

数据服务介绍

目 录

- 目 录
 - 1. 概述
 - 2. 界面概览
 - 3. 创建应用（下游系统）
 - 3.1 创建应用
 - 3.2 应用API服务订阅
 - 4. 创建API服务数据源
 - 5. 创建API服务
 - 5.1 界面功能描述
 - 5.2 配置API
 - 6. API服务调用日志

1. 概述

数据服务是我司自主研发的一款数据消费工具，提供可视化的服务配置界面，搭建统一的数据服务总线，帮助企业提升数据资产的价值，同时保证了数据的可靠性、安全性和有效性

2. 界面概览

入口



服务设计页面



数据源设计页面



3. 创建应用（下游系统）

3.1 创建应用

添加应用：添写基本信息后确定，会在页面新增一条数据

服务配置：点击进入服务配置页面，配置API服务



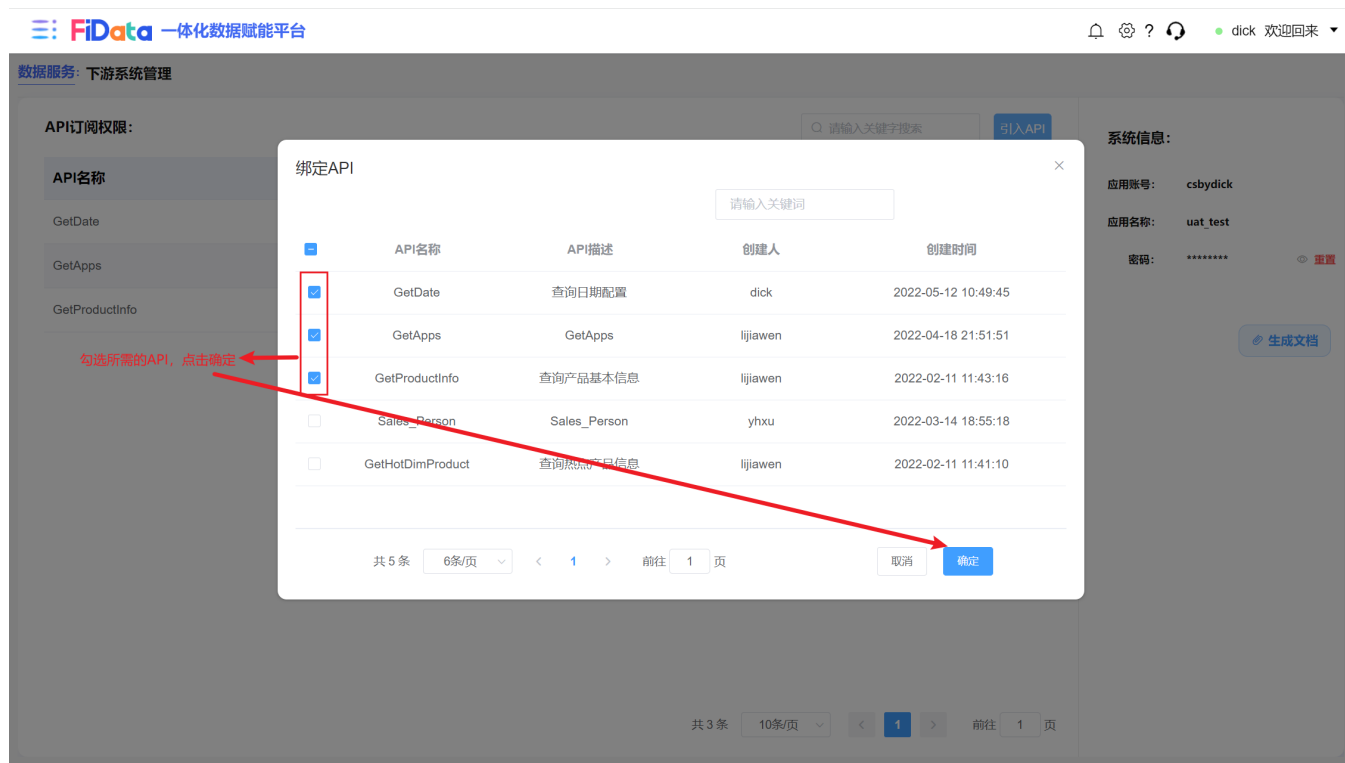
3.2 应用API服务订阅

点击应用进入API服务订阅界面，正文列表展示的是手动订阅过的API，右侧展示的是应用账号信息。

1. 引入API：弹出API服务页面，勾选需要的API服务点击确定，会在正文列表看到最新订阅的API信息，如需取消订阅可直接将启用状态切成禁用，或在引入API时取消已勾选的API服务
2. 启用/禁用：对应用下已订阅的API进行启用/禁用，禁用状态下的API服务在下游客户端不可访问
3. 参数配置：API服务内置参数配置，参数来源于该API的请求参数配置。可控制API的某些请求参数中是否体现在生成的API文档中，从而控制下游客户端调用时是否可见该参数
4. 生成文档：生成API服务的PDF版本的调用示例文档，默认在网页下以预览状态打开，支持在线下载以及联机打印。文档中含API入参出参信息、全局状态码信息、API调用示例信息，调用示例包含JAVA/.NET版本



1.引入API页面

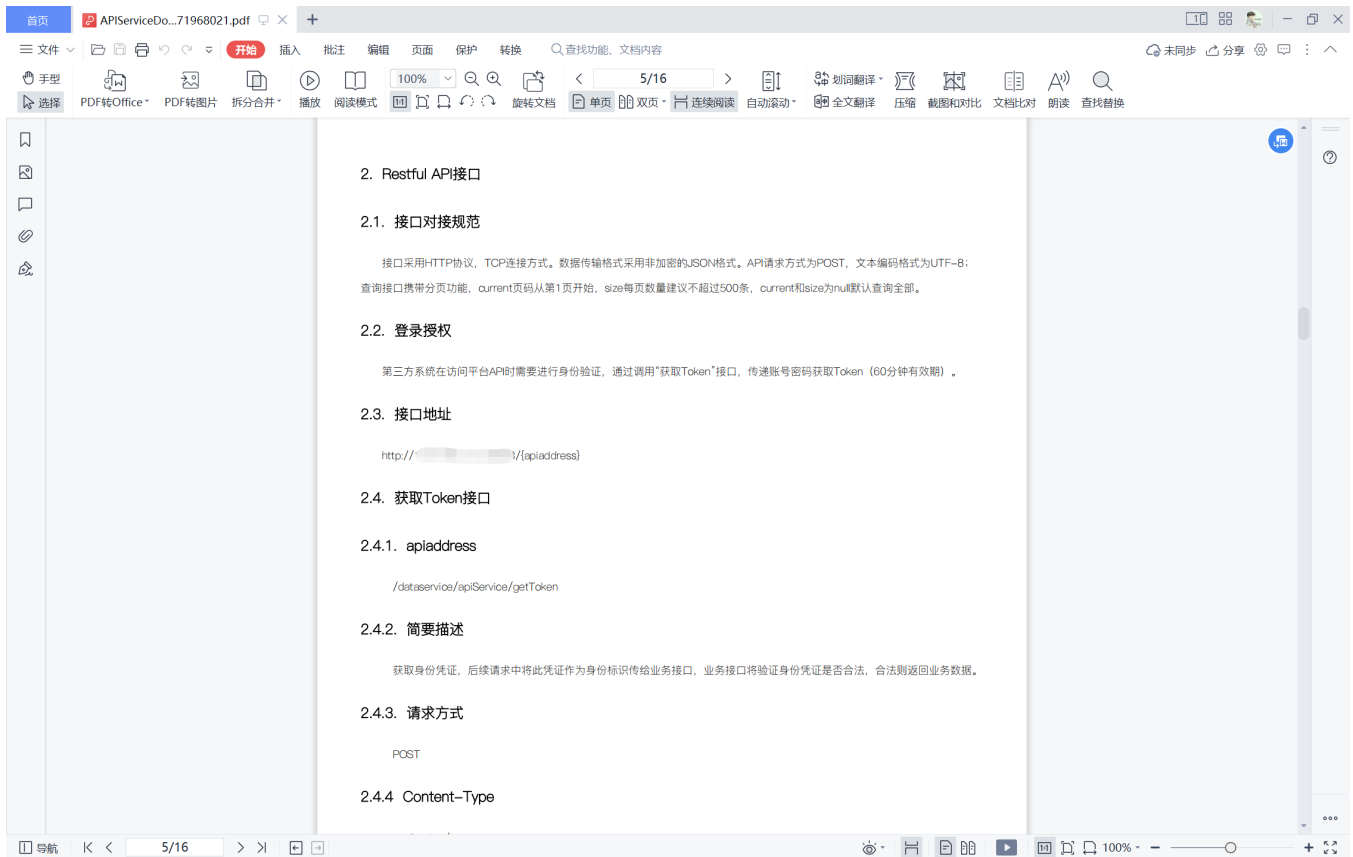


2.参数配置页面



3.生成文档案例（部分截图）





4. 创建API服务数据源

数据源列表

数据源ID	数据源名称	连接名称	数据库	服务器地址	操作
4.00	POSTORE	dmq_sdb_LUT	dmq_sdb_LUT	102.168.21.13	删除数据源失败
5.00	POSTORE	dmq_sdb_LUT	dmq_sdb_LUT	102.168.21.13	
3.00	SQLSERVER	sqlserver_31	AdventureWorks	102.168.11.133	
2.00	SQLSERVER	sqlserver-test	AdventureWorks2008R1	102.168.11.133	
1.00	MYSQL	mysql-test	dmq_database_test	102.168.11.130	

数据源新增/编辑

编辑数据源

* 连接名称: dmq_sdb_LUT

* 类型: POSTORE

* 用户名: 102.168.21.13

* 密码: 102.168.21.13

* 数据库: 102.168.21.13

* 端口: 3306

* 数据库: dmq_sdb_LUT

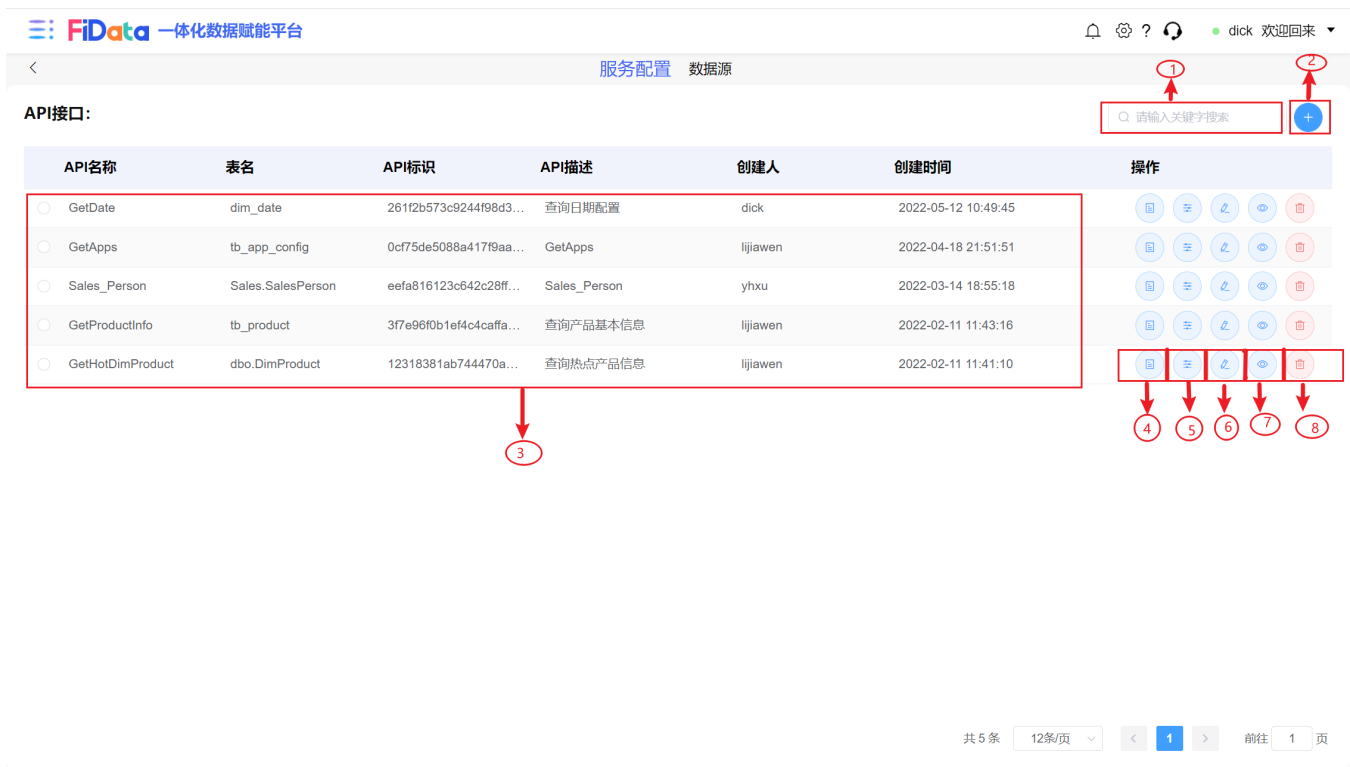
* 密码: 102.168.21.13

* 保存

5. 创建API服务

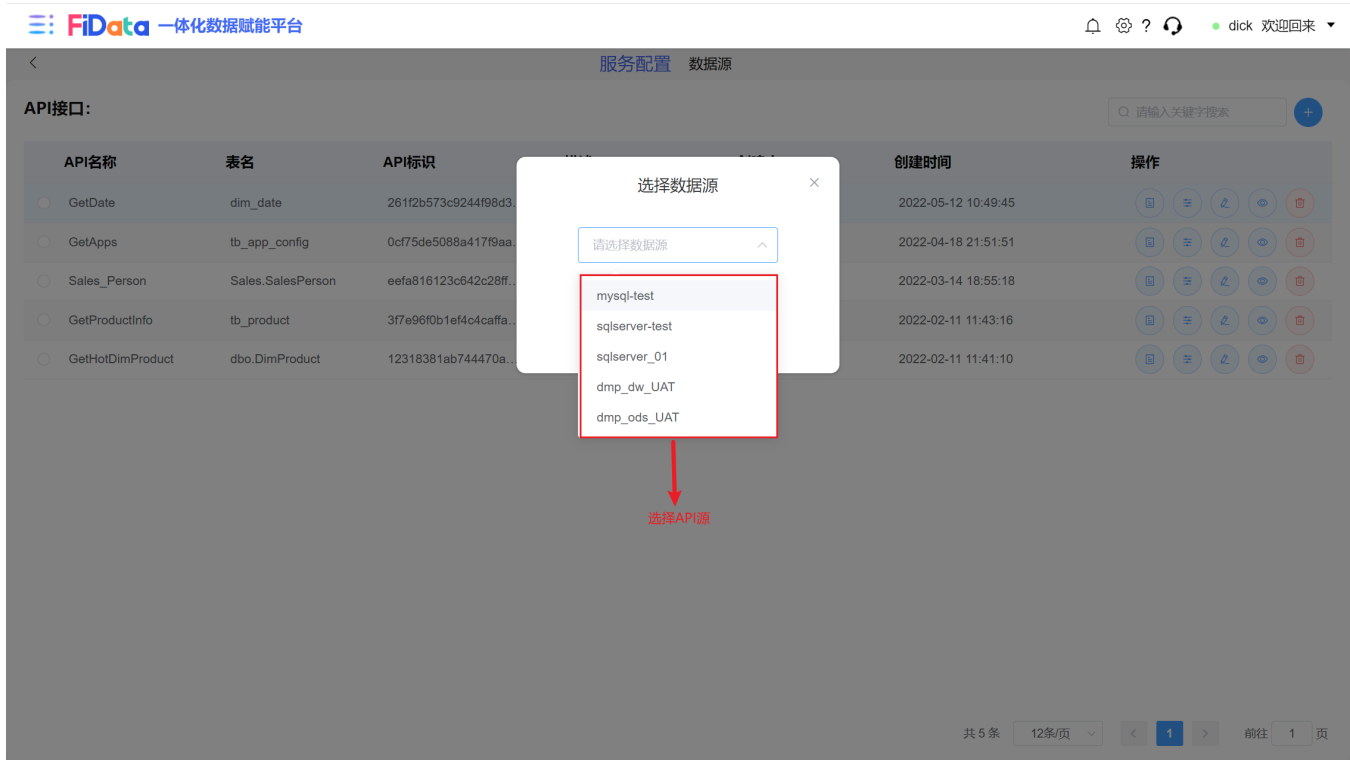
5.1 界面功能描述

1. 搜索符合条件的API
2. 新增API
3. API列表
4. 调用此API产生的调用日志
5. API字段参数自定义配置
6. 编辑API配置
7. 查看API配置
8. 删除API



5.2 配置API

1.点击右上角加号按钮新增API，弹出数据源选择窗口，选择数据源点击确定自动进入到API配置页面



2.API配置页面--配置普通API

FiData 一体化数据赋能平台

数据服务: 添加API接口

API定义

* API名称: GetProductInfo

* API描述: 查询产品基本信息

1 普通 2 自定义脚本

输入参数

参数名称	参数描述	参数值	操作
product_colour	产品颜色	blue	6

7

4

3

SQL脚本: 'product_colour'=@product_colour

SQL脚本: 1 id, product_name, product_price, product_stock, product_colour, create_time, create_user, del_flag

8

id	product_name	product_price	product_stock	product_colour	create_time	create_user	del_flag
2	海尔	6100.00	50	blue	2022-01-14 11:0...	lijiawen	1
14	微软Surface	7000.00	30	blue	2022-01-14 11:0...	lijiawen	1
15	微软Surface Lap...	7000.21	50	blue	2022-01-14 11:0...	lijiawen	1
16	微软surface boo...	7000.00	40	blue	2022-01-14 11:0...	lijiawen	1
17	微软 Surface La...	7800.96	55	blue	2022-01-14 11:0...	lijiawen	1
18	微软 Surface La...	7000.00	68	blue	2022-01-14 11:0...	lijiawen	1

9

保存

1. 选择配置的API类型，选择普通API，普通API为单表查询
2. 根据选择的数据源加载数据源Tree，Tree展示的内容为数据源的元数据信息
3. 配置API执行的SQL脚本，配置方式为拖拽左侧Tree上面某张表下面的字段
4. 配置API请求参数，配置方式为拖拽左侧Tree上面某张表下面的字段
5. 配置输入参数，输入参数即API请求参数，拖拽字段到第四步时，输入参数列表自动渲染加载出当前拖拽的字段，根据业务需求填写参数描述以及参数值
6. 删除配置参数，删除配置参数时，对应第四步展示的字也会自动删除，删除后API的查询结果将不受此参数条件限制
7. 执行：执行当前配置的API并返回前10条数据进行预览。停止：中断上一次执行过
8. 执行预览结果，执行成功后展示前10条数据进行预览
9. 保存，保存此API配置并在API列表中加载出刚配置的API

3.API配置页面-配置自定义API

FiData 一体化数据赋能平台

数据服务: 添加API接口

API定义

* API名称: GetProductInfo

* API描述: 查询产品基本信息

☐ 普通 ☒ 自定义脚本

数据:

- dmp_dataservice_db
 - tb_api_config
 - tb_app_api
 - tb_app_config
 - tb_builtin_parm
 - tb_datasource_config
 - tb_field_config
 - tb_filtercondition_config
 - tb_logs
 - tb_parm_config
 - tb_product
 - id
 - product_name
 - product_price
 - product_stock
 - product_colour
 - create_time
 - create_user

输入参数

参数名称	参数描述	参数值	操作
product_name	产品名称	联想	

▶ 执行 ▶ 停止

1 SELECT p.product_name,p.product_price,p.product_stock,p.pr FROM tb_product p WHERE p.product_name=@product_name

2

3

4

5

6

7

8

9

10

保存

1. 选择配置的API类型，选择自定义API，自定义API可单表可多表联合查询
2. 根据选择的数据源加载数据源Tree，Tree展示的内容为数据源的元数据信息
3. 配置API执行的SQL脚本，配置方式为拖拽左侧Tree上面某张表下面的字段或手动输入
4. 手动输入时支持使用表名或表别名点出该表下的字段
5. 配置输入参数，输入参数即API请求参数
6. 配置API请求参数，点击加号下方表格会新增一行，根据业务需求填写参数名称、参数描述以及参数值
7. 删除配置参数，删除后API的查询结果将不受此参数条件限制
8. 执行：执行当前配置的API并返回前10条数据进行预览。停止：中断上一次执行过程
9. 执行预览结果，执行成功后展示前10条数据进行预览
10. 保存，保存此API配置并在API列表中加载出刚配置的API

6. API服务调用日志

日志入口1：应用页面点击日志图表，查看该应用下的API调用日志情况

日志入口2：API列表页面点击日志图标，查看该API的调入日志情况

API日志:

🔍 请输入关键字搜索

应用名称	API名称	调用时间	传递参数	调用结果	返回数据行	信息	用户
uat_test	GetDate	2022-05-13 10:03:45	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-13 10:03:30	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:21	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:20	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:19	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:19	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:18	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:17	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:17	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:16	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:15	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick
uat_test	GetDate	2022-05-12 11:05:14	{"apiCode":"261f2b573c9244f98d3d..."}	成功	1		csbydick