段映射 覆盖方式

源: SQL输出

目标: 未命名

AddressID

AddressLine1

AddressLine2

City

StateProvinceID

PostalCode

SpatialLocation

rowguid

ModifiedDate

AddressID

AddressLine1

AddressLine2

City

StateProvinceID

PostalCode

SpatialLocation

rowguid

ModifiedDate

源字段	目标字段名称	目标字段描述	目标字段类型	目标字段长度	目标业务主键
AddressID	AddressID		整型	0	☒
AddressLine1	AddressLine1		字符串型	60	☒
AddressLine2	AddressLine2		字符串型	60	☒
City	City		字符串型	30	☒
StateProvinceID	StateProvinceID		整型	0	☒
PostalCode	PostalCode		字符串型	15	☒
SpatialLocation	SpatialLocation		字符串型	255	☒
rowguid	rowguid		字符串型	255	☒
ModifiedDate	ModifiedDate		日期时间型	8	☒

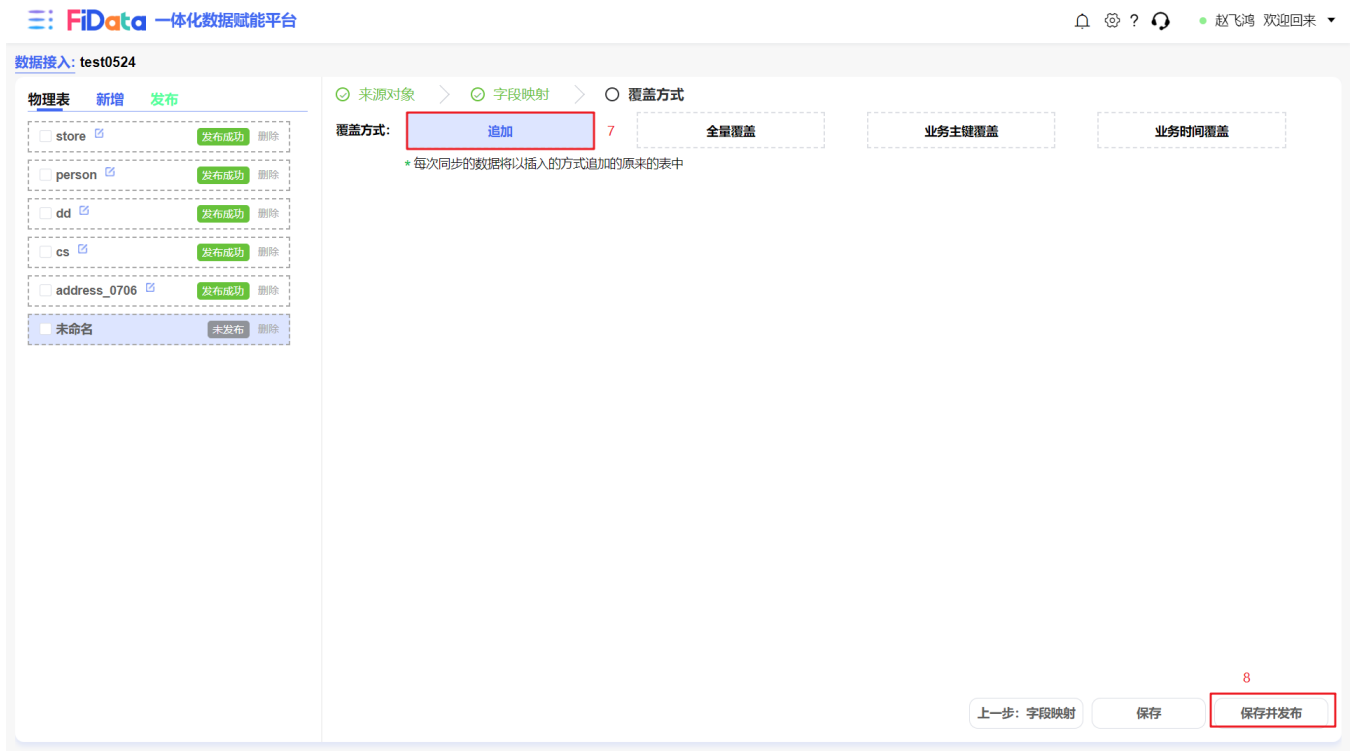
5: 对字段进行配置

6

上一步: 来源对象

下一步: 覆盖方式

7.选择同步数据的覆盖方式→ 8.保存&发布物理表



9.填写表基本信息（表名、表描述）



10.发布



同步方式：提供追加、全量覆盖、业务主键覆盖和业务时间覆盖等四种同步方式

- 追加：每次同步的数据将以插入的方式追加的原来的表中



- 全量：每次同步时系统将清空原来的表，并重新插入

数据接入: test0524

物理表 新增 发布

☐ store	发布成功	删除
☐ person	发布成功	删除
☐ dd	发布成功	删除
☐ cs	发布成功	删除
☐ address_0706	发布成功	删除
☒ tb_entity	发布成功	删除

来源对象 字段映射 覆盖方式

覆盖方式:

追加

全量覆盖

业务主键覆盖

业务时间覆盖

每次同步时系统将清空原来的表，并重新插入

上一步: 字段映射

保存

保存并发布

- 业务主键覆盖：系统将根据当前设定的业务主键进行数据覆盖

数据接入: test0524

物理表 新增 发布

☐ store	发布成功	删除
☐ person	发布成功	删除
☐ dd	发布成功	删除
☐ cs	发布成功	删除
☐ address_0706	发布成功	删除
☒ tb_entity	发布成功	删除

来源对象 字段映射 覆盖方式

覆盖方式:

追加

全量覆盖

业务主键覆盖

业务时间覆盖

* 系统将根据当前设定的业务主键: 【 】进行数据覆盖

上一步: 字段映射

保存

保存并发布

- 业务时间覆盖：系统将根据以下选择的条件先从表中删除满足条件的数据，然后重新从插入满足条件的数据

FiData 一体化数据赋能平台

数据接入: test0524

物理表

新增

发布

store

发布成功

删除

person

发布成功

删除

dd

发布成功

删除

cs

发布成功

删除

address_0706

发布成功

删除

tb_entity

发布成功

删除

来源对象

字段映射

覆盖方式

覆盖方式:

追加

全量覆盖

业务主键覆盖

业务时间覆盖

* 系统将根据以下选择的条件先从表中删除满足条件的数据, 然后重新从插入满足条件的数据

覆盖

TIMESTAMP

大于

T +

请输入内容

请选择

的数据

假如在

每年

每月

每天

请选择

月之前同步数据

覆盖

TIMESTAMP

大于

T +

请输入内容

请选择

的数据

否则

覆盖

TIMESTAMP

大于

T +

请输入内容

请选择

的数据

* 备注: T代表当前时间

上一步: 字段映射

保存

保存并发布

3.1.2 创建ftp驱动类型的应用

(1) 选择ftp驱动类型, 填写ftp服务器基本信息, 测试连接, 连接通过即可保存应用

FiData 一体化数据赋能平台

数据集: 编辑应用

基本

数据源

* 驱动类型

ftp

* 选择驱动类型: ftp

服务器配置:

* 服务器地址

192.168.1.94

* 服务器地址说明

* 端口

21

* 端口说明

* 连接账号

ftpuser

* 账号说明

* 连接密码

* 填写密码说明

* 文件后缀名

excel

* 文件后缀说明

选择读取的文件: 如选择excel, 本平台后续只会读取上游系统中的xls和xlsx文件
选择csv,则读取csv文件

上一步

保存

(2) 创建物理表

- 1.新增一张物理表
 - 2.选择新增的未命名表
 - 3.选择本次需要同步的excel/csv文件
 - 4.预览文件内容
 - 5.选择本次同步数据的sheet页
 - 6.进入字段映射页面
- 其他操作流程同mysql驱动类型创建物理表一致

FiData 一体化数据赋能平台

数据接入: test20220107ftp

物理表

新增 1 发布

1

未发布

删除

2

未发布

删除

3

未发布

删除

test0412

发布成功

删除

4

未发布

删除

test0414

发布成功

删除

cs11

发布成功

删除

amdsss

发布成功

删除

未命名

未发布

删除

2

来源对象

字段映射

覆盖方式

test20220107ftp

Windows

testDirectory

child 3

tb_app_registration1.xlsx

tb_app_registration2.xlsx

tb_app_registration3.xlsx

tb_app_registration4.xlsx

tb_app_registration5.xlsx

tb_app_registration6.xlsx

预览 4

tb_app_registration 5 test Sheet1

id	atlas_i...	app_n...	app_a...	app_des	app_t...	app_p...	app_p...	sync_...	expre...
1.0	10fc3c...	appa		appa	0.0	appa	appa		PASS
2.0		bb1		bb	0.0	bb	bb		PASS
3.0		EMP	test0011	cc	0.0	admin	admin...		PASS
4.0		AMQP		string	0.0	string	string		PASS
5.0		dd		dd	aa	dd	dd@q...		PASS
6.0	1479b...	DEMP...	dim	实时应用	0.0	Lock	ee@q...		PASS
7.0		DEMP	dem	e2	0.0	ee	ee@q...		PASS
8.0		DEMP1	dd	e3	0.0	ee	ee@q...	aa	PASS
9.0		ff		ff	0.0	ff	ff@qq...		PASS
10.0		gg		gg	0.0	gg	gg@q		PASS

6

下一步: 字段映射

3.1.3 创建API驱动类型的应用

FiData 一体化数据赋能平台

feihongz 欢迎回来

数据集成: 编辑应用

基本

数据源

驱动类型

api

选择驱动类型: api

服务器配置:

身份验证方式

☐ OAuth 1.0

☐ OAuth 2.0

☒ JWT

☐ Bearer Token

☐ 无需身份验证

选择身份验证方式: 以JWT为例

身份验证地址:

http://192.168.11.140:8080/api/get

配置身份验证地址

参数

连接账号

zhaofeihong

配置账号及密码

连接密码

上一步

保存

(1) 创建API: 点击页面“新增”，输入API基本信息，即可创建API

API名称: 中英文名称皆可。

API地址: 一般是API的访问地址。

请求方式: 调用API时指定的请求方式。

JSON定义: API的JSON解析根节点定义。

FiData 一体化数据赋能平台

赵飞鸿 欢迎回来

数据接入: 非实时示例(MDM接口)(asdm dm)

API

新增

发布

复制

人员管理-剔除

发布成功

删除

cs

发布成功

删除

规则示例

发布成功

删除

新增API

应用名称

非实时示例(MDM接口)

API名称

0/50

API地址

0/150

请求方式

☐ GET

☐ POST

JSON定义:

当前api的json解析的根节点定义

0/20

API描述

0/100

取消

确定

(2) 配置API请求方式

FiData 一体化数据赋能平台

数据接入: 非实时示例(MDM接口)(asdmmdm)

API 新增 发布 复制

- 人员管理-别删 [发布成功] [删除]
- cs [发布成功] [删除]
- 规则示例 [发布成功] [删除]
- 商品详情 [未发布] [删除]

选中新增的api

api的请求头配置

api的请求体配置

数据参数 (POST) 请求参数: Headers Body

form-data raw

请求体支持form表单和json结构

JSON 结构预览:

KEY	VALUE	操作
暂无数据		

JSON结构定义:

字段	字段类型	描述	推送规则	推送示例	操作
暂无数据					

下一步: 字段映射

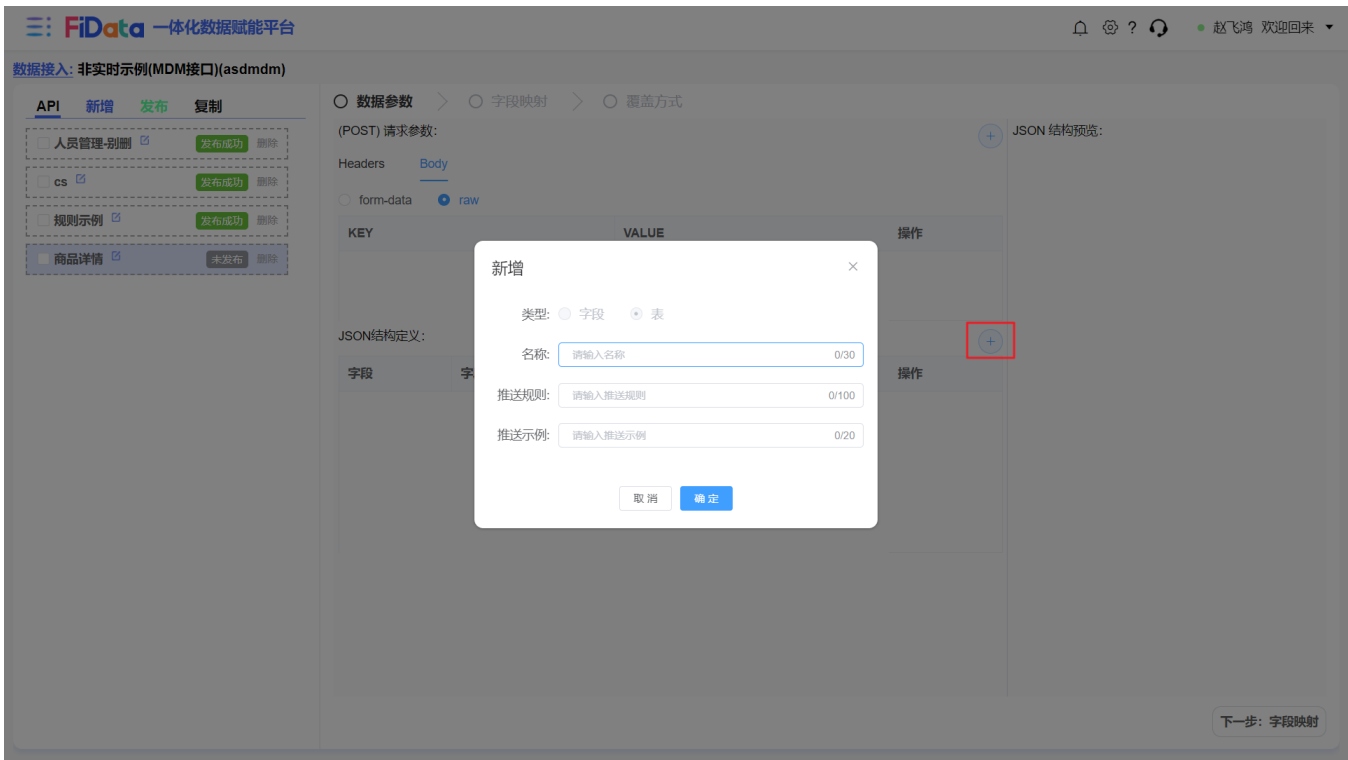
(3) 配置API的JSON结构定义

名称: 一般是表英文名称or字段英文名称

类型: 字段类型

推送规则: 一般是字段的推送规则描述, 如字段不为空

推送示例: 一般是API推送的测试数据, 如设置了name字段推送示例为“上海”, 在发布API的时候选择按照配置调用一次API, 将name字段赋值“上海”



字段映射

数据接入: 非实时示例(MDM接口)(asdmmdm)

API 新增 发布 复制

- ☐ 人员管理-别删 ☒ 发布成功 删除
- ☐ cs ☒ 发布成功 删除
- ☐ 规则示例 ☒ 发布成功 删除
- ☐ 商品详情 ☒ 未发布 删除

数据参数 字段映射 覆盖方式

(POST) 请求参数:

Headers Body

form-data raw

KEY	VALUE	操作
暂无数据		

JSON结构定义:

字段	字段类型	描述	推送规则	推送示例	操作
tb_store					
id	值类型				
name	字符串型				
age	字符串型				
tb_addr...	字符串型				
add_n...	字符串型				

JSON 结构预览:

```
{
  "tb_store": [
    {
      "id": "",
      "name": "",
      "age": "",
      "tb_address": [
        {
          "add_name": "",
          "create_time": "",
          "username": ""
        }
      ]
    }
  ]
}
```

下一步: 字段映射

比如配置的JSON结构为一主一从结构, 那么字段映射页面就要配置两张表的字段

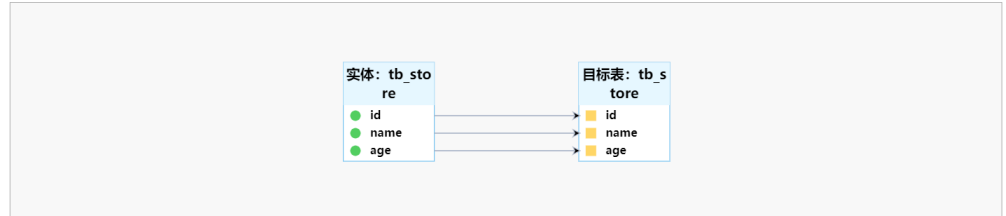
数据接入: 非实时示例(MDM接口)(asdmmdm)

API 新增 发布 复制

- ☐ 人员管理-别删 ☒ 发布成功 删除
- ☐ cs ☒ 发布成功 删除
- ☐ 规则示例 ☒ 发布成功 删除
- ☐ 商品详情 ☒ 未发布 删除

数据参数 字段映射 覆盖方式

tb_store tb_address



源字段	字段源类型	目标字段名称	目标字段描述	目标字段类型	目标字段长度	目标业务主键
id	值类型	id		字符串型	200	☒
name	字符串型	name		字符串型	200	☒
age	字符串型	age		字符串型	200	☒

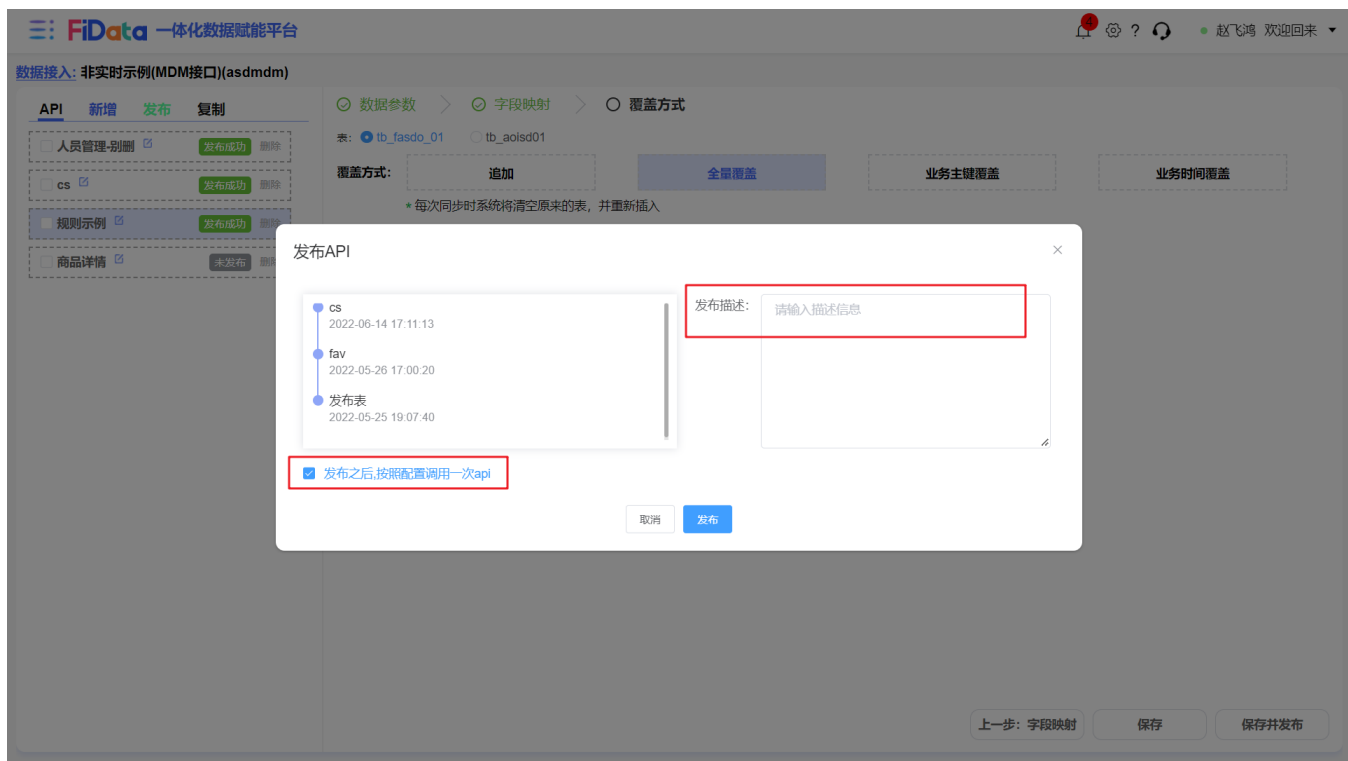
上一步: 数据参数

下一步: 覆盖方式

覆盖方式同时也需要配置多表



勾选选项，发布后，会根据配置调用一次API，并将API的数据推送到本平台中



3.1.4 使用非实时API

1.点击“开始”并配置，选择“数据湖表任务”→ 2.选择非实时API应用



3.点击“+”，添加API任务→ 4.选择具体的API，支持多API操作→5.指定到右边，方便调整API的调用顺序→ 6.保存API



7.保存并发布

FiData 一体化数据赋能平台

🔔

🔗

?

🎧

feihongz 欢迎回来

管道: 默认管道

管道首页

🕒 开始

📁 任务组

📊 数据湖表任务

📊 数仓表任务

📊 分析模型任务

开始: 开始
TIMER DRIVEN
10

数据湖表任务:
数据湖表任务
非实时示例(MDM接口)

常规

数据表

应用: 非实时示例(MDM接口)

表: 规则示例 × +

属性

名称: test0615_cos

负责人: Lock

说明:

保存 保存&发布

7

导航器

FiData
TIMER DRIVEN
10

非实时示例(MDM接口)
非实时示例(MDM接口)

管道的创建和基本使用在数据调度介绍文档。

3.2 创建实时应用

3.2.1 创建RestfulAPI驱动类型的应用

(1)创建API 及构建json结构同上述方法一致，本平台实时API没有配置API请求方式

FiData 一体化数据赋能平台

赵飞鸿 欢迎回来

数据集成: 编辑应用

基本

身份验证

接入类型

RestfulAPI

选择驱动类型: RestfulAPI

身份验证:

账号

lock

密码

配置api的账号及密码, 用于获取访问本平台的临时token

账号和密码用于刷新和获取令牌

上一步

保存

(2) 使用实时API

实时测试API

负责人

lock

应用类型

实时应用

2022-02-25 19:08:14

生成接口文档

PDF

点击生成文档，在文档中有详细的调用方式。

以postman为例：

- 1.复制pdf文档中获取临时token的接口地址；
- 2.获取当前应用的账号及密码；
- 3.使用postman，填写接口地址，填写请求参数的账号及密码，调用接口，获取临时token（有效期60min）；

4.查看pdf文档中推送数据的接口地址，在postman中填写地址，填写请求头信息，并将临时token写入请求头中；

5.根据pdf文档中的接口构造好需要推送的数据，调用接口即可， 可通过接口返回值查看接口的调用结果，也可以在数据接入首页应用查看日志。

The image shows a web application interface for configuring data integration. The top section displays API documentation for a '登录授权' (Login Authorization) interface. The documentation includes sections for '接口地址' (Interface Address), '获取Token接口' (Get Token Interface), '简要描述' (Brief Description), '请求方式' (Request Method), 'Content-Type', '参数 (header)' (Parameters), and '请求示例' (Request Example). A red box highlights the '接口地址' and '获取Token接口' sections, with a red '1' next to it.

The bottom section shows the 'FiData 一体化数据赋能平台' (FiData Integrated Data Empowerment Platform) interface. It includes a '数据集成: 编辑应用' (Data Integration: Edit Application) header and a '身份验证' (Authentication) tab. The '身份验证' tab contains a form for configuring authentication. A red box highlights the '身份验证' section, with a red '2' next to it. The form includes fields for '账号' (Account) and '密码' (Password), and a '接入类型' (Access Type) dropdown menu.

The '身份验证' section includes the following fields and options:

- * 接入类型: RestfulAPI
- * 账号: lock
- * 密码: *****

A note below the form states: * 账号和密码用于刷新和获取令牌 (Account and password are used for refreshing and obtaining tokens).

The bottom right of the interface features two buttons: '上一步' (Previous Step) and '保存' (Save).

POST 获取临时token

POST 推送数据

No Environment

实时API / json格式的数据 / 获取临时token 3

Save

POST http://192.168.11.130:8083/dataAccess/apiConfig/getToken... 填写获取临时token的地址

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "useraccount": " ",
3   "password": " ",
4 }
```

填写账号

Body Cookies Headers (6) Test Results Status: 200 OK Time: 638 ms Size: 520 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": 0,
3   "msg": "成功",
4   "data": " ",
5   "traceId": "9228aca5-d7d3-43bb-9c45-9a1c5a2f96bf"
6 }
```

得到临时token

POST 获取临时token

POST 推送数据

No Environment

实时API / json格式的数据 / 推送数据 4

Save

POST http://localhost:8083/dataAccess/apiConfig/pushdata 填写推送数据的接口，地址在生成的文档中

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

☒	User-Agent	PostmanRuntime/7.28.4
☒	Accept	*/*
☒	Accept-Encoding	gzip, deflate, br
☒	Connection	keep-alive
☒	Authorization	Bearer eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJlYmJlMzQwM...

Key Value Description

Response

Click Send to get a response

填写临时token

▼ / 实时API / json格式的数据 / 推送数据

Save ...

POST http://192.168.11.130:8083/dataAccess/apiConfig/pushdata Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON 5.根据pdf文档中的结构将数据构造好 Beautify

```
1 {
2   "apiCode": 94,
3   "pushData": {
4     "data": [
7       {
8         "id": "AE86",
9         "name": "设备111",
10        "age": "zhangsan"
11      },
12      {
13        "id": "AE86",
14        "name": "设备111",
15        "age": "lisi"
16      }
17     ]
18   }
19 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 1612 ms Size: 478 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": 0,
3   "msg": "成功",
4   "data": "成功: 数据推送到[stg_asdfasd_tb_test281]: 成功,推送的条数为: 2; 数据同步到[ods]: stg同步数据到ods报错。--[本次同步的数据为正式数据]",
5   "traceId": "7c4a65a6-44c9-48c2-bf8d-5332d03a6039"
6 }
```

推送后的日志,也可以在数据接入首页查看

4. 查看应用日志

The screenshot shows the '应用管理' (Application Management) page. At the top, there's a '管理控制台' (Management Console) header. Below it, a red box highlights the '查看应用日志' (View Application Logs) button for the 'File_app' application. The page lists several applications, each with its name, owner, application type, and creation time. The applications are: 'File_app' (owner: yhxu, type: 非实时应用, time: 2022-04-11 11:33:48), 'SQLServer' (owner: YuG, type: 非实时应用, time: 2022-07-05 15:22:52), 'Show_AdventureWorks' (owner: yhxu, type: 非实时应用, time: 2022-07-05 14:20:34), 'test0513-非实时api发布' (owner: test051301, type: 非实时应用, time: 2022-05-13 18:27:06), '医疗数据' (owner: 施阳, type: 非实时应用, time: 2022-04-11 13:42:41), '菲斯科主数据管理系统' (owner: 马灿荫, type: 实时应用, time: 2022-03-02 15:26:25), '菲斯科工时管理系统' (owner: 马灿荫, type: 非实时应用, time: 2022-03-02 15:12:32), '实时测试API' (owner: lock, type: 实时应用, time: 2022-02-25 19:27:08), and '长三角教育现代化检测大...' (owner: yhxu, type: 非实时应用, time: 2021-12-13 02:19:44).

[数据接入: 日志详情](#)

应用/业务域名称: File_app

[illegible]