

# 仓鼠数据服务平台

Hamster Data Service Platform

V8.0

使用手册

近咖软件

公司站点: <http://www.flyenter.com>

在线文档: <http://doc.flyenter.com>

## 目录

第 1 章	系统介绍 .....	3
1.1.	基本介绍 .....	3
1.2.	应用案例 .....	4
1.3.	应用场景 .....	17
第 2 章	系统部署 .....	17
2.1.	系统架构 .....	17
2.2.	系统规划 .....	19
2.3.	Windows 部署 .....	21
2.4.	Linux 部署 .....	24
第 3 章	环境配置 .....	34
3.1.	用户配置 .....	34
3.2.	平台配置 .....	35
3.3.	账户配置 .....	37
第 4 章	公用模块 .....	41
4.1.	数据源注册 .....	41
第 5 章	技术参数 .....	44
5.1.	变量定义 .....	44
5.2.	文件筛选 .....	49
5.3.	通用数据连接 .....	51
5.4.	数据格式 .....	54
第 6 章	常见问题 .....	61
6.1.	FTP/SFTP .....	61
6.2.	创建报表文件 .....	61
6.3.	企业微信审批 .....	63

# 第1章 系统介绍

作为一款专为企业数据集成与业务分析精心打造的卓越开发平台，全方位涵盖数据提取、清洗、深度剖析及精准发布等关键流程，同时巧妙整合诸多常用企业级应用服务，在确保数据处理过程透明、可控的基础之上，有效助力企业数字化战略的规划和实施！

平台引入数据库沙盒执行模式，各个任务彼此独立隔绝，有力保障企业内部不同业务部门以及流程之间，在数据处理时的独立性与保密性。同时，一方面深度挖掘数据库自身强大的数据处理实力，另一方面极大幅度削减对开发及实施人员的技术门槛要求，切实助力企业的降本增效战略。

依托特有的统一数据管理模型，平台能够对各类异构数据源实施统一、高效管理，再佐以精细的账号角色权限管控体系，达成快速、高效且安全无虞的数据资源调配与管理成效。

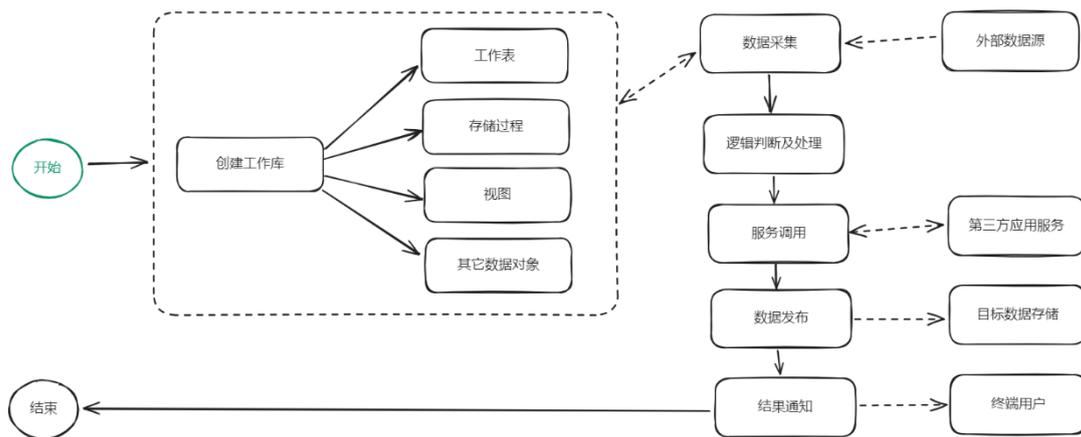
面对企业级用户，平台预先整合了大量的企业级应用与数据服务的交互能力。数据库服务：常规的关系型和非关系型数据库及各类云数据库；企业内部应用：如 SAP/BW 等主流系统；文件服务系统：涵盖 SFTP、共享目录以及 AWS S3 等云服务；即时通讯领域，企业微信、钉钉、飞书和 Teams 等常用通讯工具均进行内部整合，并成功实现各类系统之间的功能交互、数据流转与精准发布，促使企业内部信息的流通与分发效率实现质的飞跃。

系统还提供了强大的 Excel 报表处理技能，支持自定义与模板等多元模式，全面实现单元格公式、样式以及条件控制等特性。结合平台的数据集成优势，用户能够以平台所支持的各类数据源为根基，轻松构建功能强大的自动化报表，并借助邮箱或集成的即时通讯服务对外发布。

鉴于业务需求的个性化差异，平台集成 Python 开发环境，用户可以通过自定义代码段既可以精准、高效地解决企业个性化需求难题，进一步拓宽平台的应用边界，解锁更多可能场景。

## 1.1. 基本介绍

下面是仓鼠数据平台执行一个预定义任务的基本工作流程。



### 处理步骤：

- a) 任务开始后，系统会判断当前任务组是否需要工作库的支持，如果需要的话，在会自动创建一个空的工作库供后续任务使用。
- b) 如果用户定义了工作表等内容，则系统会在工作库中创建相应的表结构和数据库对象。
- c) 完成基础内容准备后，就可以进入用户定义的任务流依次运行。例如：

- d) 通过数据提取功能，从外部数据源中获取满足特定条件的数据，并同步到新创建的工作表中
- e) 按特定业务逻辑，或添加判断，对数据进行计算处理
- f) 处理过程中可能需要通过类似 RestfulAPI 接口访问外部服务，获取服务状态或服务数据来参与当前计算
- g) 计算的中间过程或结果数据可能需要发布到外部数据库，或文件系统进行存储
- h) 处理完成后，可以通过整合的第三方应用通知终端用户数据处理结果
- i) 任务执行完成，当前任务使用的工作库资源会由系统任务进行定期清理

## 1.2. 应用案例

我们以一个典型的自动化数据报表案例，来介绍如何在仓鼠平台上快速实现并交付业务。

### 案例需求

业务需求：

- 1、需要从不同的数据源（表）中获取员工信息、订单信息和业绩提成信息。
- 2、汇总计算出上个月每个业务人员的业绩销量和相应的奖金。
- 3、结果将以 Excel 报表的方式，邮件发送给部门主管审查。

背景及要求：

- 1、订单系统中的数据只有到下一个月 1 号才能完成本月数据的同步和更新。
- 2、每个人的奖金计算基准不一样，以业绩提成表里面的设定为准。销量\*奖金基数=奖金金额
- 3、该报表需要每个月 2 号上班前发给销售主管审阅
- 4、报表中仅需要包含上一个月完整的数据即可。
- 5、业绩提成信息需要使用对应月份的提成比例。Copy to clipboardErrorCopied

### 实施步骤

下面是完成上述业务需求的基本步骤：

注册使用到的数据源的连接信息

建立任务包

创建工作组

建立工作表，分别对应员工信息、订单信息和业绩提成等计算所需的数据结构

定义时间变量，每次运行时自动计算出上一个月的月份，用于查询条件

同步报表计算所需的源数据

按业务逻辑编写 SQL 脚本，计算报表结果

创建 Excel 报表，将计算结果导入报表

创建邮件发送任务，并添加报表附件

定义调度计划

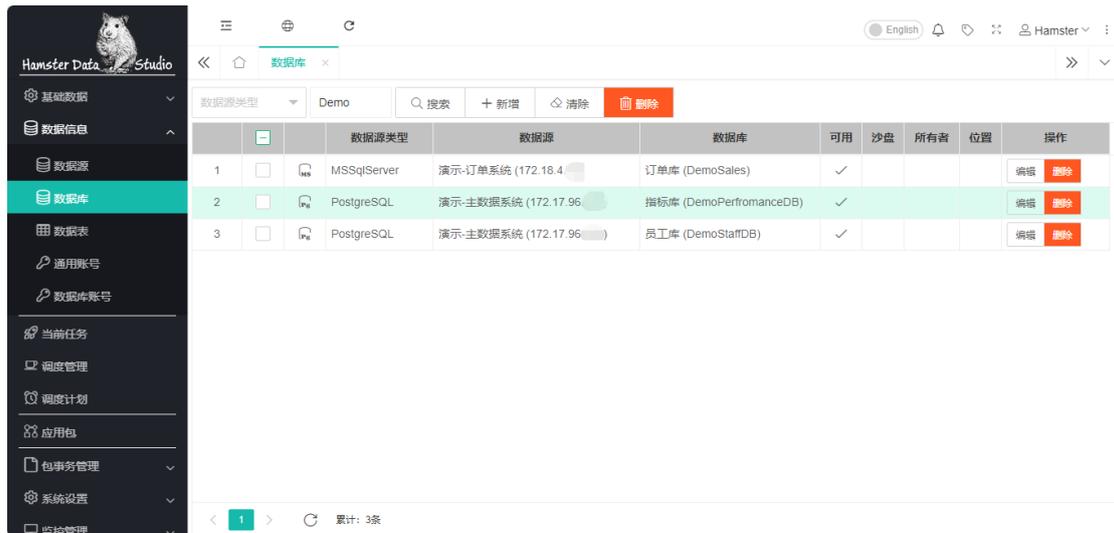
手工测试验证

激活调度计划

### 动手实施

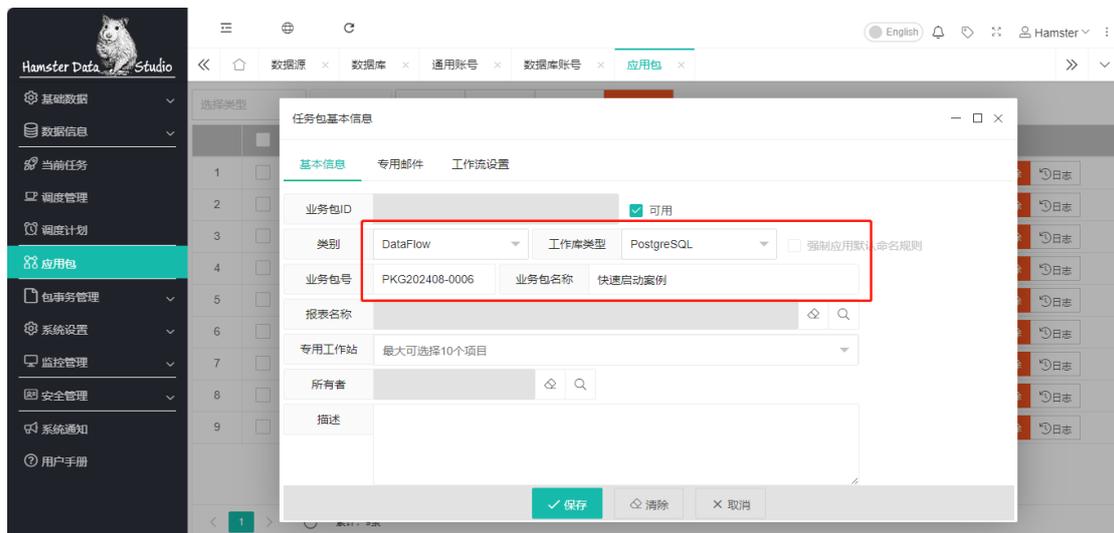
#### 注册数据源

参考 [数据源注册](#) 功能完成 3 类数据源注册，并绑定数据库访问账号，如图：



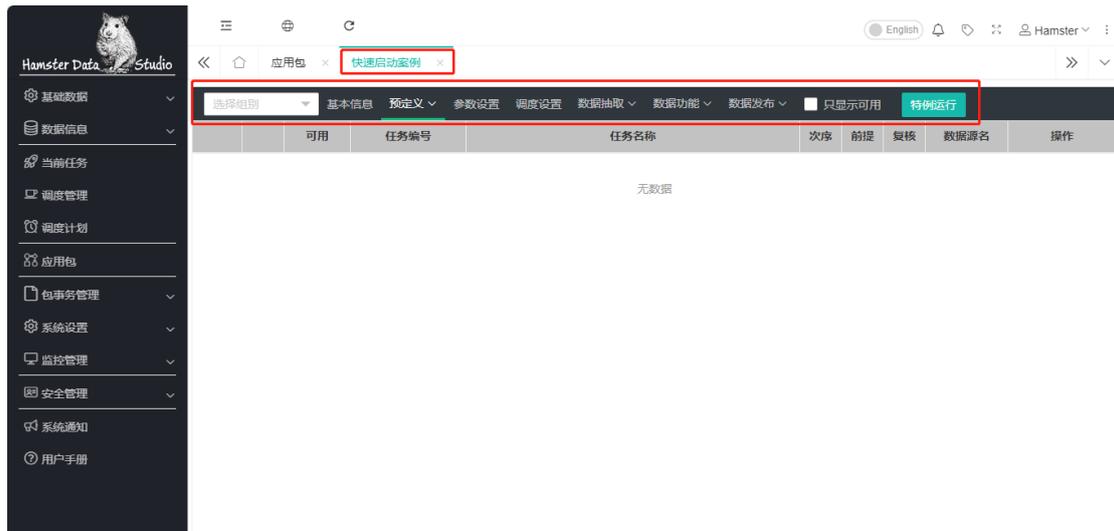
### 创建任务包

创建当前案例对应的任务包。包里面包含了所有任务处理相关的内容，后期设计操作主要都在任务包里面进。



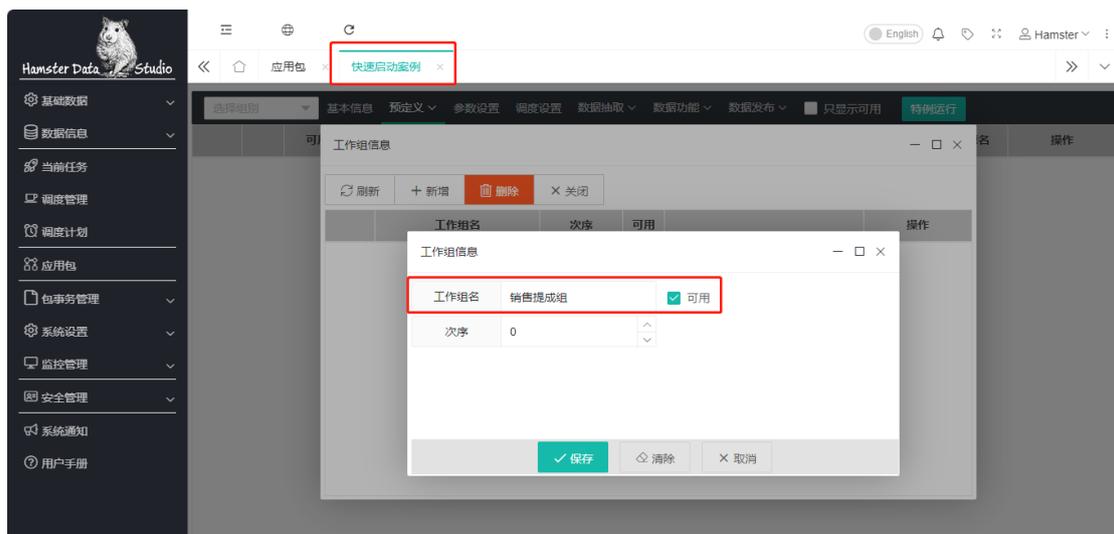
- 1) 工作库类型：即后续任务依赖的工作库将在哪种数据库服务器上执行，目前平台支持 PostgreSQL 和 MS Sql Server。选择不同类型的工作库，将决定后续 SQL 脚本的语法区别。
- 2) 业务包名称：自定义的任务包业务名称
- 3) 专用工作站：可以选择当前任务专用的工作站数据库服务器。如果不选择，则由系统动态分配。

保存完成后，即可双击刚新建的任务包记录，进入到任务包的设计页面，后续所有和包相关的任务配置都是通过设计界面中的功能菜单完成的。



### 建立工作组

每个任务包可能包含多个不同的任务，这些任务可以支持不同的业务场景。为了有序的组织这些任务功能，需要通过预定义 - 任务组创建任务组的方式类归类各个任务项目。本测试案例，我们创建一个销售提成组的任务组，后续所有任务均关联该组。



### 建立工作表

为了保存同步过来的业务数据，可以通过预定义 - 工作表功能，创建 3 张工作表，并定义相应的字段清单，分别对应：staff\_list, order\_list, performance\_setting。



各表结构依次定义为：

staff\_list：用于保存员工信息

工作表信息

表信息 字段列表 索引列表

同步主数据结构 同步外部表结构 同步查询结果结构 </> 解析文本行结构 + 添加字段  允许全空 删除字段

	<input type="checkbox"/>	字段名	数据类型	字符长度	数字长度	小数位数	可用	可空	主键	Max	自增	次序	默认值	备
1	<input type="checkbox"/>	user_id	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
2	<input type="checkbox"/>	user_name	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2		

order\_list : 用于保存销售订单信息

工作表信息

表信息 字段列表 索引列表

同步主数据结构 同步外部表结构 同步查询结果结构 </> 解析文本行结构 + 添加字段  允许全空 删除字段

	<input type="checkbox"/>	字段名	数据类型	字符长度	数字长度	小数位数	可用	可空	主键	Max	自增	次序	默认值	备
1	<input type="checkbox"/>	order_id	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
2	<input type="checkbox"/>	qty	int	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2		
3	<input type="checkbox"/>	user_id	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3		
4	<input type="checkbox"/>	offer_date	date	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4		

performance\_setting : 用于保存员工奖金设定参数

工作表信息

表信息 字段列表 索引列表

同步主数据结构 同步外部表结构 同步查询结果结构 </> 解析文本行结构 + 添加字段  允许全空 删除字段

	<input type="checkbox"/>	字段名	数据类型	字符长度	数字长度	小数位数	可用	可空	主键	Max	自增	次序	默认值	备
1	<input type="checkbox"/>	user_id	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
2	<input type="checkbox"/>	offer_date	date	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2		
3	<input type="checkbox"/>	bonus_base	numeric	0	0	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3		

report\_data : 用于保存计算出的报表数据

工作表信息

表信息 字段列表 索引列表

同步主数据结构 同步外部表结构 同步查询结果结构 </> 解析文本行结构 + 添加字段  允许全空 删除字段

	<input type="checkbox"/>	字段名	数据类型	字符长度	数字长度	小数位数	可用	可空	主键	Max	自增	次序	默认值	备
1	<input type="checkbox"/>	user_id	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
2	<input type="checkbox"/>	user_name	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2		
3	<input type="checkbox"/>	offer_month	text	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3		
4	<input type="checkbox"/>	sum_qty	int	0	0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4		
5	<input type="checkbox"/>	bonus	numeric	0	0	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5		

### 定义时间变量

因为该报表需要每个月头运行，并获取上一个月份对应的数据，所以需要定一个日期类型的用户自定义变量，运行时，系统根据设定自动计算出上一个月份的时间，从而供查询脚本使用。

通过参数设置功能创建一个新的自定义变量 LastMonth，如下图，具体参数说明，可以参考 [系统日期变量定义](#)

自定义参数信息

该参数将会以<>这样的方式应用在SQL脚本里面，注意变量名中不要包含空格。

任务组	最大可选择50个项目	<input checked="" type="checkbox"/> 可用
支持对象	支持对象类型 <input type="checkbox"/> 动态参数 <input type="checkbox"/> 数据连接参数	
参数名	LastMonth	参数类型 Date
参数值	<dt.Var_CurrDate()-1m/>	日期格式 YYYY-MM
数据库名		
次序	0	
描述		

### 创建数据同步任务

现在基础工作都已经准备好了，我们现在着手创建具体的数据处理同步任务了。

通过数据抽取 - 抽取数据库数据功能打开数据同步任务定义页面，以抽取员工主数据任务为例：

抽取数据库信息

基本信息 执行前提 前置脚本 任务脚本 字段映射 善后脚本

任务ID	4f78fe6d-e0d7-4d42-99c8-372840e69efe	次序	5	<input checked="" type="checkbox"/> 可用
任务组	最大可选择50个项目			
任务编号	FDB202408-0004	任务名称	同步 员工主数据	
数据源	演示-主数据系统 (172.17.96.173) - 员工库 (DemoStaffDB)			
目标表	staff_list			
描述				

数据源：为之前步骤中注册的含员工主数据的数据库

目标表：为在任务包里面创建的工作表，用于暂存同步过来的员工主数据



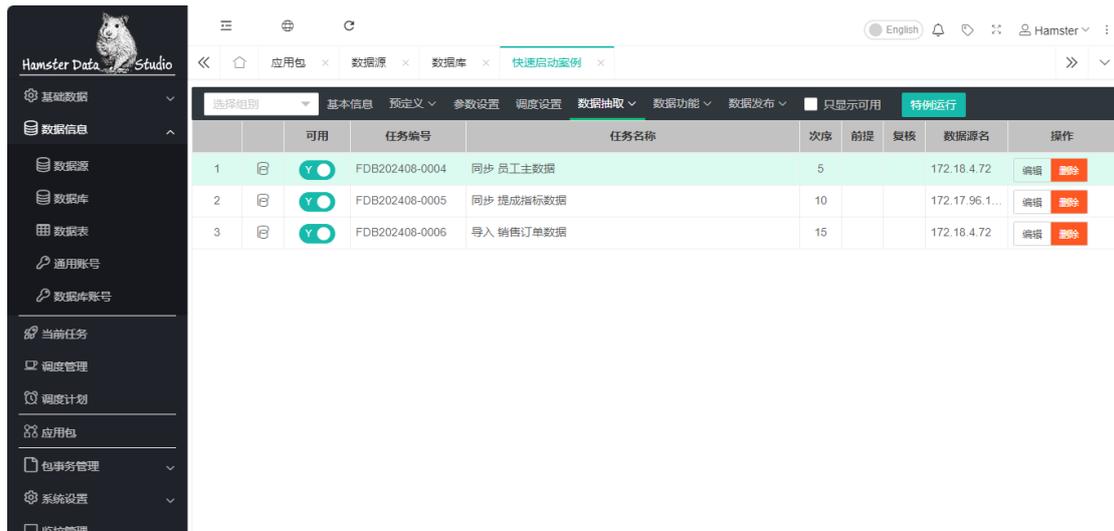
在任务脚本中填写用于从员工主数据库中查询员工数据的 SQL 查询脚本。(该 SQL 脚本语法由数据所在数据库的类型决定)，本脚本 staff\_list\_src 为数据源表，非前面创建的工作表。



在同步数据前，先清空目标表。选中该选项，用于在同步数据之前，先清空目标工作表，防止多次运行导致的垃圾数据，确保每次运行时，可以有一个干净的运行环境。

字段匹配关系：目标字段为工作表中的字段名，来源字段为员工主数据库中表的字段名，字段名可以不相同，但字段类型需要能兼容

当前完成后的任务状态如：



### 创建 SQL 脚本任务

现在所需的源数据已经同步到工作表了，我们需要根据业务逻辑需求，在工作库中编写 SQL 脚本来进行计算，并最终生成报表所需的聚合数据结果。

创建一个数据功能 - 脚本执行 脚本执行任务，并在任务中添加如下的计算脚本。注意：脚本中嵌入了自定义的时间变量<var:LastMonth/>，该变量在运行时会被自动替换成真实值。



脚本如下（本案例使用 PostgreSQL 工作库）：

```

truncate table report_data;

with _order as
(
  select user_id, sum(qty) as sum_qty
  from order_list
  where to_char(offer_date,'YYYY-MM')='<var:LastMonth/>'
  group by user_id
),

```

```

_cost as(
  select ps.user_id, to_char(ps.offer_date,'YYYY-MM') as offer_month, od.sum_qty ,ps.bonus_base *
  od.sum_qty as bonus
  from performance_setting ps inner join _order od
    on ps.user_id=od.user_id
  where to_char(ps.offer_date,'YYYY-MM')=<var:LastMonth/>
),
_usr as(
  select cst.user_id,stf.user_name,cst.offer_month, cst.sum_qty ,cst.bonus
  from _cost cst inner join staff_list stf
    on cst.user_id=stf.user_id
)
insert into report_data (user_id,user_name,offer_month,sum_qty,bonus)
  select user_id,user_name,offer_month,sum_qty,bonus from _usr;Copy to clipboardErrorCopied

```

**创建 Excel 报表任务**

数据准备好后，现在可以创建一个 Excel 报表，用于保存结果数据，并作为服务发送审阅了。使用数据功能 - 创建报表文件 功能创建一个报表文件。在任务窗体中，选择报表文件页签，更新报表文件名称和格式。名称中可以引用自定义变量来填入报表日期。



保存后，再编辑打开，点击添加数据页签功能来添加数据页签，并完善页签名称。

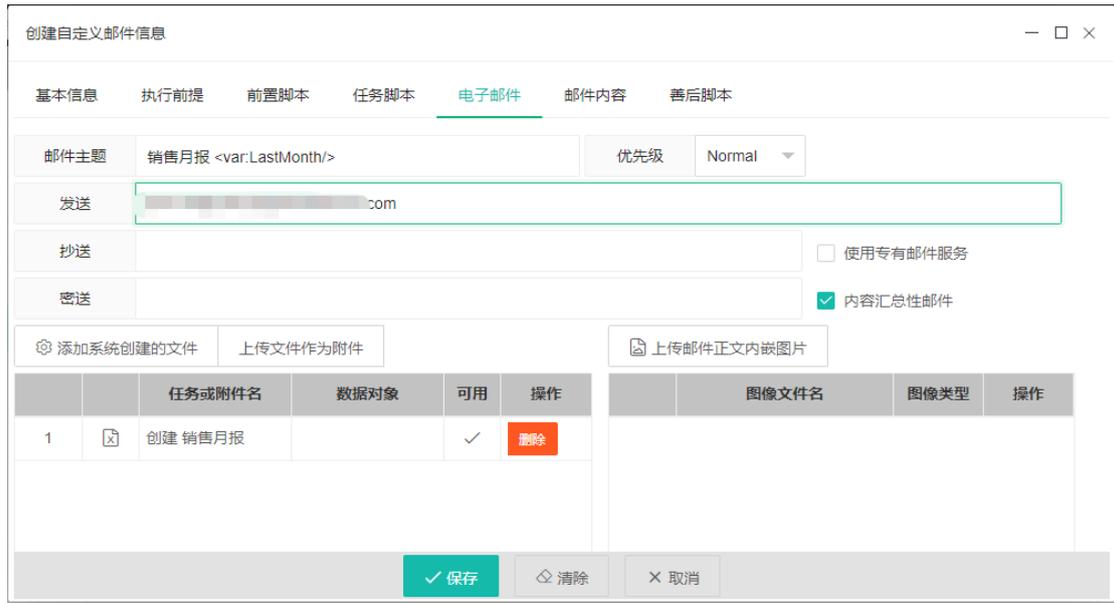


在数据查询页签中输入数据查询脚本，系统将基于该脚本在工作库中获取数据，并导入到 Excel 文件中。



### [创建邮件发送任务](#)

在数据发布 - 发送用户邮件功能中，添加一个邮件发送功能。在邮件主题和正文中都可以应用自定义变量来动态显示相应的信息。详细配置可以参考：[发送用户邮件](#)



### 定义调度计划

为了在每个月第二天凌晨运行，我们需要创建一个调度计划来执行这个任务。调度设置 里面可以创建一个新的执行计划，详细配置可以参考：[调度配置](#)

任务调度信息

调度名称	CD02 0830	任务组	销售提成组	<input checked="" type="checkbox"/> 可用 <input type="checkbox"/> 激活
调度类别	按月日	所有者		🔍
起始时间	2024-08-13 08:30:00	截至时间		
每隔	1		月	
运行日期	<input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> 8 <input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 <input type="checkbox"/> 15 <input type="checkbox"/> 16 <input type="checkbox"/> 17 <input type="checkbox"/> 18 <input type="checkbox"/> 19 <input type="checkbox"/> 20 <input type="checkbox"/> 21 <input type="checkbox"/> 22 <input type="checkbox"/> 23 <input type="checkbox"/> 24 <input type="checkbox"/> 25 <input type="checkbox"/> 26 <input type="checkbox"/> 27 <input type="checkbox"/> 28 <input type="checkbox"/> 29 <input type="checkbox"/> 30 <input type="checkbox"/> 31 <input type="checkbox"/> 最后一天			
注意：最后一天和其它日期不可以同时选择				
<input checked="" type="button" value="保存"/> <input type="button" value="清除"/> <input type="button" value="取消"/>				

### 手工测试

截至为此，我们已经完成了所有任务的设计，清单如下：

Hamster Data Studio

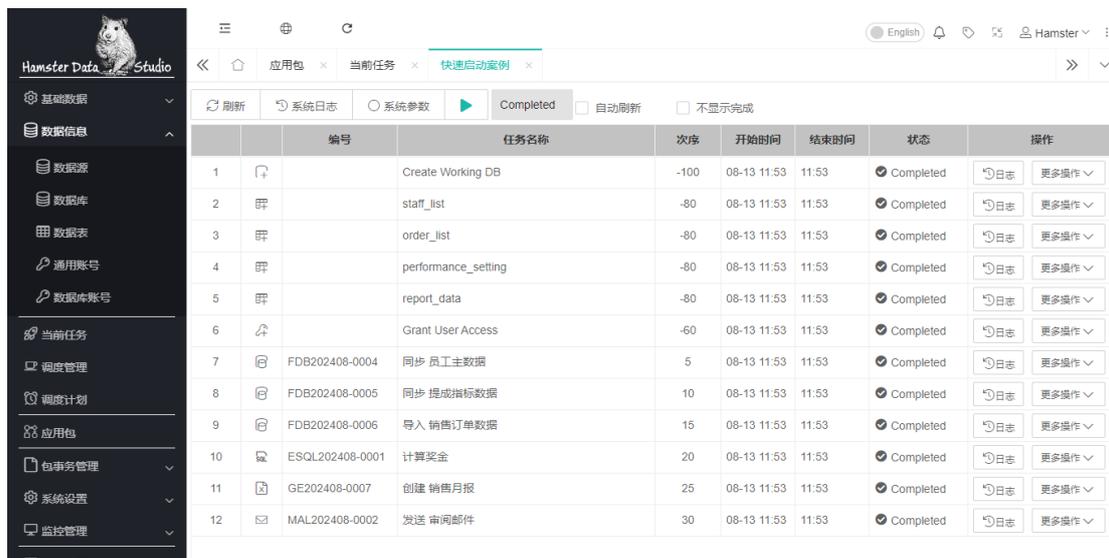
快速启动案例

可用	任务编号	任务名称	次序	前提	复核	数据来源	操作
<input checked="" type="checkbox"/>	FDB202408-0004	同步 员工主数据	5			172.17.96.1...	编辑 删除
<input checked="" type="checkbox"/>	FDB202408-0005	同步 提成指标数据	10			172.17.96.1...	编辑 删除
<input checked="" type="checkbox"/>	FDB202408-0006	导入 销售订单数据	15			172.18.4.72	编辑 删除
<input checked="" type="checkbox"/>	ESQL202408-0001	计算奖金	20				编辑 删除
<input checked="" type="checkbox"/>	GE202408-0007	创建 销售月报	25				编辑 删除
<input checked="" type="checkbox"/>	MAL202408-0002	发送 审阅邮件	30				编辑 删除

点击设计菜单上的特例运行 按钮，既可以弹出即时运行窗体，用户可以修改模拟日期，系统将基于模拟日期来计算涉及到的所有时间参数。

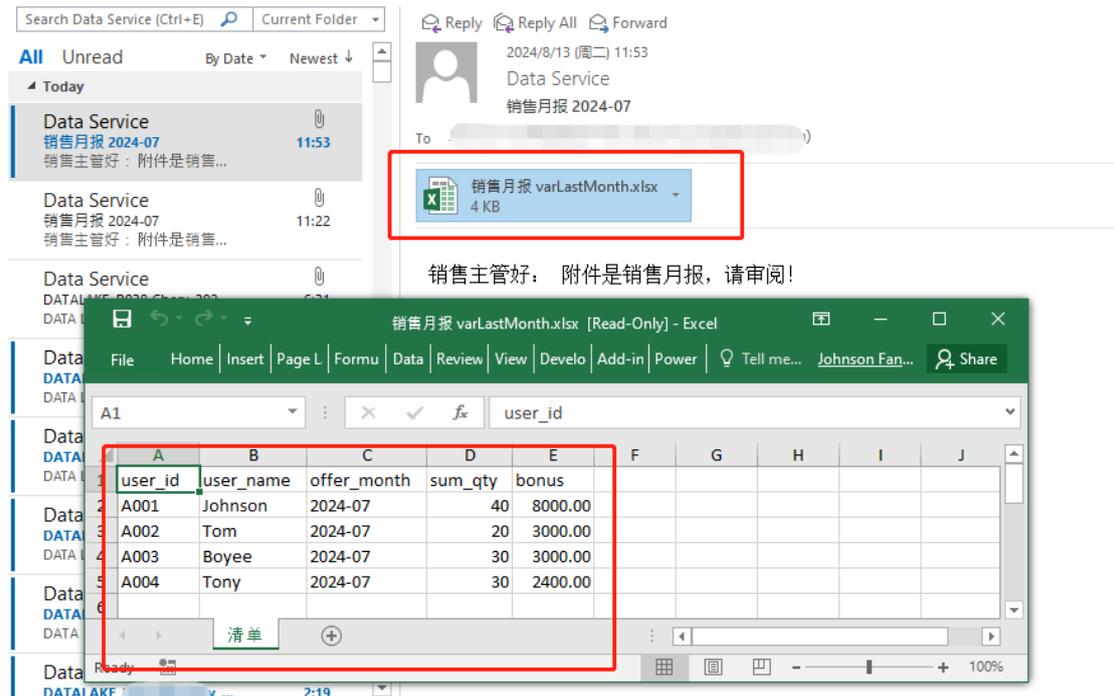


点击运行后，系统即启动一次运行任务，在左侧菜单当前任务中可以查看到刚刚发送的任务，点击编辑按钮后，可以查看当前任务运行的详细内容，等任务运行完成后，也可以收到系统推送的邮件信息。



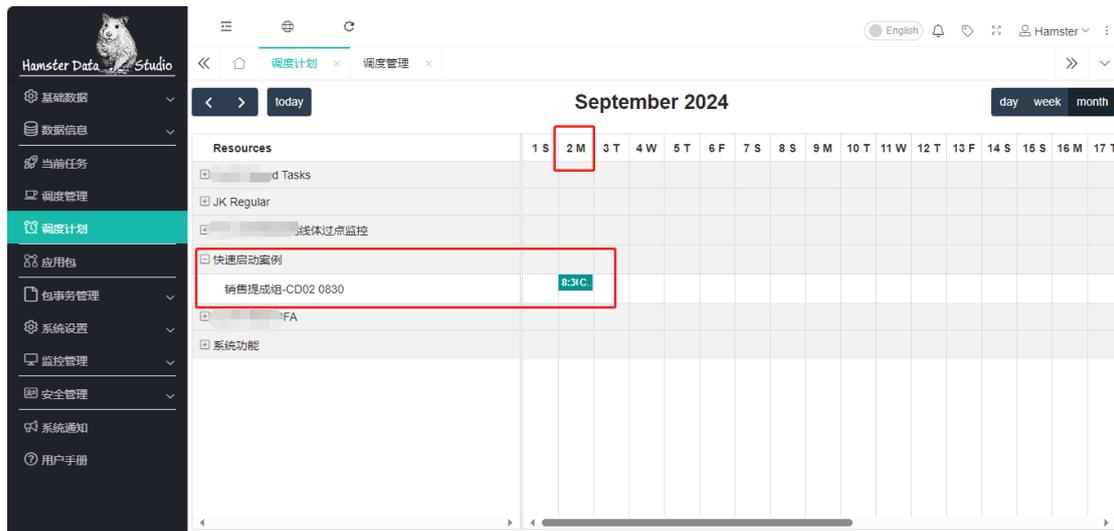
序号	任务名称	次序	开始时间	结束时间	状态	操作
1	Create Working DB	-100	08-13 11:53	11:53	Completed	日志 更多操作
2	staff_list	-80	08-13 11:53	11:53	Completed	日志 更多操作
3	order_list	-80	08-13 11:53	11:53	Completed	日志 更多操作
4	performance_setting	-80	08-13 11:53	11:53	Completed	日志 更多操作
5	report_data	-80	08-13 11:53	11:53	Completed	日志 更多操作
6	Grant User Access	-60	08-13 11:53	11:53	Completed	日志 更多操作
7	FDB202408-0004 同步 员工主数据	5	08-13 11:53	11:53	Completed	日志 更多操作
8	FDB202408-0005 同步 提成指标数据	10	08-13 11:53	11:53	Completed	日志 更多操作
9	FDB202408-0006 导入 销售订单数据	15	08-13 11:53	11:53	Completed	日志 更多操作
10	ESQL202408-0001 计算奖金	20	08-13 11:53	11:53	Completed	日志 更多操作
11	GE202408-0007 创建 销售月报	25	08-13 11:53	11:53	Completed	日志 更多操作
12	MAL202408-0002 发送 审阅邮件	30	08-13 11:53	11:53	Completed	日志 更多操作

邮件信息



### 激活调度

测试结果通过后，就可以在左侧菜单调度管理中激活该任务，并在调度计划中就可以看到该任务了。



### 体验优化

该方案仅实现了业务的基本需求，如果需要的话，还可以在此基础上做进一步的优化，提升用户使用体验，如：

可以使用 Excel 模版的模式，在报表中创建预定义好的图表，用以展示业务销售排名、奖金占比等；如果需要，也可以增加历史记录，给销售主管呈现半年来的销售趋势等。

可以完善邮件内容，将报表中的图表以图片的模式展示在邮件正文中，让销售主管第一时间就对数据有个整体的了解。

可以使用企业微信智能表单数据同步功能，将结果数据以智能表单上面的报表模式展示，从而可以实现报表移动化功能。

可以整合企业内部的即时通讯功能，如企业微信、订单、飞书或 Teams 等，即时将销售数据以文件或图表的模式，对特定人或群体进行推送，让用户可以及时了解销售状况。

### 1.3. 应用场景

仓鼠数据是一个企业级的数据处理和应用服务整合平台，可以为用户提供一个完整的数据仓库、数据自动化和应用服务整合解决方案。

我们支持但不限于以下应用场景：

- 1) 周期性的数据采集、处理及发布需求场景
- 2) 敏捷的数据验证、开发和实施需求场景
- 3) 报表自动化应用场景
- 4) 企业内部及时通讯系统整合场景，如企业微信、钉钉、飞书和 Teams 等
- 5) 批量文件处理需求
- 6) 多数据源间数据同步功能，如关系型数据库系统和文件系统间的协调等。
- 7) 第三方应用服务整合和调度等。

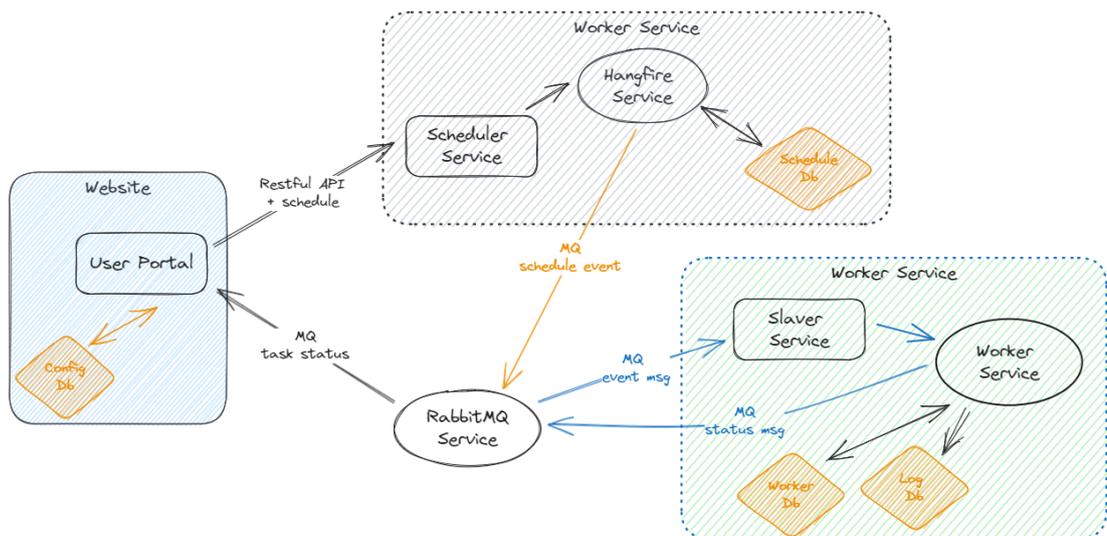
随着系统的不断优化和升级，我们将不断推出更多的业务应用场景支持功能！

## 第2章 系统部署

### 2.1. 系统架构

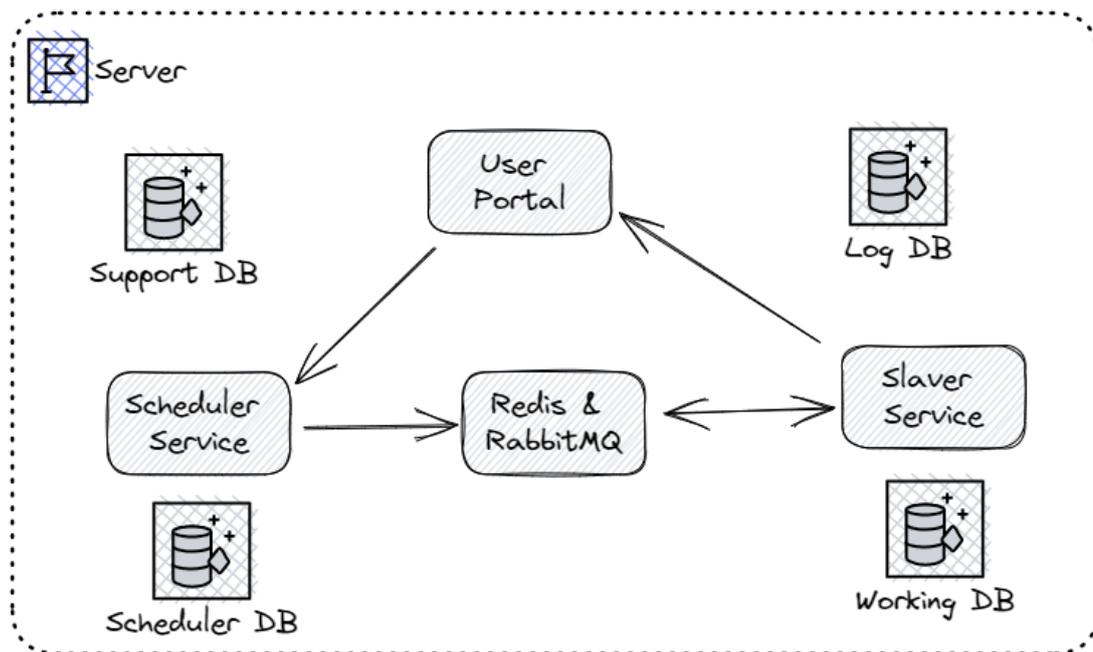
仓鼠平台采用分布式、微服务架构，各类服务可以依据业务需求进行动态拓展，从而可以实现对高并发、高性能及高可用的支持。服务架构设计

仓鼠数据平台基于消息队列，实现各类服务间的有序协调和调度，从而实现了一个高效、健壮且解耦的数据处理平台。



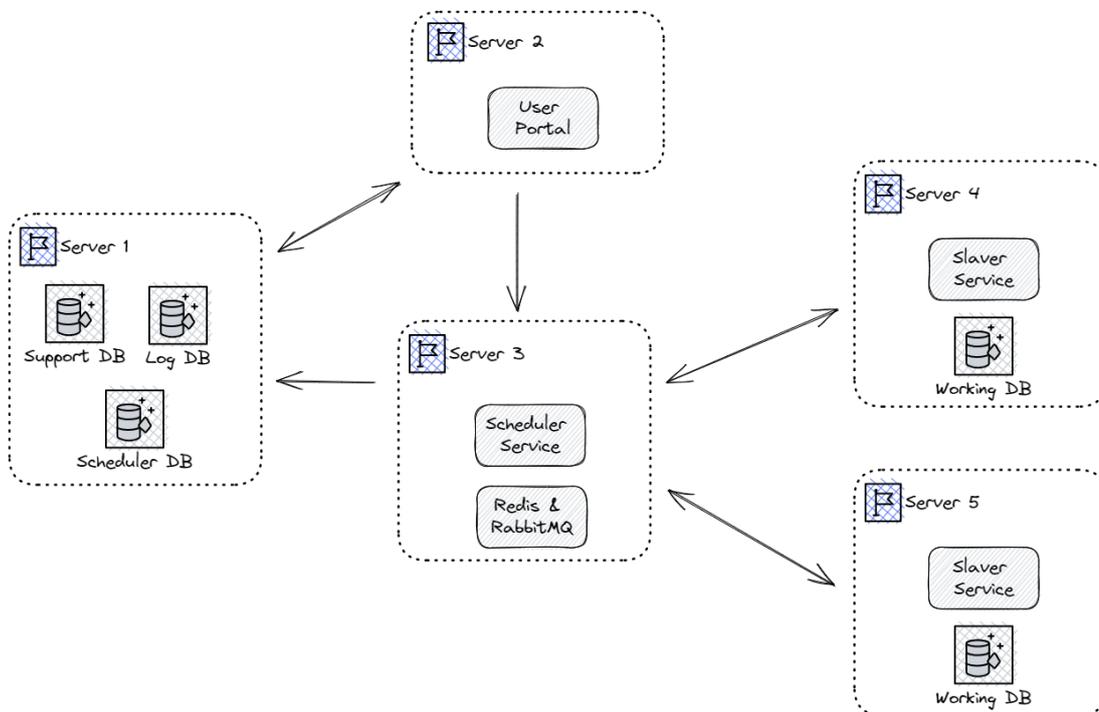
#### 单节点部署图

单节点模式适用于轻量型数据处理场景，占用资源少，并可实现快速部署上线等优点。



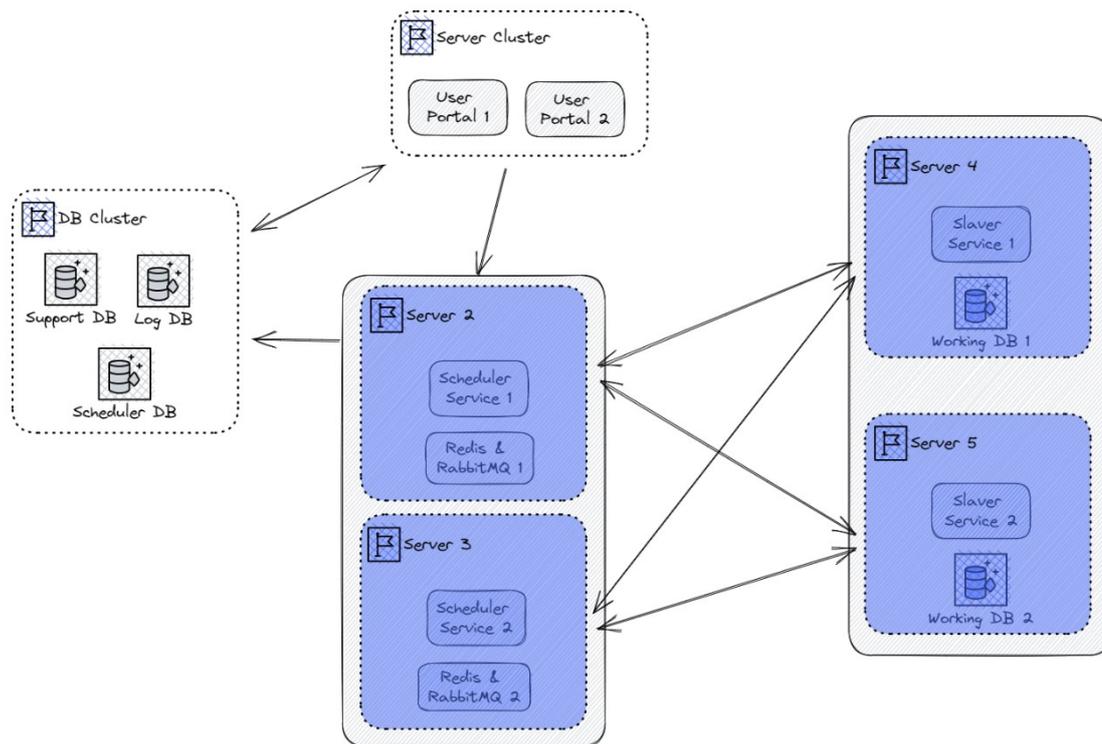
多节点部署图

多节点模式一般用于数据处理任务较频发或工作负载较重环境，用户可以部署多台工作站，如图中标识的 Server 4、Server 5 等。运行时，系统会将任务在多台机器间进行分配。（专用服务器只可以被分配给特定的任务，该服务器资源不会被其他任务分享）



高可用部署图

高可用架构主要支持对服务器运行环境要求较苛刻的环境，仓鼠系统支持针对系统后台数据库集群部署，调度服务器集群部署，多工作服务器部署等高可用组合方案。



## 2.2. 系统规划

### 部署资源规划

部署之前需要先规划好服务所使用到的服务器资源、目录结构、服务端口号以及服务账号等。

### 服务器资源

依据部署架构不同，可能使用到的服务器资源会有数量上的差异，但基本的服务器职能还是一致的，以下清单中的各类服务器都可以配置成多节点高可用模式。(IP 地址只是用来演示，具体部署时可以根据网络环境进行调整)

服务类型	服务器名称或地址	备注
用户站点服务器	172.10.10.100	
调度服务服务器	172.10.10.101	
中间件服务服务器	172.10.10.101	
任务服务服务器	172.10.10.102	可以将工作数据库服务和任务服务安装在同一台服务器中，在多工作库和任务服务节点资源的情况下，系统会优先将相关任务分配到同一台服务器上执行，以提升数据处理效率。
工作数据库服务器	172.10.10.102	

服务类型	服务器名称或地址	备注
系统数据库服务器	172.10.10.103	建议独立一台，也可以和用户站点公用

#### 目录结构

以下列表以 Windows 下面的目录结构为例，Linux 下面的结构类似。根目录可以根据服务器资源状况进行调整，但不建议放在系统盘中。

服务类型	目标目录	备注
部署根目录	D:\flyenter\hamster_data_service	
用户站点目录	D:\flyenter\hamster_data_service\Portal	用于部署用户操作界面服务，如任务包配置页面等
工作服务目录	D:\flyenter\hamster_data_service\Slaver	用于部署任务的执行服务，负责完成特定的服务，如果任务量大，可以部署多个节点。
调度服务目录	D:\flyenter\hamster_data_service\Scheduler	用于部署平台的调度服务，如果调度任务繁多，且为了保持高可用，也可以部署多个节点。
日志文件目录	D:\flyenter\hamster_data_service\Logs	用于保存由各个服务写入的日志文件。
工作文件目录	D:\flyenter\hamster_data_service\Working_folder	工作目录，支持用户操作界面的文件操作或执行服务在运行中涉及到的结果文件。一般将该目录设置成共享目录，并赋读写权限给特定的域服务账号。（平台配置中用到）

#### 服务端口号

不同的服务运行在不同的端口上，部署前需要规划好，中间件采取默认端口号。

服务类型	端口号	备注
用户站点	40020	
Hangfire 服务	40021	
调度服务	40022	
工作服务	40023	

上述配置端口将会在各个服务的配置文件中使用到。

#### 服务账号

平台运行时需要访问到对应的服务资源，如共享的工作目录，数据库服务等，所以需要为仓鼠平台申请特定的服务账号和权限来访问相应的资源。

账号类型	推荐账号名	备注
服务或域账户	hamster_svc	
数据库账号	hamster_svc	

### 数据库服务

仓鼠平台将使用到 2 类数据库服务，即平台自身的系统数据库和为支持特定任务而动态创建的工作库。这 2 类数据库可以公用一台数据库服务，但建议将系统库和工作库分别部署在 2 台不同的数据库实例上。因为工作库的工作负载较大，同时，仓鼠平台需要拥有工作库 sa/pgadmin 权限。

数据库类型	数据库服务器	数据库名称	备注
系统配置库	172.10.10.100	Hamster_2023_Support	用于保存数据源、任务设计及平台配置等信息
系统调度库	172.10.10.100	Hamster_2023_HF	用于保存调度计划
系统日志库	172.10.10.100	Hamster_2023_Log	用于保存任务执行明细日志
工作库 1	172.10.10.101	*	工作库服务器中的数据库会由平台动态创建，根据任务负载，选择单节点或多节点

## 2.3. Windows 部署

基于 Windows 平台部署时，需要服务器环境具备以下要求：

#### 环境要求：

Windows Server 2016 及以上

IIS Service

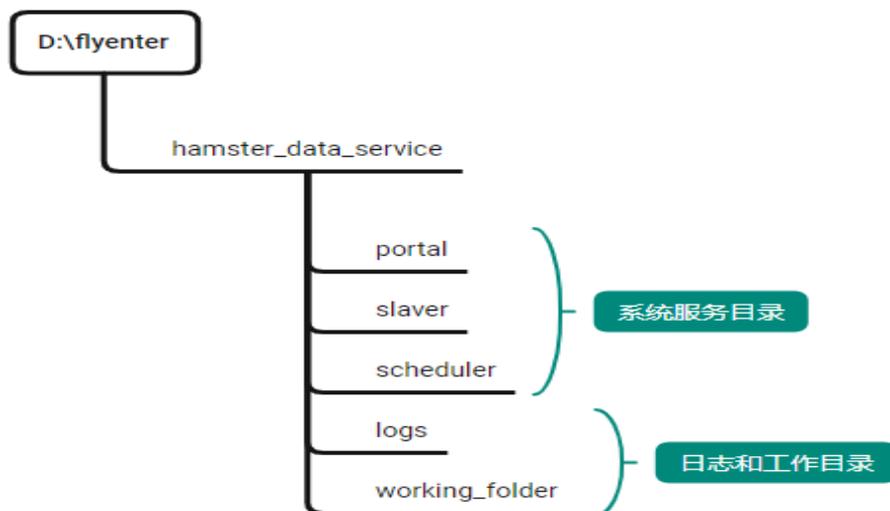
PostgreSQL 15 及以上

安装 .Net 8 Runtime

#### 部署步骤：

##### [创建服务目录结构](#)

基于规划中预定义的服务目录结构，创建类似如下的目录结构。



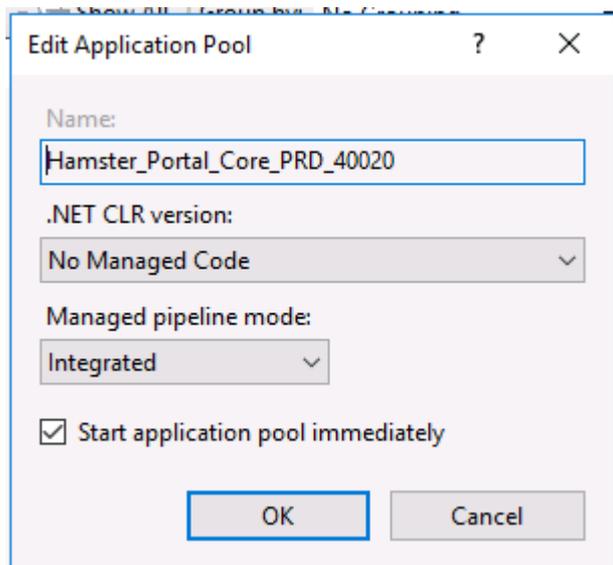
目录结构创建完成后，将相应的系统服务文件解压并复制到对应的服务根目录下。

#### [部署 IIS Portal 站点-User Portal](#)

在 IIS 中创建类似如下的站点目录，并将站点目录指定上一步创建的 portal 目录。



并将对应的 Application Pool 的属性调整如下图：



#### [部署调度服务-Scheduler Service](#)

登录预订的调度服务服务器，创建 Windows 服务注册脚本。

```
cd d:\flyenter\hamster_data_service\Copy to clipboardErrorCopied
```

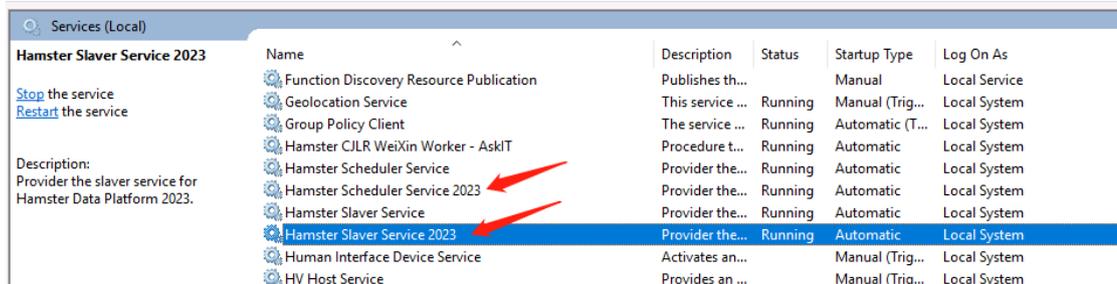
即所有服务的根目录中，创建 Create\_Scheduler\_Service.bat 文件，文件内容中的路径对应当前真实部署的调度服务目录：

```
sc create flyenter.hamster.scheduler2024 binpath=
"D:\flyenter\hamster_data_service\scheduler\FlyEnter.Hamster.Scheduler.exe"
start=auto displayname="Hamster Scheduler Service 2024"
sc description flyenter.hamster.scheduler2024 "Provider the scheduler service for
```

Hamster Data Platform 2024.”

```
sc start flyenter.hamster.scheduler2024Copy to clipboardErrorCopied
```

上述脚本利用 Windows 服务注册功能实现调度服务的注册，并设置为自启动模式。文件创建完成后，右击并选择以管理员模式运行，完成后，可以从 Windows 的服务模块中查询到注册结果，如下：



同时，创建服务卸载脚本 Remove\_Schedulerr\_Service.bat，内容如下：

```
sc stop flyenter.hamster.scheduler2024
```

```
sc delete flyenter.hamster.scheduler2024Copy to clipboardErrorCopied
```

### 部署工作服务-Slaver Service

登录预订的工作服务服务器，创建 Windows 服务注册脚本。

```
cd d:\flyenter\hamster_data_service\Copy to clipboardErrorCopied
```

即所有服务的根目录中，创建 Create\_Slaver\_Service.bat 文件，文件内容中的路径对应当前真实部署的工作服务目录：

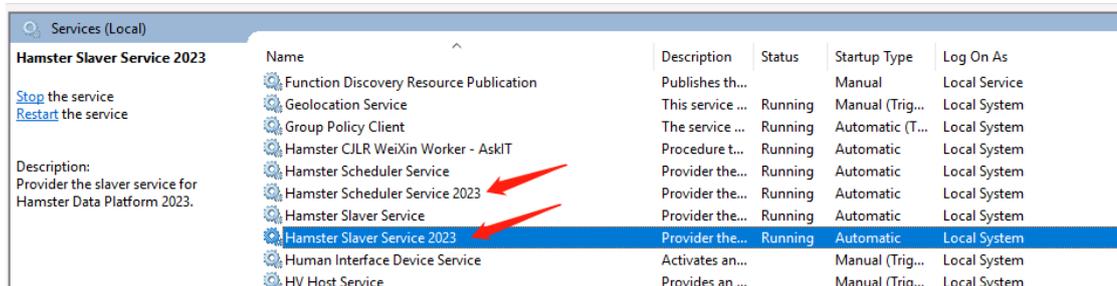
```
sc create flyenter.hamster.slaver2024 binpath=
```

```
"D:\flyenter\hamster_data_service\slaver\FlyEnter.Hamster.Slaver.exe" start=auto displayname="Hamster Slaver Service 2024"
```

```
sc description flyenter.hamster.slaver2024 "Provider the slaver service for Hamster Data Platform 2024."
```

```
sc start flyenter.hamster.slaver2024Copy to clipboardErrorCopied
```

上述脚本利用 Windows 服务注册功能实现工作服务的注册，并设置为自启动模式。文件创建完成后，右击并选择以管理员模式运行，完成后，可以从 Windows 的服务模块中查询到注册结果，如下：



同时，创建服务卸载脚本 Remove\_Slaver\_Service.bat，内容如下：

```
sc stop flyenter.hamster.slaver2024
```

```
sc delete flyenter.hamster.slaver2024
```

## 2.4. Linux 部署

本地安装基于 AWS Linux2 版本进行安装

[系统更新及资源下载](#)

[配置安全组](#)

添加如下端口：

22：远程客户端访问 EC2

15672：RabbitMQ 的网页端访问

42000：仓鼠站点端口

42010：仓鼠调度接口端口

[更新系统环境](#)

**更新新建的 EC2 系统环境，**

```
[ec2-user@ip-172-31-29-0 ~]$ sudo su
[root@ip-172-31-29-0 tools]# yum update
[root@ip-172-31-29-0 tools]# yum install ca-certificates
[root@ip-172-31-29-0 tools]# update-ca-trust
# 调整时区
[root@ip-172-31-29-0 ec2-user]# timedatectl set-timezone "Asia/Shanghai"
[root@ip-172-31-29-0 ec2-user]# timedatectl status
    Local time: Tue 2024-12-24 13:58:50 CST
    Universal time: Tue 2024-12-24 05:58:50 UTC
    RTC time: Tue 2024-12-24 05:58:51
    Time zone: Asia/Shanghai (CST, +0800)
    NTP enabled: yes
NTP synchronized: yes
    RTC in local TZ: no
    DST active: n/a
[root@ip-172-31-29-0 ec2-user]#
Copy to clipboardErrorCopied
下载中间件安装包
[root@ip-172-31-29-0 tools]# mkdir /hamster
[root@ip-172-31-29-0 tools]# mkdir -p /hamster/tools
[root@ip-172-31-29-0 tools]# chmod 777 /hamster/tools/
[root@ip-172-31-29-0 tools]# cd /hamster/tools/
[root@ip-172-31-29-0 tools]# wget
https://download.visualstudio.microsoft.com/download/pr/4e3b04aa-c015-4e06-a42e-
05f9f3c54ed2/74d1bb68e330eeal3ecfc47f7cf9aeb7/dotnet-sdk-8.0.404-linux-x64.tar.gz
[root@ip-172-31-29-0 tools]# wget https://github.com/erlang/otp/releases/download/OTP-
27.2/otp_src_27.2.tar.gz
[root@ip-172-31-29-0 tools]# wget https://github.com/rabbitmq/rabbitmq-
server/releases/download/v4.0.5/rabbitmq-server-generic-unix-4.0.5.tar.gz
[root@ip-172-31-29-0 tools]# wget
https://ftp.postgresql.org/pub/source/v17.2/postgresql-17.2.tar.gz
```

Copy to clipboardErrorCopied

### [安装 Erlang/OTP 27](#)

#### [安装系统](#)

```
[root@ip-172-31-29-0 tools]# yum -y install gcc glibc-devel make ncurses-devel openssl-  
devel xmlto perl wget gtk2-devel binutils-devel  
[root@ip-172-31-29-0 tools]# tar -zxf otp_src_27.2.tar.gz  
[root@ip-172-31-29-0 tools]# cd otp_src_27.2/  
[root@ip-172-31-29-0 otp_src_27.2]# ./configure --prefix=/hamster/erlang27  
[root@ip-172-31-29-0 otp_src_27.2]# make && make install
```

Copy to clipboardErrorCopied

#### [配置系统](#)

```
[root@ip-172-31-29-0 otp_src_27.2]# vi /etc/profile
```

```
export ERLPATH=/hamster/erlang27
```

```
export PATH=$PATH:$ERLPATH/bin
```

```
export USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

```
[root@ip-172-31-29-0 otp_src_27.2]# source /etc/profile
```

```
[root@ip-172-31-29-0 otp_src_27.2]# vi /root/.bashrc
```

```
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
```

```
    . /etc/bashrc
```

```
fi
```

```
# load /etc/profile auto
```

```
if [ -f /etc/profile ]; then
```

```
    . /etc/profile
```

```
fi
```

```
# 验证安装是否成功
```

```
[root@ip-172-31-29-0 otp_src_27.2]# erl
```

```
Erlang/OTP 27 [erts-15.2] [source] [64-bit] [smp:2:2] [ds:2:2:10] [async-threads:1]
```

```
Eshell V15.2 (press Ctrl+G to abort, type help(). for help)
```

```
1> halt().
```

```
# 清理安装的中间过程目录
```

```
[root@ip-172-31-29-0 otp_src_27.2]# rm -rf /hamster/tools/otp_src_27.2
```

Copy to clipboardErrorCopied

### [安装 RabbitMQ4](#)

#### [安装系统](#)

```
[root@ip-172-31-29-0 tools]# mkdir /hamster/rabbitmq4
```

```
[root@ip-172-31-29-0 tools]# tar -xvf rabbitmq-server-generic-unix-4.0.5.tar.xz -C
```

```
/hamster/rabbitmq4 --strip-components=1  
[root@ip-172-31-29-0 tools]# vi /etc/profile
```

```
export ERLPATH=/hamster/erlang27  
export RABPATH=/hamster/rabbitmq4  
export PATH=$PATH:$ERLPATH/bin:$RABPATH/sbin
```

```
export USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

```
[root@ip-172-31-29-0 tools]# source /etc/profileCopy to clipboardErrorCopied
```

### [配置服务](#)

```
[root@ip-172-31-29-0 tools]# cd /hamster/rabbitmq4/sbin/  
[root@ip-172-31-29-0 sbin]# ./rabbitmq-plugins enable rabbitmq_management  
Enabling plugins on node rabbit@ip-172-31-29-0:
```

```
rabbitmq_management
```

The following plugins have been configured:

```
rabbitmq_management  
rabbitmq_management_agent  
rabbitmq_web_dispatch
```

Applying plugin configuration to rabbit@ip-172-31-29-0...

The following plugins have been enabled:

```
rabbitmq_management  
rabbitmq_management_agent  
rabbitmq_web_dispatch
```

```
[root@ip-172-31-29-0 sbin]# cd /hamster/rabbitmq4/etc/rabbitmq/
```

```
[root@ip-172-31-29-0 rabbitmq]# ls
```

```
enabled_plugins
```

# 创建 rabbitmq.conf 配置文件，并下载模板内容后再修改

```
[root@ip-172-31-29-0 rabbitmq]# vi rabbitmq.conf
```

# 创建 rabbitmq.conf 文件，默认是没有的，修改，将超时时间设置为 4 小时

```
consumer_timeout = 14400000
```

# 运行 guest 用户从任何机器上登录，默认只能 localhost 登录，修改配置可以从任何机器上访问站点。

```
loopback_users.guest = false
```

# 额外添加配置：max message size

```
max_message_size = 10485760
```

Copy to clipboardErrorCopied

### [配置自启动服务](#)

#### [rabbitmq.conf 模板](#)

```
[root@ip-172-31-29-0 rabbitmq]# vi /etc/systemd/system/rabbitmq.service
```

```
[Unit]
```

```
Description=RabbitMQ Service
```

```
After=network.target

[Service]
Type=simple
User=root
Group=root
Environment=PATH=$PATH:/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/hamster/erlang27/
bin:/hamster/rabbitmq4/sbin
ExecStart=/hamster/rabbitmq4/sbin/rabbitmq-server
ExecStop=/hamster/rabbitmq4/sbin/rabbitmqctl stop
restart=always
```

```
[Install]
WantedBy=multi-user.target
```

```
[root@ip-172-31-29-0 rabbitmq]# systemctl daemon-reload
[root@ip-172-31-29-0 rabbitmq]# systemctl enable rabbitmq.service
Created symlink from /etc/systemd/system/multi-user.target.wants/rabbitmq.service to
/etc/systemd/system/rabbitmq.service.
[root@ip-172-31-29-0 rabbitmq]# systemctl start rabbitmq.service
[root@ip-172-31-29-0 rabbitmq]# systemctl status rabbitmq.service
● rabbitmq.service - RabbitMQ Service
   Loaded: loaded (/etc/systemd/system/rabbitmq.service; enabled; vendor preset:
disabled)
   Active: active (running) since Tue 2024-12-24 14:31:14 CST; 7s ago
 Main PID: 13817 (rabbitmq-server)
   CGroup: /system.slice/rabbitmq.service
           └─13817 /bin/sh /hamster/rabbitmq4/sbin/rabbitmq-server
           └─13836 /hamster/erlang27/lib/erlang/erts-15.2/bin/beam.smp -W w -MBas
ageffcbf -MHas ageffcbf -MBlmbcs 512 -MHLmbcs 512 -MMmcs 30 -pc u...
           └─13877 erl_child_setup 65535
           └─14054 sh -s disksup
           └─14057 /hamster/erlang27/lib/erlang/lib/os_mon-2.10.1/priv/bin/memsup
           └─14059 /hamster/erlang27/lib/erlang/lib/os_mon-2.10.1/priv/bin/cpu_sup
           └─16037 /hamster/erlang27/lib/erlang/erts-15.2/bin/inet_gethost 4
           └─16041 /hamster/erlang27/lib/erlang/erts-15.2/bin/inet_gethost 4
           └─17480 /bin/sh -s rabbit_disk_monitorCopy to clipboardErrorCopied
```

### 创建用户

```
[root@ip-172-31-29-0 rabbitmq]# rabbitmqctl add_user hamster hamster123
Adding user "hamster" ...
Done. Dont forget to grant the user permissions to some virtual hosts! See rabbitmqctl
help set_permissions to learn more.
[root@ip-172-31-29-0 rabbitmq]# rabbitmqctl set_user_tags hamster administrator
Setting tags for user "hamster" to [administrator] ...
```

```
[root@ip-172-31-29-0 rabbitmq]# rabbitmqctl set_permissions -p / hamster ".*" ".*" ".*"
Setting permissions for user "hamster" in vhost "/" ...
[root@ip-172-31-29-0 rabbitmq]# rabbitmqctl list_users
Listing users ...
user      tags
guest     [administrator]
hamster   [administrator]
Copy to clipboardErrorCopied
```

### [客户端访问](#)

在启动服务并添加安全组端口后，既可以访问 RabbitMQ 的 Web 端界面，地址一般类似为：

<http://ec2-xxx.cn-north-1.compute.amazonaws.com.cn:15672>

### [安装 PostgreSQL17](#)

#### [安装编译环境](#)

```
[root@ip-172-31-29-0 rabbitmq]# cd /hamster/tools/
[root@ip-172-31-29-0 tools]# yum install -y perl-ExtUtils-Embed readline-devel zlib-
devel pam-devel libxml2-devel libxslt-devel openldap-devel python-devel gcc-c++
openssl-devel cmake bison
[root@ip-172-31-29-0 tools]# yum install -y libicu-devel --skip-broken
[root@ip-172-31-29-0 tools]# yum install -y libuuid-develCopy to clipboardErrorCopied
```

#### [安装系统](#)

```
[root@ip-172-31-29-0 tools]# tar -zxvf postgresql-17.2.tar.gz
[root@ip-172-31-29-0 tools]# cd postgresql-17.2/
[root@ip-172-31-29-0 postgresql-17.2]# ./configure --prefix=/hamster/postgresql17 --
with-uuid=e2fs
[root@ip-172-31-29-0 postgresql-17.2]# make && make install
[root@ip-172-31-29-0 postgresql-17.2]# cd contrib/uuid-oss/
[root@ip-172-31-29-0 uuid-oss]# make && make installCopy to clipboardErrorCopied
```

#### [创建操作账号](#)

```
[root@ip-172-31-29-0 uuid-oss]# groupadd postgres
[root@ip-172-31-29-0 uuid-oss]# useradd -g postgres postgres
[root@ip-172-31-29-0 uuid-oss]# passwd postgres
Changing password for user postgres.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-29-0 uuid-oss]# id postgres
uid=1001(postgres) gid=1001(postgres) groups=1001(postgres)Copy to clipboardErrorCopied
```

#### [创建数据目录并配置环境](#)

```
[root@ip-172-31-29-0 uuid-oss]# mkdir -p /hamster/postgresql17/data
[root@ip-172-31-29-0 uuid-oss]# chown -R postgres:postgres /hamster/postgresql17/data/
[root@ip-172-31-29-0 uuid-oss]# chmod -R 700 /hamster/postgresql17/data/
[root@ip-172-31-29-0 uuid-oss]# vi /home/postgres/.bashrc
# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=
```

```
export PGHOME=/hamster/postgresql17
export PGDATA=/hamster/postgresql17/data

export PATH=$PATH:/$PGHOME/bin
[root@ip-172-31-29-0 uuid-oss]# source /home/postgres/.bashrc
[root@ip-172-31-29-0 uuid-oss]# su postgres
[postgres@ip-172-31-29-0 uuid-oss]$ initdb
[postgres@ip-172-31-29-0 uuid-oss]$ cd /hamster/postgresql17/data/
[postgres@ip-172-31-29-0 data]# vi postgresql.conf
# 修改配置文件，修改为*，允许其他机器可以连接改数据库
listen_addresses = '*'                # what IP address(es) to listen on;
                                       # comma-separated list of addresses;
                                       # defaults to 'localhost'; use '*' for all
                                       # (change requires restart)
#port = 5432                          # (change requires restart)
max_connections=200                   # 调整数据库最大用户连接数，可以适当调整

[postgres@ip-172-31-29-0 data]$ vi pg_hba.conf
# IPv4 local connections:
### 下面可选修改，配置后所有机器均可以无密码访问数据库（仅仅在安全的环境下可用，不建议
修改）
# host    all             all             0.0.0.0/0          trust
# 下面配置将限制仅 IP 地址为 239 的服务器可远程连接数据库，且需要密码访问
host     all             all             192.168.13.239/32 md5
# 本地也需要使用密码访问。如果设置为：trust，则可以无密码登陆
host     all             all             127.0.0.1/32      md5
# 启动数据库服务测试
[postgres@ip-172-31-29-0 data]$ pg_ctl -D /hamster/postgresql17/data -l
/hamster/postgresql17/data/logfile start
waiting for server to start.... done
server started
# 登陆数据库服务，修改 postgres 登陆密码为：abc123456
[postgres@ip-172-31-29-0 data]$ psql
psql (17.2)
Type "help" for help.

postgres=# alter user postgres with password 'abc123456';
ALTER ROLE
postgres=# \q
[postgres@ip-172-31-29-0 data]$ pg_ctl stop
waiting for server to shut down.... done
server stoppedCopy to clipboardErrorCopied
配置自启动服务
[postgres@ip-172-31-29-0 data]$ exit
```

```
exit
[root@ip-172-31-29-0 ec2-user]# vi /etc/systemd/system/postgresql.service
[Unit]
Description=PostgreSQL Database Server
After=network.target

[Service]
Type=forking
User=postgres
Group=postgres
ExecStart=/hamster/postgresql17/bin/pg_ctl start -D /hamster/postgresql17/data -l
/hamster/postgresql17/data/logfile
ExecStop=/hamster/postgresql17/bin/pg_ctl stop -D /hamster/postgresql17/data
ExecReload=/hamster/postgresql17/bin/pg_ctl reload -D /hamster/postgresql17/data

[Install]
WantedBy=multi-user.target
[root@ip-172-31-29-0 ec2-user]#
[root@ip-172-31-29-0 data]# systemctl daemon-reload
[root@ip-172-31-29-0 data]# systemctl enable postgresql.service
Created symlink from /etc/systemd/system/multi-user.target.wants/postgresql.service to
/etc/systemd/system/postgresql.service.
[root@ip-172-31-29-0 data]# systemctl start postgresql.service
[root@ip-172-31-29-0 data]# systemctl status postgresql.service
● postgresql.service - PostgreSQL Database Server
   Loaded: loaded (/etc/systemd/system/postgresql.service; enabled; vendor preset:
disabled)
   Active: active (running) since Tue 2024-12-24 16:04:48 CST; 8s ago
     Process: 24885 ExecStart=/hamster/postgresql16/bin/pg_ctl start -D
/hamster/postgresql16/data -l /hamster/postgresql16/data/logfile (code=exited,
status=0/SUCCESS)
    Main PID: 24887 (postgres)
      CGroup: /system.slice/postgresql.service
              └─24887 /hamster/postgresql16/bin/postgres -D /hamster/postgresql16/data
              └─24888 postgres: checkpointer
              └─24889 postgres: background writer
              └─24891 postgres: walwriter
              └─24892 postgres: autovacuum launcher
              └─24893 postgres: logical replication launcher
Copy to
clipboardErrorCopied
安装.Net8 SDK
[root@ip-172-31-29-0 data]# cd /hamster/tools/
[root@ip-172-31-29-0 tools]# mkdir -p /hamster/dotnet8
[root@ip-172-31-29-0 tools]# tar -zxf dotnet-sdk-8.0.404-linux-x64.tar.gz -C
```

```
/hamster/dotnet8/  
[root@ip-172-31-29-0 tools]# vi /etc/profile  
  
export DOTNET_ROOT=/hamster/dotnet8  
export ERLPATH=/hamster/erlang27  
export RABPATH=/hamster/rabbitmq4  
export PATH=$PATH:$ERLPATH/bin:$RABPATH/sbin:$DOTNET_ROOT
```

```
export USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL  
[root@ip-172-31-29-0 tools]# source /etc/profile  
[root@ip-172-31-29-0 tools]# dotnet --version  
8.0.404Copy to clipboardErrorCopied
```

### 部署仓鼠数据服务

#### Hamster Portal

```
[root@ip-172-31-29-0 data]# cd /hamster/tools/  
[root@ip-172-31-29-0 tools]# unzip hamster_portal_linux_2024.zip -d /hamster/hamster-portal  
[root@ip-172-31-29-0 tools]# chmod +X /hamster/hamster-portal/FlyEnter.Hamster.Portal.dll  
[root@ip-172-31-29-0 tools]# vi /etc/systemd/system/hamster-portal.service  
[Unit]  
Description=Hamster Portal Service  
After=network.target rabbitmq.service postgresql.service  
Requires=rabbitmq.service postgresql.service  
  
[Service]  
Type=simple  
Environment=DOTNET_ROOT=/hamster/dotnet8  
Environment=ASPNETCORE_ENVIRONMENT=Linux  
Environment=PATH=$PATH:/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/hamster/dotnet8  
ExecStart=/hamster/dotnet8/dotnet /hamster/hamster-portal/FlyEnter.Hamster.Portal.dll  
WorkingDirectory=/hamster/hamster-portal/  
Restart=always  
RestartSec=10  
KillSignal=SIGINT  
SyslogIdentifier=Hamster Portal  
User=root  
Group=root  
  
[Install]  
WantedBy=multi-user.target  
[root@ip-172-31-29-0 tools]# systemctl daemon-reload  
[root@ip-172-31-29-0 tools]# systemctl enable hamster-portal.service  
Created symlink from /etc/systemd/system/multi-user.target.wants/hamster-portal.service
```

```
to /etc/systemd/system/hamster-portal.service.  
[root@ip-172-31-29-0 tools]# systemctl start hamster-portal.service  
[root@ip-172-31-29-0 tools]# systemctl status hamster-portal.service  
● hamster-portal.service - Hamster Portal Service  
   Loaded: loaded (/etc/systemd/system/hamster-portal.service; enabled; vendor preset:  
disabled)  
   Active: active (running) since Tue 2024-12-24 16:35:47 CST; 8s ago  
   Main PID: 3479 (dotnet)  
   CGroup: /system.slice/hamster-portal.service  
           └─3479 /hamster/dotnet8/dotnet /hamster/hamster-  
portal/FlyEnter.Hamster.Portal.dll
```

Copy to clipboardErrorCopied

### Hamster Scheduler

```
[root@ip-172-31-29-0 tools]# unzip hamster_scheduler_linux_2024.zip -d  
/hamster/hamster-scheduler  
[root@ip-172-31-29-0 tools]# chmod +X /hamster/hamster-  
scheduler/FlyEnter.Hamster.Scheduler.dll  
[root@ip-172-31-29-0 tools]# vi /etc/systemd/system/hamster-scheduler.service  
[Unit]  
Description=Hamster Scheduler Service  
After=network.target rabbitmq.service postgresql.service  
Requires=rabbitmq.service postgresql.service  
  
[Service]  
Type=simple  
Environment=DOTNET_ROOT=/hamster/dotnet8  
Environment=ASPNETCORE_ENVIRONMENT=Linux  
Environment=PATH=$PATH:/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/hamster/dotnet8  
ExecStart=/hamster/dotnet8/dotnet /hamster/hamster-  
scheduler/FlyEnter.Hamster.Scheduler.dll  
WorkingDirectory=/hamster/hamster-scheduler/  
Restart=always  
RestartSec=10  
KillSignal=SIGINT  
SyslogIdentifier=Hamster Scheduler 2024  
User=root  
Group=root  
  
[Install]  
WantedBy=multi-user.target  
[root@ip-172-31-29-0 tools]# systemctl daemon-reload  
[root@ip-172-31-29-0 tools]# systemctl enable hamster-scheduler.service  
Created symlink from /etc/systemd/system/multi-user.target.wants/hamster-  
scheduler.service to /etc/systemd/system/hamster-scheduler.service.
```

```
[root@ip-172-31-29-0 tools]# systemctl start hamster-scheduler.service
[root@ip-172-31-29-0 tools]# systemctl status hamster-scheduler.service
[root@ip-172-31-29-0 tools]# systemctl status hamster-scheduler.service
● hamster-scheduler.service - Hamster Scheduler Service
   Loaded: loaded (/etc/systemd/system/hamster-scheduler.service; enabled; vendor
   preset: disabled)
   Active: active (running) since Tue 2024-12-24 16:39:18 CST; 14s ago
   Main PID: 3836 (dotnet)
   CGroup: /system.slice/hamster-scheduler.service
           └─3836 /hamster/dotnet8/dotnet /hamster/hamster-
scheduler/FlyEnter.Hamster.Scheduler.dll
Copy to clipboardErrorCopied
Hamster Slaver
[root@ip-172-31-29-0 tools]# unzip hamster_slaver_linux_2024.zip -d /hamster/hamster-
slaver
[root@ip-172-31-29-0 tools]# chmod +X /hamster/hamster-
slaver/FlyEnter.Hamster.Slaver.dll
[root@ip-172-31-29-0 tools]# vi /etc/systemd/system/hamster-slaver.service
[Unit]
Description=Hamster Slaver Service
After=network.target rabbitmq.service postgresql.service
Requires=rabbitmq.service postgresql.service

[Service]
Type=simple
Environment=DOTNET_ROOT=/hamster/dotnet8
Environment=ASPNETCORE_ENVIRONMENT=Linux
Environment=PATH=$PATH:/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/hamster/dotnet8
ExecStart=/hamster/dotnet8/dotnet /hamster/hamster-slaver/FlyEnter.Hamster.Slaver.dll
WorkingDirectory=/hamster/hamster-slaver/
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=Hamster Slaver 2024
User=root
Group=root

[Install]
WantedBy=multi-user.target
[root@ip-172-31-29-0 tools]# systemctl daemon-reload
[root@ip-172-31-29-0 tools]# systemctl enable hamster-slaver.service
Created symlink from /etc/systemd/system/multi-user.target.wants/hamster-slaver.service
to /etc/systemd/system/hamster-slaver.service.
[root@ip-172-31-29-0 tools]# systemctl start hamster-slaver.service
```

```
[root@ip-172-31-29-0 tools]# systemctl status hamster-slaver.service
● hamster-slaver.service - Hamster Slaver Service
   Loaded: loaded (/etc/systemd/system/hamster-slaver.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-12-24 16:43:23 CST; 8s ago
   Main PID: 4293 (dotnet)
   CGroup: /system.slice/hamster-slaver.service
           └─4293 /hamster/dotnet8/dotnet /hamster/hamster-slaver/FlyEnter.Hamster.Slaver.dll
```

Copy to clipboardErrorCopied

### [验证仓鼠服务](#)

仓鼠站点:

访问站点: <http://ec2-xxx.cn-north-1.compute.amazonaws.com.cn:42000/>

使用默认管理账号登陆, 并创建用户

进入左侧菜单: 系统设置-服务节点 模块, 情况服务列表, 并重启如下服务, 便于系统自动刷新配置

```
[root@ip-172-31-29-0 tools]# systemctl stop hamster-slaver
[root@ip-172-31-29-0 tools]# systemctl stop hamster-scheduler
[root@ip-172-31-29-0 tools]# systemctl start hamster-slaver
[root@ip-172-31-29-0 tools]# systemctl start hamster-schedulerCopy to clipboardErrorCopied
```

进入左侧菜单: 安全管理-功能菜单 模块, 编辑 Monitor - Scheduler Monitor 项目, 调整其中的默认模块的连接地址, 如下: <http://ec2-xxx.cn-north-1.compute.amazonaws.com.cn:42010/myhangfire>

### 环境配置

正式使用系统前, 需要对系统进行基础的功能配置, 确保各类基础服务都已经配置完成。

主要配置内容有:

基本用户信息: 建立需要登录用户的个人信息, 如名称、域账号等。

平台系统配置: 基础服务, 如公用邮箱、外网访问代理以及平台授权信息等。

登录账号配置: 建立操作用户角色, 绑定用户, 并授权登录等。

## 第3章 环境配置

### 3.1. 用户配置

访问路径: 基础数据 -> 员工信息

#### [功能简述](#)

建立用户个人信息, 为后续创建登录账户做准备。

#### [创建用户](#)

用户基本信息			
工号	00009948	性别	男性 <input type="checkbox"/> 可用
姓名	Hamster	状态	在职
生日	<input type="text"/>	入职日	<input type="text"/>
员工类别	员工类别	部门	<input type="text"/>
部门角色	最大可选择10个项目		
域账号	Domain\user.name	邮箱	user.name@domain.com
个人电话	<input type="text"/>	工作电话	<input type="text"/>
职位	<input type="text"/>	级别	<input type="text"/>

姓名：会显示在登录界面右上角

域账号：如果平台开启域账号登录功能，系统会以该账户进行匹配

邮箱：用户任务报错，系统会使用该邮件地址进行通知

## 3.2. 平台配置

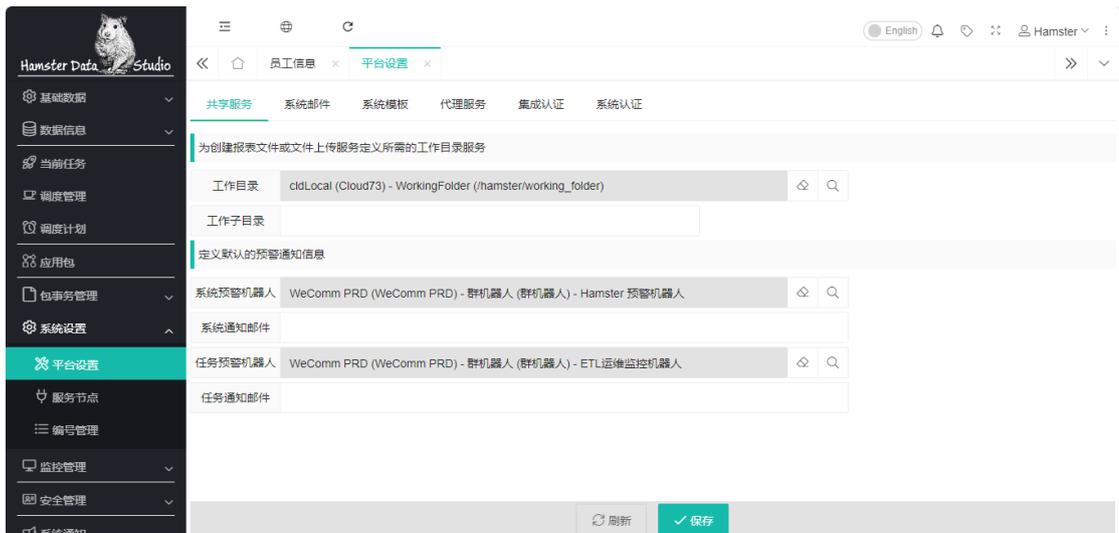
访问路径：系统设置 -> 平台设置

### 功能简述

配置平台公用的基础服务和运行模式。

### 功能配置

#### 共享服务



Hamster Data Studio

系统设置 > 平台设置

共享服务

为创建报表文件或文件上传服务定义所需的工作目录服务

工作目录: cldLocal (Cloud73) - WorkingFolder (/hamster/working\_folder)

工作子目录:

定义默认的预警通知信息

系统预警机器人: WeComm PRD (WeComm PRD) - 群机器人 (群机器人) - Hamster 预警机器人

系统通知邮件:

任务预警机器人: WeComm PRD (WeComm PRD) - 群机器人 (群机器人) - ETL运维监控机器人

任务通知邮件:

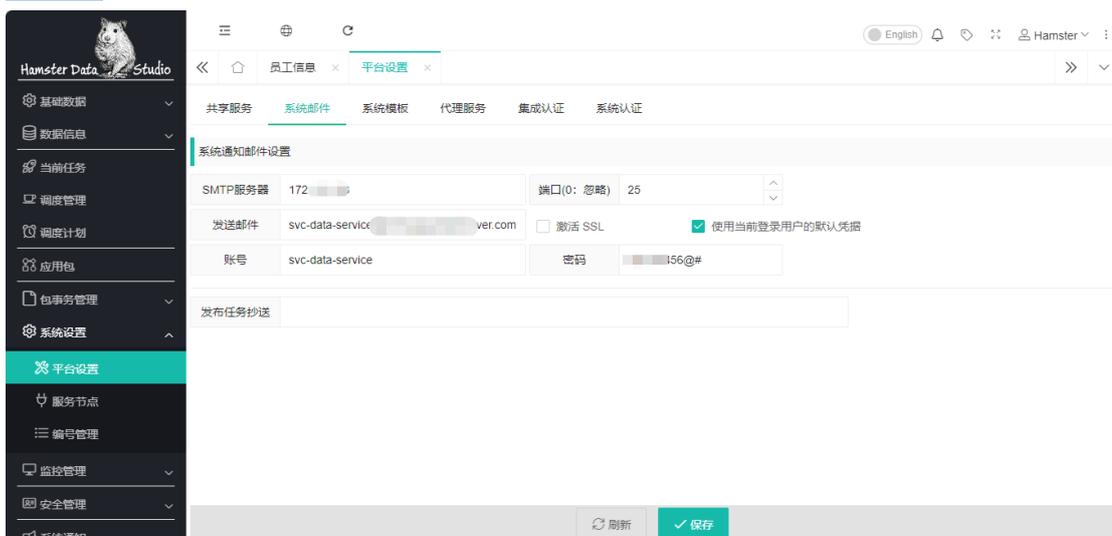
刷新

工作目录：用于保持任务运行期间创建的报表文件等

工作子目录：工作目录下的子目录，存放具体文件

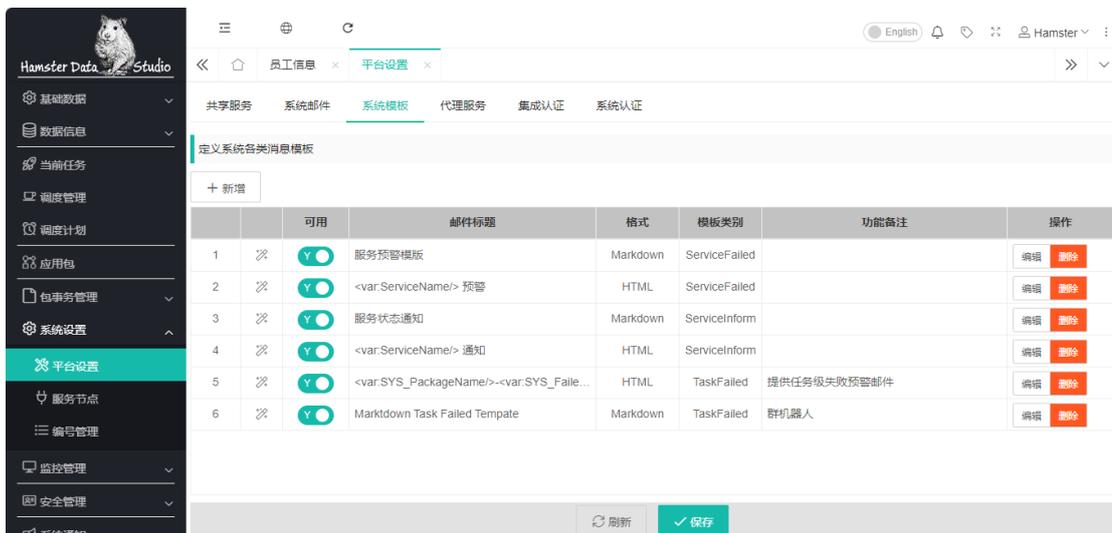
预警通知：提供在平台或任务异常时通知用户，可以选择 Webhook 模式的群机器人或邮件

### 系统邮件



配置系统级的邮件服务，系统默认使用该邮件服务器来推送邮件消息

### 系统模版



用于定义系统服务和任务运行时状态通知的消息内容模版，支持邮件和群消息通知模式

### 代理服务



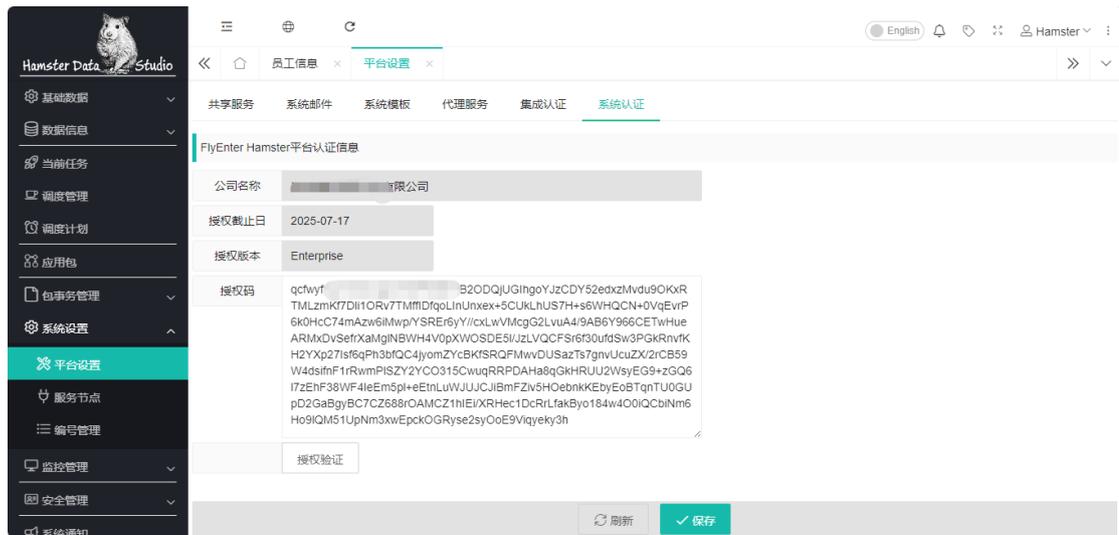
如果是企业内网部署的服务，则需要配置 Nginx 正向代理服务，便于自动将外部域名地址转换成代理地址访问外网。

### 集成认证



如果激活，则启动 Windows 域集成登录功能，用户将使用域账户和域密码进行登录。系统本身不记录域登录密码。

### 系统认证

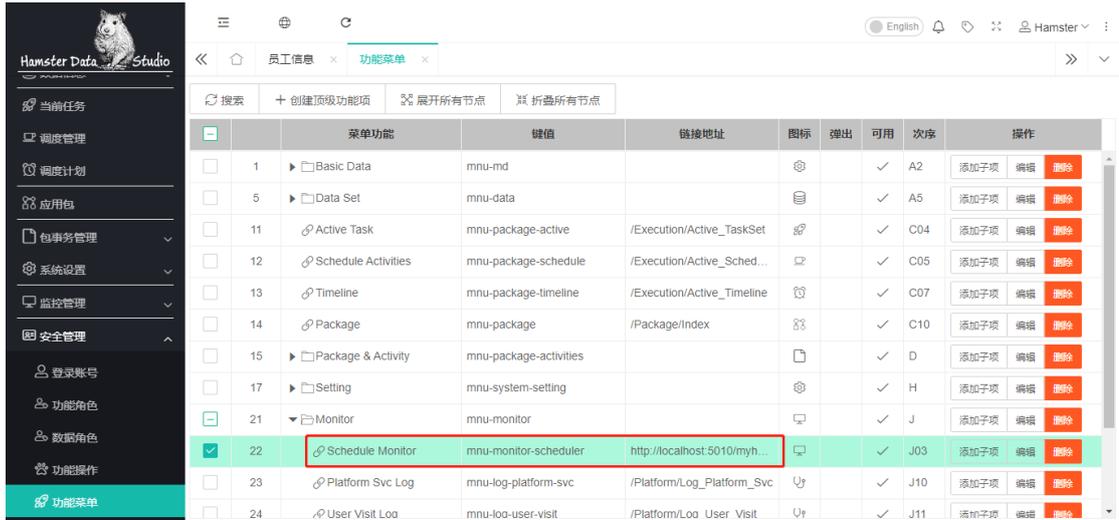


系统的授权信息，社区版在每日调度计划触发的次数上有所限制。

## 3.3. 账户配置

访问路径：安全管理 -> 功能菜单

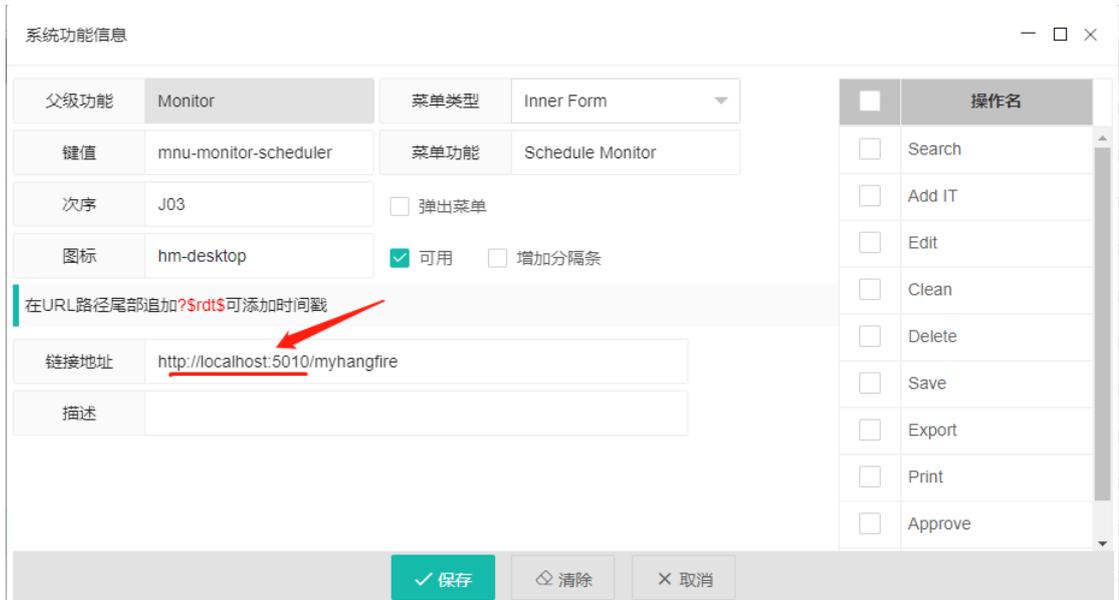
依据实际部署地址来调整后台调度服务监控的监控地址。



菜单功能	键值	链接地址	图标	弹出	可用	次序	操作
1	Basic Data	mnu-md		<input type="checkbox"/>	<input checked="" type="checkbox"/>	A2	添加子项 编辑 删除
5	Data Set	mnu-data		<input type="checkbox"/>	<input checked="" type="checkbox"/>	A5	添加子项 编辑 删除
11	Active Task	mnu-package-active	/Execution/Active_TaskSet	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C04	添加子项 编辑 删除
12	Schedule Activities	mnu-package-schedule	/Execution/Active_Sched...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C05	添加子项 编辑 删除
13	Timeline	mnu-package-timeline	/Execution/Active_Timeline	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C07	添加子项 编辑 删除
14	Package	mnu-package	/Package/Index	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C10	添加子项 编辑 删除
15	Package & Activity	mnu-package-activities		<input type="checkbox"/>	<input checked="" type="checkbox"/>	D	添加子项 编辑 删除
17	Setting	mnu-system-setting		<input type="checkbox"/>	<input checked="" type="checkbox"/>	H	添加子项 编辑 删除
21	Monitor	mnu-monitor		<input type="checkbox"/>	<input checked="" type="checkbox"/>	J	添加子项 编辑 删除
22	Schedule Monitor	mnu-monitor-scheduler	http://localhost:5010/myh...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	J03	添加子项 编辑 删除
23	Platform Svc Log	mnu-log-platform-svc	/Platform/Log_Platform_Svc	<input type="checkbox"/>	<input checked="" type="checkbox"/>	J10	添加子项 编辑 删除
24	User Visit Log	mnu-foo-user-visit	/Platform/Loa_User_Visit	<input type="checkbox"/>	<input checked="" type="checkbox"/>	J11	添加子项 编辑 删除

将原有地址前缀调整成实际站点部署所在的服务器及端口号,

如: `http://xxxx.xx.xx:0000/myhangfire`



系统功能信息

父级功能	Monitor	菜单类型	Inner Form
键值	mnu-monitor-scheduler	菜单功能	Schedule Monitor
次序	J03	<input type="checkbox"/> 弹出菜单	
图标	hm-desktop	<input checked="" type="checkbox"/> 可用	<input type="checkbox"/> 增加分隔条

在URL路径尾部追加?\${rdt}\${可添加时间戳

链接地址	<code>http://localhost:5010/myhangfire</code>
描述	

操作名

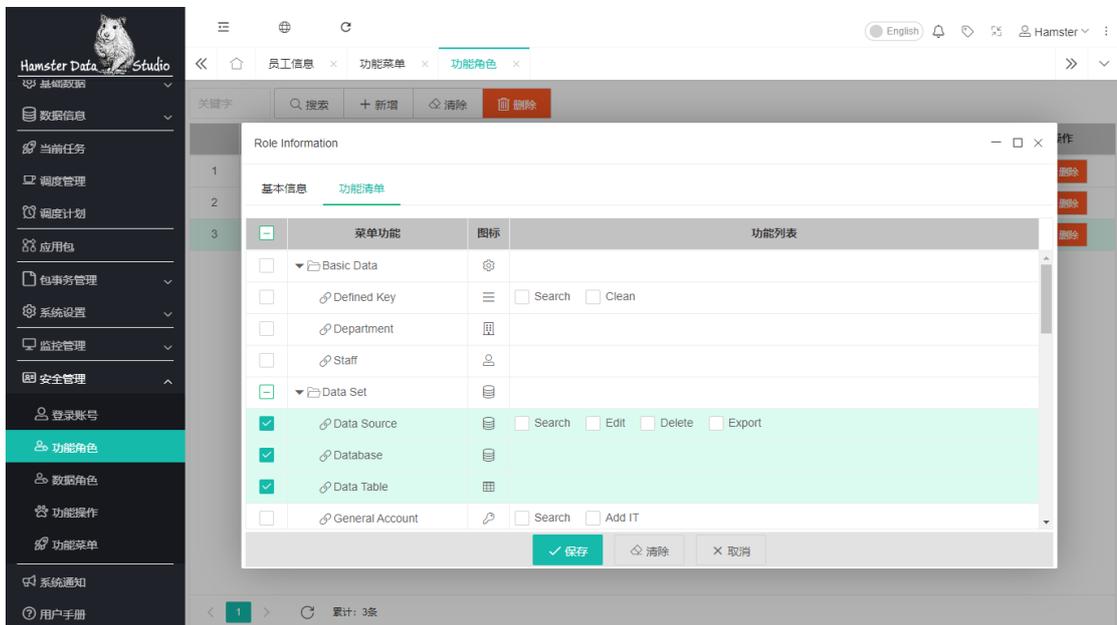
<input type="checkbox"/>	Search
<input type="checkbox"/>	Add IT
<input type="checkbox"/>	Edit
<input type="checkbox"/>	Clean
<input type="checkbox"/>	Delete
<input type="checkbox"/>	Save
<input type="checkbox"/>	Export
<input type="checkbox"/>	Print
<input type="checkbox"/>	Approve

保存 清除 取消

### 功能角色

访问路径: 安全管理 -> 功能角色

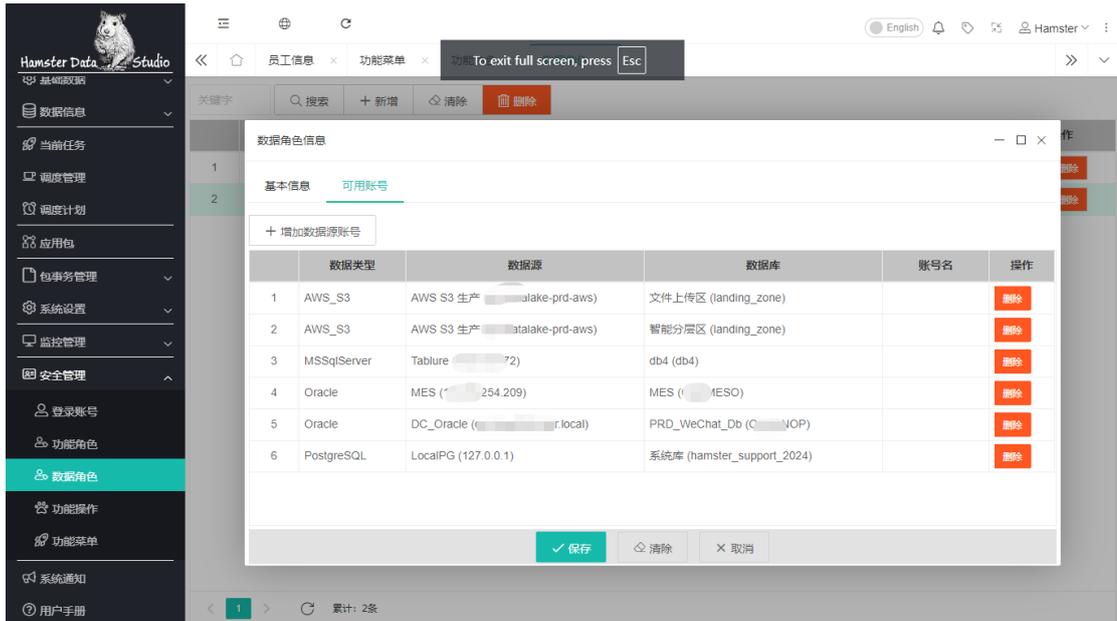
可以创建多个任务操作角色, 不同角色对应不同的功能菜单集, 从而可以支持不同角色的操作人员使用。



### 数据角色

访问路径: 安全管理 -> 数据角色

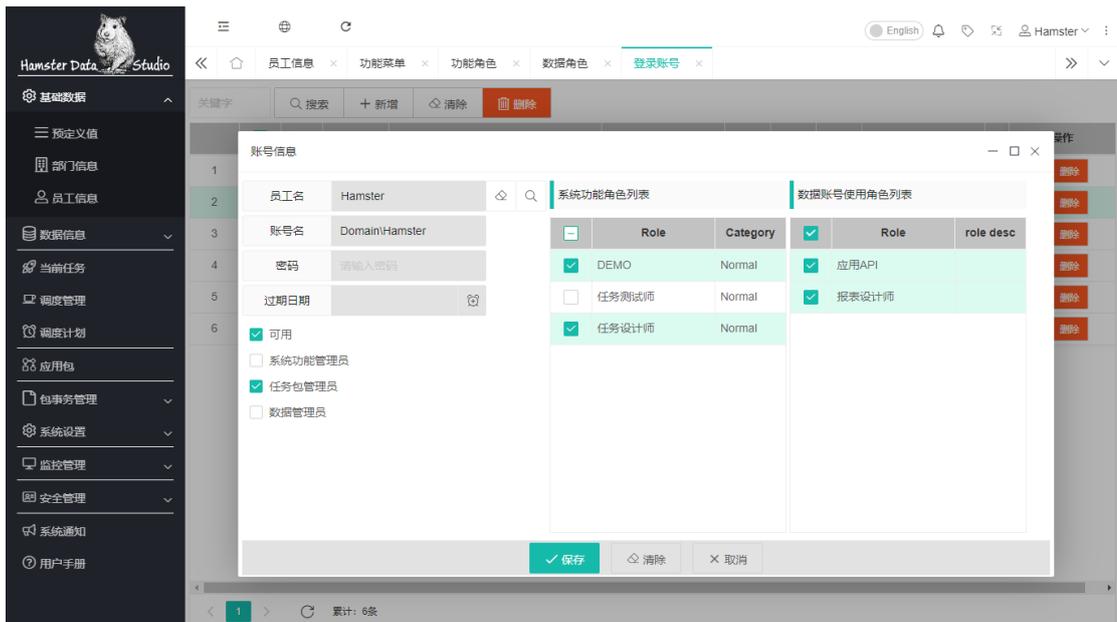
可以创建多个数据操作角色, 不同角色对应不同的数据源清单, 从而可以控制操作人员可以访问使用的源数据。



### 登录账号

访问路径: 安全管理 -> 登录账号

用于创建绑定特定员工的系统登录账号，并设定该账号的访问权限。



基本账号信息:

员工名: 通过搜索按钮选择特定员工

账号名: 对应员工基本信息里面的域账号

密码: 可以预设密码, 当启用“Windows 域登录”时, 该密码不被启用

过期日期: 如果设置了, 过期后, 账号将不可再登录

角色信息:

功能角色: 可以叠加使用

数据角色: 可以叠加使用

管理员信息:

可用: 只有启用了, 账号才可以登录

系统功能管理员: 可以操作平台所有功能, 启用后将忽略功能角色的设置

数据管理员：可以操作平台所有数据源，启用后忽略数据角色的设置

任务包管理员：启用后可以查看所有人设置的任务包及相关的调度计划和运行日志，否则仅可以查看自建的，或被授权的任务包

## 第4章 公用模块

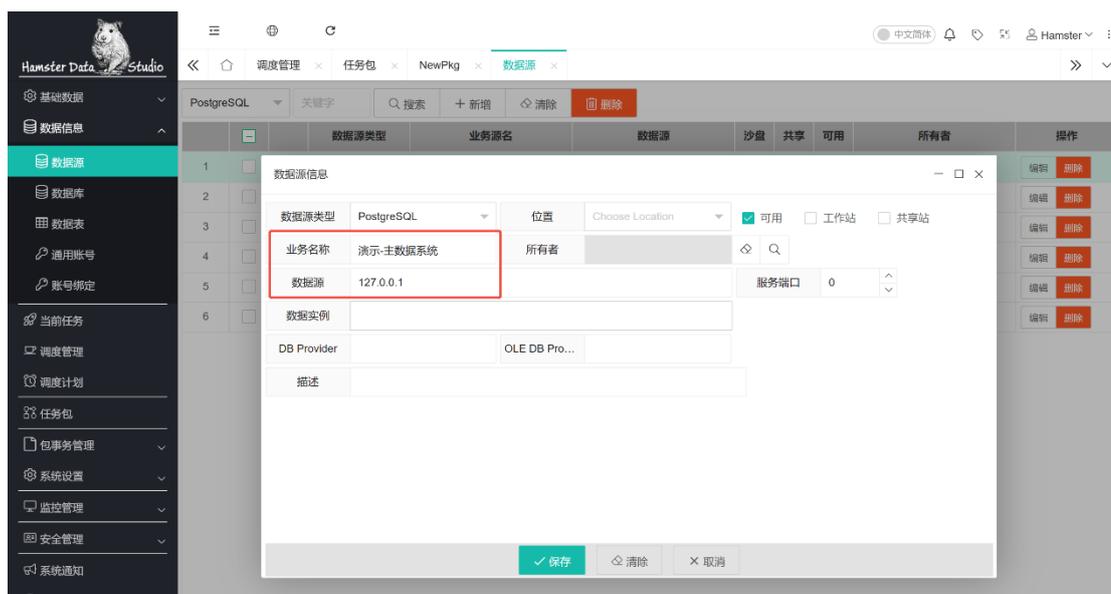
### 4.1. 数据源注册

为了能访问外部数据源，需要在仓鼠平台中先注册数据源信息，数据源注册是一次性的，注册后信息可以共享给其他项目组（如果没有做数据角色限制的话）。仓鼠平台对所有需要访问的数据源均采用一致的3级：[统一数据模型](#)，方便理解和管理。

下面以 PostgreSQL 数据源为例来注册：

#### [注册 - 数据源](#)

在数据信息-数据源模块中，新建一个数据源，在创建窗体中，配置如下信息：



数据源类型：标识各类服务类型

业务源名：提供自定义的数据源业务描述名称

数据源/环境/站点集：提供具体服务地址，如 IP、服务器名等。

注：其它选项将依据不同的数据源参考配置

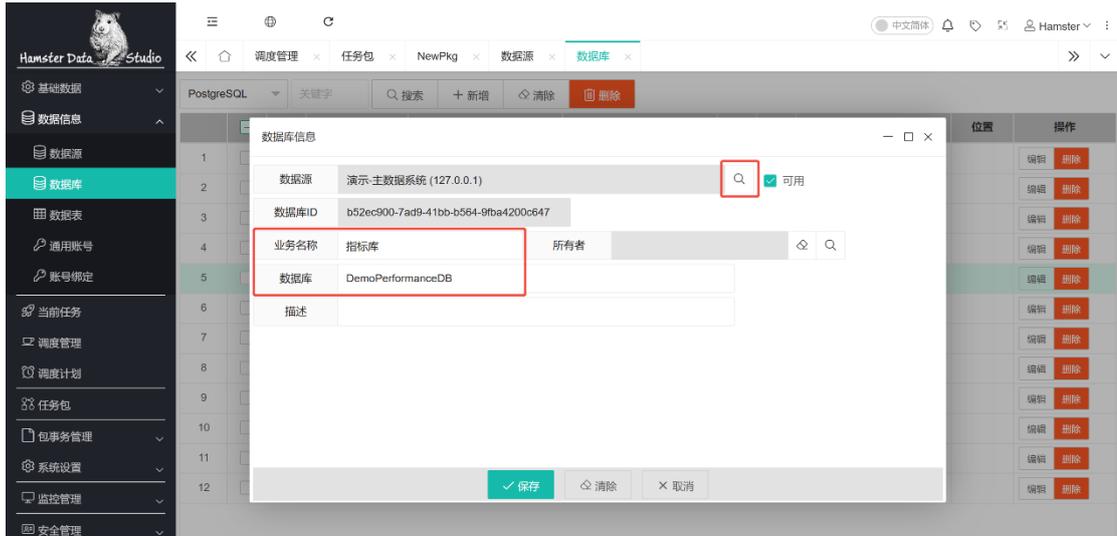
如果当前注册的数据库服务器将作为平台工作站使用的话（后期任务会在该数据库服务器上动态创建工作库），需要配置：

工作站：需要勾选该选项，否则仅为任务使用的一个数据源服务而已。

共享站：如果勾选，则所有任务都可以使用该数据库服务器，如果不勾选，则是专用工作站数据库服务器，可以由特定的任务包使用。

#### [注册 - 数据库](#)

在数据信息-数据库模块中，新建一个数据库，在创建窗体中，配置如下信息：



在新窗体中，点击查询按钮选择先前已注册的数据源记录，然后提供相应的数据库信息。

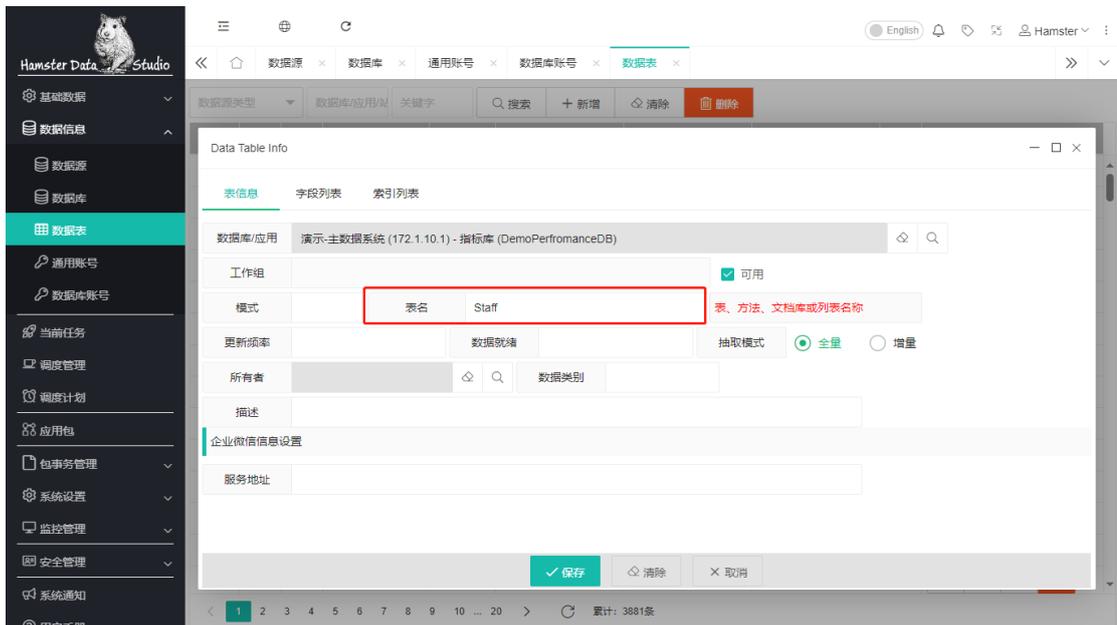
业务库名：提供自定义的数据库业务描述名称

数据库/应用/站点：提供具体的数据库名

注：其它选项将依据不同的数据库参考配置

### [注册 - 数据表](#)

在数据信息-数据表模块中，可以创建多个数据表，如果只是查询外部数据源，注册的时候可以不用状态特定的数据表。只有在需要向特定数据源推送数据的时候，才需要定义表结构。

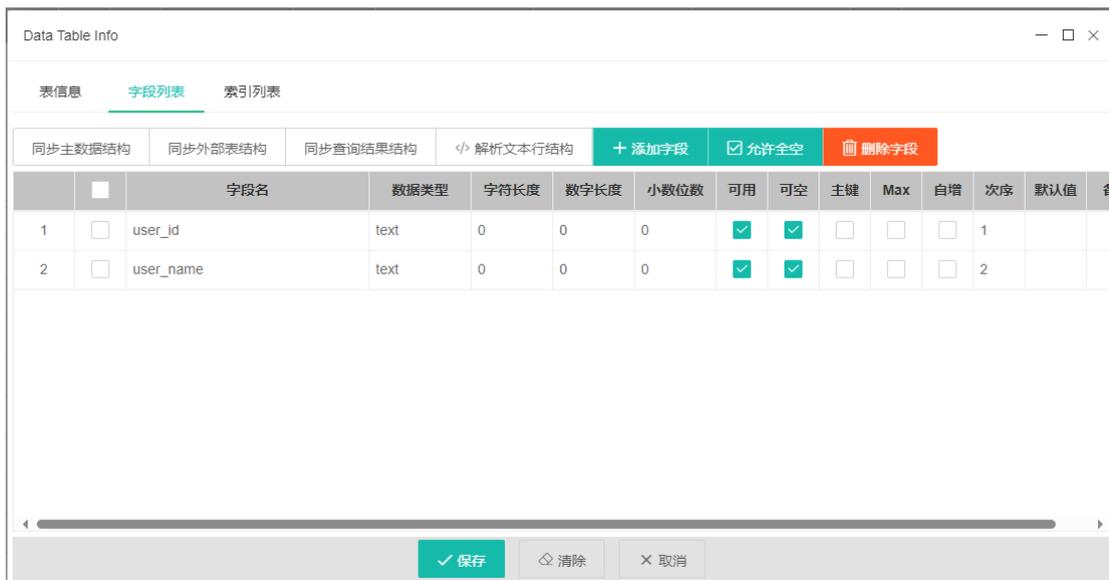


模式：模式的数据库表模式可以不写，如 dbo, public 等，系统会自动检测数据源类型后，在访问时自动添加。

表名：数据库中特定的表名。基于不同的数据库命名习惯，需要注意表名大小写。

服务地址：在访问 Restful API 时，可用于存储完整的 URL 服务地址

在字段列表页签中，可以手工添加表字段列表，也可以通过 [数据库结构同步](#) 功能，自动从相应的数据源中同步表清单和表字段列表。



添加字段：用于手工添加新的字段项

删除字段：删除选中的不需要的字段 注：

系统提供了多种获取现有表结构的方式：

同步主数据库结构：从仓鼠主数据库中读取，并复制已有的表结构信息

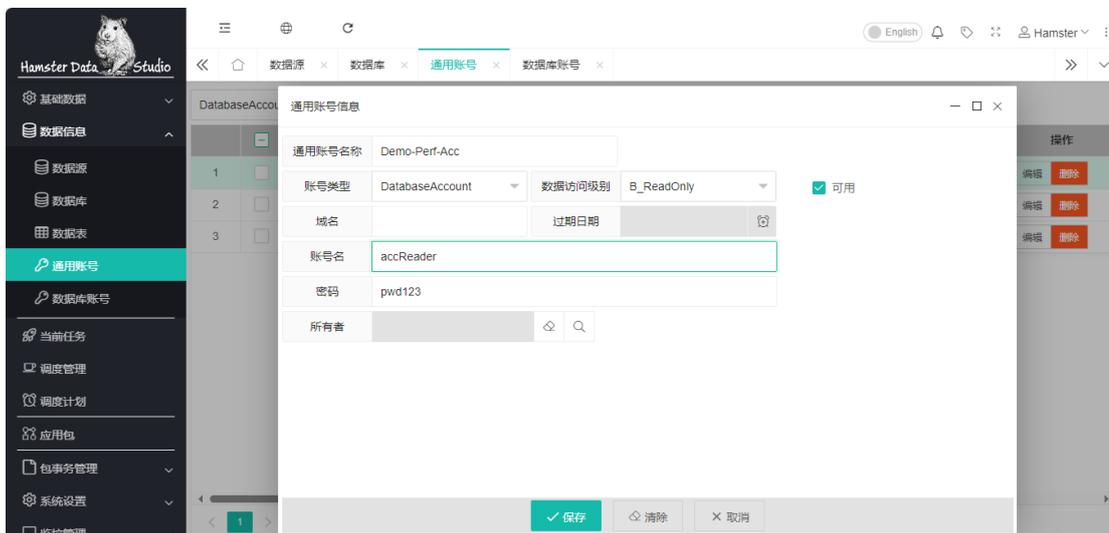
同步外部表结构：动态从外部数据源中查询整表结构

同步查询结果结构：依据查询语句，动态从外部数据源中查询，并提取结果集数据机构

解析文本行结构：通过行模式，粘贴并解析出字段结构信息

### 注册 - 数据账号

在数据信息-通用账号模块中，可以新建一个数据库访问账号，该账号可以绑定多个不同的数据源，从而实现统一管理，并统一更新的能力。



账号类型：可以预定义账号类型，如域账号、数据库通用账号等。

数据访问级别：不同的任务对所使用的账号类型有要求，如只读、可读写或管理员账号等。

域名：如果登记的是域账号，则需要提供账号域名

### 绑定 - 访问账号

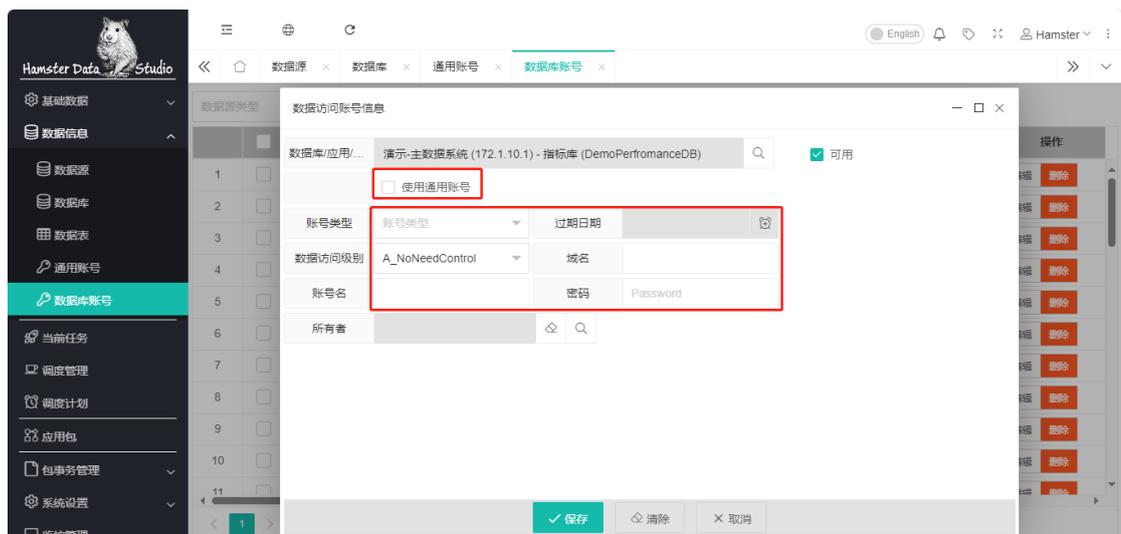
在数据信息-账号绑定模块中，可以将预定义的数据库和一个注册的账号进行绑定，当任务访问该数据库时，系统会自动调用该账号进行访问。

数据库的账号绑定，可以有 2 中模式，

模式 1: 绑定预订账号 (推荐) 该模式下, 需要先沟通使用通用账号选项, 则可以在出现的查询按钮中, 绑定预定义的特定通用账号。



模式 2: 创建专用账号 该模式下, 可以为当前数据库访问提供专用的, 且不被分享使用的账号。后续密码更新时, 需要逐一更新绑定的密码信息。



## 第5章 技术参数

### 5.1. 变量定义

变量是仓鼠系统中一个很强大的功能, 为用户数据操作提供极大的便利性。仓鼠变量可以分为如下几类:

- 系统定义变量
- 系统日期变量
- 用户定义变量

数据遍历变量

任务临时变量

Excel 专用变量

邮件专用变量

上述各类变量类型都可以在任务设计时引用，并起到占位符的作用。运行时，仓鼠系统将根据实际运行环境进行替换成预期的内容。

#### 系统定义变量

该变量是由仓鼠系统在运行前或运行中，为特定任务包创建的。如记录当前任务名称、运行状态或数据采集数量等。调用方式如：<sys:SysParamName/> 其中 *SysParamName* 为系统参数名称，其它部分为系统变量的标识格式。

系统参数名 - SysParamName	含义	备注
SYS_TSK_TaskID_Name	用于记录 TaskID 关联任务的名称	TaskID 需要替换成实际任务 ID，从而与具体任务绑定。例： SYS_TSK_32F31BD1C6D942D2963BAE5D7EE62744_NAME
SYS_TSK_TaskID_QTY	用于记录 TaskID 关联任务的数据量	
SYS_TSK_TaskID_STS	用于记录 TaskID 关联任务的当前状态	
SYS_ExecCaseID	当前执行任务包对应的执行实例 ID	
SYS_PackageID	当前执行任务包 ID	
SYS_PackageName	当前执行任务包名称	
SYS_PackageNo	当前执行任务包编号	
SYS_PackageOwnerEmail	当前执行任务包所有者邮箱	
SYS_ReportName	当前执行任务包关联报	

	表名称	
SYS_ScheduleOwnerEmail	当前执行任务包调度所有者邮箱	
SYS_ScheduleOwnerID	当前执行任务包调度所有者 ID	
SYS_ScheduleOwnerName	当前执行任务包调度所有者名称	
SYS_TaskGroupName	当前执行任务包分组名称	
SYS_FailedTaskCategory	最近失败的任务类型	该组系统变量将在任务实施执行期间出错时，由系统自动创建并更新内容。可在系统预警的邮件或消息提醒中使用。
SYS_FailedTaskID	最近失败的任务 ID	
SYS_FailedTaskNo	最近失败的任务编号	
SYS_FailedTaskName	最近失败的任务名称	
SYS_FailedMessage	最近失败的错误提示信息	
SYS_FailedDateTime	最近失败的时间点	

**系统日期变量**

系统日期变量主要内嵌与用户定义变量中，起到依据任务运行前指定的业务时间进行相应的计算，并自动赋值于隶属的用户定义变量中。该变量由系统预定义名称，不可以修改。

基本格式如：<dt:ParamName(dataParam) ops delta,DateFormat/> 其中可以调整的内容有：

ParamName：必选，为日期变量名称，可以参考如下表。

dataParam：可选，为了支持多个日期变量嵌套使用，如

<dt:Var\_CurrMonday(<dt:Var\_CurrMonth()+9d/>),YYYYMMDD/>，其中 Var\_CurrMonth() 可以作为 Var\_CurrMonday() 变量的计算日期参数，如果没有，则将依据任务的业务运行时间为基准。

ops：可选，提供+或-操作符号，用于增加或减少相应的天数、月份等

delta:如果没有选择 ops，则可选，和 ops 元素配合使用，提供具体的偏移量，如 5y(5 年), 3m(3 个月), 2d(2 天), 2w(2 周), 4h(4 小时)。

DateFormat: 可空, 用于特殊指定当前日期的格式, 如: YYYY 年, MM 月, DD 日, HH24 小时, MI 分钟, SS 秒等。具体格式字符串依据系统数据库运行的数据库平台为标准。

自定义参数信息 - □ ×

该参数将会以<>这样的方式应用在SQL脚本里面, 注意变量名中不要包含空格。

任务组	最大可选择50个项目	<input checked="" type="checkbox"/> 可用
支持对象	支持对象类型	<input checked="" type="checkbox"/> 动态参数 <input type="checkbox"/> 数据连接参数
参数名	preMonth	参数类型 Date
参数值	<dt.Var_CurrMonth()-1m>	日期格式 yyyyMM
数据库名		
次序	0	^ v
描述		

变量名称 ParamName	输出格式案例	功能备注
Var_CurrDate	2024-05-14	获取当期日期, 是任务运行时选择的业务日期为当前日期, 如果没有特殊指定, 则以当前日历日期为准
Var_CurrDateTime	2024-05-14 17:46:12	带时分秒的当前日期
Var_CurrFullTime	2024-05-14 17:46:12.234	带时分秒和毫秒的当前日期
Var_CurrSysDateTime	2024-05-14 17:46:12.234	获取当前操作系统时间, 不受任何变量参数影响, 常用于记录时间戳
Var_CurrMonday	2024-05-13	当前周的周一日期
Var_CurrMonth	2024-05	当前年份
Var_LastDayWithinCurrMonth	2024-05-31	当前月最后一天
Var_FstDayOnNextMonth	2024-06-01	下一个月的第一天
Var_FstMonWithinCurrMonth	2024-05-06	本月内第一个周一
Var_LstMonWithinCurrMonth	2024-05-27	本月内最后一个周一
Var_LastFirIfCurrIsMonOrYesterday	2024-05-13	如果当前是周一, 则返回上周五, 否则返回昨天的日期
Var_Sub3dIfCurrIsMonOrSub2d	2024-05-12	如果当前是周一, 则返回上周五 (-3 天), 否则返回大前天 (-2

变量名称 ParamName	输出格式案例	功能备注
		天)
Var_FstMonthWithinCurrQuarter	2024-04-01	获取当前月份所在季度的第一个年月

案例：（当前日期为 2024-05-14 20:15:20）

表达式	输出结果	含义描述
<dt:Var_CurrDate()-2d/>	2024-05-12	获取当前日期前 2 天日期
<dt:Var_FstMonWithinCurrMonth(<dt:Var_CurrMonth()+1m/>)/>	2024-06-03	获取下一个月的第一个周一日期
<dt:Var_CurrMonday()-2d, YYYYMMDD/> - <dt:Var_CurrMonday()-1d, MMDD/>	20240511 - 0512	获取上个周日和周六的日期，并拼接成特定的格式

### 用户定义变量

由用户在界面中自定义的变量。调用方式如：<var:UserParamName|{dataformat}/>，其中 *UserParamName* 为用户自定义变量的名称，*format* 为当前变量值的显示格式，如果不需要考虑显示格式，则后半段可以忽略，简写成：<var:UserParamName/>

### 数据遍历变量

用户在可循环遍历的任务中，引用任务查询中涉及到的数据库字段时用到的数据库字段变量。通过该变量，用户可以引用当前数据集中特定的字段内容。调用方式

如：<db:DbFieldName|{dataformat}/>，其中 *DbFieldName* 为当前数据集中的字段名称，*format* 为当前变量值的显示格式，如果不需要考虑显示格式，则后半段可以忽略，简写成：<db:DbFieldName/>

### 任务临时变量

该变量只存在于特定任务的生命执行周期内，如钉钉消息，企业微信消息等任务。用于临时保存当前任务逐条数据记录执行时获取的运行（发送）结果。如在发送批量消息时，记录每个消息返回的消息 ID 或任务 ID，该变量可以在当前 *任务的遍历脚本* 中被使用，可用于更新特定消息的发送状态，如回写消息 ID：<tsk:returnId/>，变量的内容形式由当前任务决定。该变量在下一条记录发送后会被重写，任务结束后，变量会被清空。

记录当前子记录任务的执行返回结果，如消息 ID 等。



**Excel 专用变量**

[xls:currow], [xls:nextrow], [FieldName]Copy to clipboardErrorCopied

**邮件专用变量**

变量	场景	备注
<sys:rowindex/>	数据行	自动实现行号
<db:FiledName  {0:D2} />	数据行	用于显示当前数据行中特定的字段值, 字段名:FieldName
<fun:sum(FieldName)   {0:C2} />	汇总行	针对当前数据集中每行 FieldName 字段值进行累加计算
<fun:count(FieldName)/>	汇总行	针对当前数据集中, 每行的 FieldName 字段值进行记数, 非累加
<fun:max(FieldName)/>	汇总行	获取当前数据集中, FieldName 字段值最大的值
<fun:min(FieldName)/>	汇总行	获取当前数据集中, FieldName 字段值最小的值
<obj:SheetObject_SheetId/>	内嵌图表图片	用于在邮件中迁入对应报表中特定图表保存的图片, 其中 SheetId 为对应报表页签 Guid 值

**5.2. 文件筛选**

系统中涉及到的文件过滤功能, 是基于正则表达式功能实现。同时, 由支持常规的文件筛选和过滤模式:

系统支持多个过滤项, 各个过滤项以分号进行分割。当目标文件条件只要满足其中一个过滤项条件, 就可以被选中进行后续操作。每个过滤项内部, 又可以通过关键字 and 或 or 进行组合多个不同子筛选条件。子筛选条件含对文件名、文件大小和文件最后修改日期的检测。

系统提供两种不同的匹配模式:

**简单查询模式**

过滤模版	匹配描述	备注
*.*	所有文件	
*.xlsx	只获取 xlsx 文件类型	
*.csv;*.xlsx	只获取 csv 或 xlsx 这两类文件	其它类型将被忽略

### 高级正则模式

该模式下，将利用正则表达式的强大过滤功能进行文件条件的过滤匹配。

不区分文件名的大小写。

当前正则表达式默认有添加 `^` 前置符号，确保匹配必须从字符串或一行的开头开始。

`.` 代表一个非回车符的任何一个字符，配合 `*` 一起，表示零个或多个字符。配合 `?` 一起，则标识零个或一个字符。如果仅仅只表示文件扩展名前点号，则需要标识为：`\.`

`.*` 代表的是零个或多个字符。

`.?` 代表是是零个或 1 个字符。

`\.` 代表的是文件扩展名前面的 `.` 点符号。

`sz` 代表对文件的字节数进行大小匹配，比较符号为：`>`，`<`和`=`，单位有：KB，MB 和 GB。

`md` 代表对文件的最后修改日期进行匹配，比较符号支持：`>`，`<`和`=`。

`and` 标识文件名要满足当前筛选项中所有的子项目匹配条件，才可以被选中(同一个筛选项中，不可以同时含有 `and` 和 `or`)。

`or` 标识文件名只要满足当前筛选项中一个子项目的匹配条件，既可以被选中(同一个筛选项中，不可以同时含有 `and` 和 `or`)。

`<var:Var_Name/>` 系统在拆选项中支持自定义变量名的迁入，运行时，会先解析出变量名，后再用于文件过滤。

过滤模版	匹配描述
CRM_202012_.*\.csv	仅匹配文件名称以 CRM_202012_ 开头，后面可能含有若干字符，且以 .csv 结尾的文件
CRM_202012_?.?.?.xlsx	仅匹配文件名以 CRM_202012_ 开头，且随后可能含 0~3 个字符的 .xlsx 文件，如：CRM_202012_A.xlsx；CRM_202012_ABC.xlsx
CRM_<var:CurrentDay/>.*\.xlsx	以 CRM_ 字符开头，含自定义变量 <var:CurrentDay/>，随后为 _ 下划线字符，其后再含若干字符的 .xlsx 文件，匹配类似于：CRM_20210427_release_II.xlsx，CRM_20210427_ABC.xlsx 文件
CRM_2021_.*\.csv and sz>10kb and md<2021-05-01	匹配文件名以 CRM_2021_ 开头的 .csv 文件，且 size>10KB，且 在 2021-05-01 之前被修改 的文件
*.xlsx or sz<10mb or md>2021-02-01 14:30:00	匹配所有 .xlsx 文件，或文件 size<10MB，或 在 2021-02-01 14:30:00 点后修改过的文件

### 5.3. 通用数据连接

通用数据源连接字符串主要用于传递标准化的数据源连接信息，如统一了关系型数据库连接字符串中各命名参数名称不一致的情况，便于第三方代码滴哦用。如执行 Python 代码任务中获取数据库连接信息等。

连接信息以 JSON 格式序列化，如下案例：

```
{
  "DsCategory": "PostgreSQL",
  "Host": "127.0.0.1",
  "Port": 5432,
  "DbName": "pkg2024090001_04f91d6454814380ae3d6570100eb7dc",
  "UserId": "hm_04f91d6454814380ae3d6570100eb7dc",
  "UserPwd": "8b96c57010cba11249ff76c4bdfdc8b",
  "Timeout": 120,
  "IsDbBasedConnection": true,
  "IsForSSISPackage": false
}
```

#### [关系型数据源](#)

关系型数据库连接字符串中较好的实现了标准化，支持常规的 PostgreSQL、MSSqlServer、Oracle 和 MySQL，包括与之兼容的各类云衍生数据库产品。

连接参数：

参数名称	类型	案例	备注
DsCategory	string	PostgreSQL	数据源类型，见随后类型定义清单
AccCategory	string	DatabaseAccount	用户账号类型，见随后账号类型定义清单
Host	string	127.0.0.1	数据库服务名或地址
Port	int	5432	端口号，为 0 则不设置
UserId	string	postgres	用户账号
UserPwd	string	****	用户密码
Timeout	int	0	连接超时，为 0 则不设置
DbName	string	hamster_support	数据库名称
OleDbProvider	string		OleDb 驱动，可空
IsDbBasedConnection	bool	true	建立连接时是否含有数据库名称
IsForSSISPackage	bool	false	是否为支持 SSIS 连接

#### [文件服务源](#)

提供针对文件服务的各类数据源的连接信息，各类服务参数上可能有所不同。

#### [SharePoint 站点](#)

参数名称	类型	案例	备注
DsCategory	string	SharePoint	数据源类型，见随后类型定义清单
AccCategory	string	NTServiceAccount	用户账号类型，见随后账号类型定义清单
HostUrl	string	<a href="https://www.flyenter.com">https://www.flyenter.com</a>	SharePoint 服务域名地址
SiteUri	string	sites/it/doc	站点相对地址
Port	int	0	端口号，为 0 则不设置
UserNtAccount	string	domain\ops	用户域账号
UserPwd	string	****	用户密码
IsSSLCred	bool	true	是否建立 SSL 加密验证

#### 文件服务器(SFTP/共享目录/本地目录)

参数名称	类型	案例	备注
DsCategory	string	SFTP	数据源类型，见随后类型定义清单
AccCategory	string	DatabaseAccount	用户账号类型，见随后账号类型定义清单
Host	string	129.10.10.20	SFTP 服务地址
RootFolder	string	share	根目录
WkrFolder	string	it/working_folder	子工作目录
Port	int	0	端口号，为 0 则不设置
UserId	string	postgres	用户域账号
UserPwd	string	****	用户密码

#### AWS S3 服务

参数名称	类型	案例	备注
DsCategory	string	AWS_S3	数据源类型，见随后类型定义清单
AccCategory	string	DatabaseAccount	用户账号类型，见随后账号类型定义清单

参数名称	类型	案例	备注
RegionEndPoint	string	CNNorthWest1	服务区域
ServiceURL	string	<a href="https://bucket.vpce-0xx.s3.cn-northwest-1.vpce.amazonaws.com.cn">https://bucket.vpce-0xx.s3.cn-northwest-1.vpce.amazonaws.com.cn</a>	内部服务端点地址
BucketName	string	hamster_dlk_prd	S3 桶名
WkrFolder	string	it/working_folder	子工作目录
DefS3StorageClass	string	INTELLIGENT_TIERING	文件存储类型
UserAccessID	string	AKIATMVGdxxxx	AK
UserSecretKey	string	tW0Yp//LYU4+dxxxx	SK

#### [数据源和账号类别信息](#)

##### [用户账号类型定义](#)

账号类型	案例	备注
DatabaseAccount	postgres	数据库账号
NTServiceAccount	domain\ops	NT 域账号
LocalServiceAccount	.\ops	NT 本地域名账号

##### [数据源类型定义](#)

账号类型	案例	备注
PostgreSQL		
MSSqlServer		
MySQL		
Oracle		
SFTP		
LocalFolder		本地目录
ShareFolder		共享目录
AWS_S3		
EnterpriseWeChat		企微
DingTalk		钉钉

账号类型	案例	备注
Feishu		飞书
Teams		Teams
MiniO		
Redis		
SAPR3		
SAPBW		
Azkaban		阿兹卡班
SharePoint		
MSAnalysisService		
Vertica		
File		

## 5.4. 数据格式

### 货币值

格式符	初始值	输出样式	备注
{0:C}	10.2	¥10.20	当前环境币别，未指定小数位数参数，则默认 2 位小数
{0:C1}	10.25	¥10.3	当前环境币别，指定保留一位小数，四舍五入

### 整数

格式符	初始值	输出样式	备注
{0:D}	12345	12345	以数值模式显示，默认不调整位数
{0:D6}	12345	012345	因指定 6 位数，则前置补 0
{0:D6}	-12345	-012345	负数，则负号后补 0

### 所有数值

格式符	初始值	输出样式	备注
{0:G}	1234.5	1234.5	默认为原有数值位数

格式符	初始值	输出样式	备注
{0:G4}	1234.5	1235	4位有效位数，则四舍五入，仅保留4位数字
{0:G4}	12345	1.235E+04	仅保留4位有效位，已经小于有效位数了
{0:G6}	1234.5	1234.5	超过当前有效位数，仍保留原位数

**整数和十进制小数**

格式符	初始值	输出样式	备注
{0:F}	12345	12345.00	默认保留2位小数
{0:F}	12345.6	12345.60	默认保留2位小数
{0:F0}	12345.6	12346	保留0位小数，四舍五入
{0:F1}	12345.67	12345.7	保留1位小数，四舍五入
{0:F3}	-12345.6	-12345.600	支持负数，补为3位小数
{0:N}	12345	12,345.00	默认2位小数，整数部分添加千分位
{0:N3}	12345.6	12,345.600	指定3位小数，整数部分添加千分位
{0:N3}	-12345.6	-12,345.600	指定3位小数，整数部分添加千分位

**百分比**

格式符	初始值	输出样式	备注
{0:P}	0.12345	12.35%	乘以100后，默认保留2位小数，四舍五入
{0:P1}	0.12345	12.3%	乘以100后，保留1位小数，四舍五入
{0:P1}	1.2345	123.5%	乘以100后，保留1位小数，四舍五入

**十六进制**

格式符	初始值	输出样式	备注
{0:X}	15	F	整数，转换成十六进制
{0:X2}	15	0F	整数，转换成十六进制
{0:X}	255	FF	整数，转换成十六进制
{0:X}	-255	FFFFFF01	整数，转换成十六进制

**日期格式**

格式符	初始值	输出样式	备注
{0:d}	2009-06-15T13:45:30	6/15/2009	短日期模式(en-

格式符	初始值	输出样式	备注
			US)
{0:D}	2009-06-15T13:45:30	Monday, June 15, 2009	长日期模式(en-US)
{0:f}	2009-06-15T13:45:30	Monday, June 15, 2009 1:45 PM	完整日期/时间模式(短时间)(en-US)
{0:F}	2009-06-15T13:45:30	Monday, June 15, 2009 1:45:30 PM	完整日期/时间模式(长时间)(en-US)
{0:g}	2009-06-15T13:45:30	6/15/2009 1:45 PM	常规日期/时间模式(短时间)(en-US)
{0:G}	2009-06-15T13:45:30	6/15/2009 1:45:30 PM	常规日期/时间模式(长时间)(en-US)
{0:M}/{0:m}	2009-06-15T13:45:30	June 15	月/日模式(en-US)
{0:R}/{0:r}	2009-06-15T13:45:30	Mon, 15 Jun 2009 20:45:30 GMT	RFC1123 模式
{0:s}	2009-06-15T13:45:30 (DateTimeKind.Local)	2009-06-15T13:45:30	可排序日期/时间模式
{0:t}	2009-06-15T13:45:30	1:45 PM	短时间模式(en-US)
{0:T}	2009-06-15T13:45:30	1:45:30 PM	长时间模式(en-US)
{0:u}	2009-06-15T13:45:30	2009-06-15 13:45:30Z	通用可排序日期/时间模式
{0:U}	2009-06-15T13:45:30	Monday, June 15, 2009 8:45:30 PM	通用完整日期/时间模式(en-US)
{0:Y}/{0:y}	2009-06-15T13:45:30	June 2009	年月模式(en-US)

下面提供的是自定义的格式字符串，区别与上述的系统标识格式符号，自定义的格式符是可以多个连写来标识一个特定格式的，如果是碰到一个单个的自定义格式符时，优先会按系统标准符号进行解析 使用年方式如下：**{0: 0000}**，**{0: #, ##}**，**{0: dd MM YYYY}**

#### 自定义数字格式符

格式符	属性	说明	示例
0	零占位符	用对应的数字（如果存在）替换零；否则，将在结果字符串中显示零。	1234.5678 ("00000") -> 01235; 0.45678 ("0.00", en-US) -> 0.46; 0.45678 ("0.00", fr-FR) -> 0,46
#	数字占位符	用对应的数字（如果存在）替换“#”符号；否则，不会在结果字符串中显示任何数字。请注意，如果输入字符串中的相应数字是无意义的 0，则在结果字符串中不会出现任何数字。例如，0003 ("####") -> 3。	1234.5678 ("#####") -> 1235; 0.45678 ("#.##", en-US) -> .46; 0.45678 ("#.##", fr-FR) -> ,46
.	小数点	确定小数点分隔符在结果字符串中的位置。	0.45678 ("0.00", en-US) -> 0.46; 0.45678 ("0.00", fr-FR) -> 0,46
,	组分隔符和数字比例换算	用作组分隔符和数字比例换算说明符。作为组分隔符时，它在各个组之间插入本地化的组分隔符字符。作为数字比例换算说明符，对于每个指定的逗号，它将数字除以 1000。	组分隔符说明符：2147483647 ("##,#", en-US) -> 2,147,483,647; 2147483647 ("##,#", es-ES) -> 2.147.483.647; 比例换算说明符：2147483647 ("#,#,," , en-US) -> 2,147; 2147483647 ("#,#,," , es-ES) -> 2.147
%	百分比占位符	将数字乘以 100，并在结果字符串中插入本地化的百分比符号。	0.3697 ("%#0.00", en-US) -> %36.97; 0.3697 ("%#0.00", el-GR) -> %36,97; 0.3697 ("###.0%", en-US) -> 37.0%; 0.3697 ("###.0%", el-GR) -> 37,0%
‰	千分比占位符	将数字乘以 1000，并在结果字符串中插入本地化的千分比符号。	0.03697 ("%#0.00‰", en-US) -> 36.97‰; 0.03697 ("%#0.00‰", ru-RU) -> 36,97‰

格式符	属性	说明	示例
	符		
E0, E+0, E-0, E0, E+0, E-0	指数表示法	如果后跟至少一个 0 (零), 则使用指数表示法设置结果格式。 “E” 或 “e” 指示指数符号在结果字符串中是大写还是小写。 跟在 “E” 或 “e” 字符后面的零的数目确定指数中的最小位数。 加号 (+) 指示符号字符总是置于指数前面。 减号 (-) 指示符号字符仅置于负指数前面。	987654 (“#0.0e0”) -> 98.8e4; 1503.92311 (“0.0##e+00”) -> 1.504e+03; 1.8901385E-16 (“0.0e+00”) -> 1.9e-16
\	转义符	使下一个字符被解释为文本而不是自定义格式说明符	987654 (“###00#”) -> #987654#
string	文本字符串分隔符	指示应复制到未更改的结果字符串的封闭字符。	68 (“# ’degrees’”) -> 68 degrees; 68 (“# degrees’”) -> 68 degrees
;	部分分隔符	通过分隔格式字符串定义正数、负数和零各部分。	12.345 (“#0.0#; (#0.0#); - \0-”) -> 12.35; 0 (“#0.0#; (#0.0#); -\0-”) -> -0-; -12.345 (“#0.0#; (#0.0#); -\0-”) -> (12.35); 12.345 (“#0.0#; (#0.0#)”) -> 12.35; 0 (“#0.0#; (#0.0#)”) -> -> 0.0; -12.345 (“#0.0#; (#0.0#)”) -> (12.35)

#### 自定义日期格式符

格式符	说明	示例
d	一个月中的某一天 (1 到 31)	2009-06-01T13:45:30 -> 1; 2009-06-15T13:45:30 -> 15

格式符	说明	示例
dd	一个月中的某一天（01 到 31）。	2009-06-01T13:45:30 -> 01; 2009-06-15T13:45:30 -> 15
ddd	一周中某天的缩写名称。	2009-06-15T13:45:30 -> Mon (en-US)
dddd	一周中某天的完整名称	2009-06-15T13:45:30 -> Monday (en-US)
f”	日期和时间值的十分之几秒	2009-06-15T13:45:30.6170000 -> 6; 2009-06-15T13:45:30.05 -> 0
ff	日期和时间值的百分之几秒	2009-06-15T13:45:30.6170000 -> 61; 2009-06-15T13:45:30.0050000 -> 00
fff	日期和时间值的千分之几秒	6/15/2009 13:45:30.617 -> 617; 6/15/2009 13:45:30.0005 -> 000
ffff	日期和时间值的万分之几秒	2009-06-15T13:45:30.6175000 -> 6175; 2009-06-15T13:45:30.0000500 -> 0000
ffffff	日期和时间值的十万分之几秒	2009-06-15T13:45:30.6175400 -> 61754; 6/15/2009 13:45:30.000005 -> 00000
fffffff	日期和时间值的百万分之几秒	2009-06-15T13:45:30.6175420 -> 617542; 2009-06-15T13:45:30.0000005 -> 000000
fffffff	日期和时间值的千万分之几秒	2009-06-15T13:45:30.6175425 -> 6175425; 2009-06-15T13:45:30.0001150 -> 0001150
F	如果非零，则为日期和时间值的十分之几秒。	2009-06-15T13:45:30.6170000 -> 6; 2009-06-15T13:45:30.0500000 -> (无输出)
yyy	年份（最少三位数字）	0001-01-01T00:00:00 -> 001; 0900-01-01T00:00:00 -> 900; 1900-01-01T00:00:00 -> 1900; 2009-06-15T13:45:30 -> 2009
yyyy	由四位数字表示的年份	0001-01-01T00:00:00 -> 0001; 0900-01-01T00:00:00 -> 0900; 1900-01-01T00:00:00 -> 1900; 2009-06-15T13:45:30 -> 2009
yyyyy	由五位数字表	0001-01-01T00:00:00 -> 00001; 2009-06-15T13:45:30 -> 02009

格式符	说明	示例
	示的年份	
z	相对于 UTC 的小时偏移量，无前导零。	2009-06-15T13:45:30-07:00 -> -7
zz	相对于 UTC 的小时偏移量，带有表示一位数值的前导零。	2009-06-15T13:45:30-07:00 -> -07
string	文本字符串分隔符	2009-06-15T13:45:30 ("arr:" h:m t) -> arr: 1:45 P; 2009-06-15T13:45:30 ('arr:' h:m t) -> arr: 1:45 P
%	将下面的字符定义为自定义格式说明符	2009-06-15T13:45:30 (%h) -> 1
\	转义字符	2009-06-15T13:45:30 (h \h) -> 1 h

#### 自定义日期变量格式字符串

类别	格式符	说明	示例
年	YYYY	四位数年份	2023
	YY	两位数年份	23
月	Month	完整的月份名称	January
	Mon	月份缩写	Jan
	MM	两位数月份	01-31
日	Day	完整的星期几名称	Sunday
	Dy	星期几缩写	Sun
	DD	两位数日期	01
小时	HH	12 进制两位数小时	01-12
	HH24	24 进制两位数小时	00-23
分钟	MI	两位数分钟	00-59
秒	SS	两位数的秒	00-59

## 第6章 常见问题

### 6.1. FTP/SFTP

如果是默认端口，如 22，则不需要额外配置端口号

如果 FTP 部署在 Linux 环境，则需要注意路径名称字母的大小写（路径大小写敏感）

### 6.2. 创建报表文件

Tablename is not unique

错误描述：在创建 Excel 报表文件时，在更新数据页签时报错

触发方法：HandleRawDataWorksheetGeneration。

错误排查：

如果创建 Excel 文件时使用了模版文件

可以排查一下数据页签是否按选择了按表对象模式创建 选项

如果是，再排查一下模版文件中对应数据区域是否已经有了同名的数据对象。

如果是，需要先清理模版文件里面数据区域的对象命名，否则冲突

导出图表对象图片不完整，或未更新

当前功能仅支持是通过用户自定义模版方式创建的图表导出。

Index was outside the bounds of the array.

错误描述：在创建报表后，导出其中的图表图像时报错

触发方式：ExportChartImage

错误排查：

先排查图表对象关联的透视图是否配置正确

如果没问题，再排查透视图关联的数据表对象的数据更新方式是否有问题，因为使用的是用户自定义模版，所以在数据更新时，为了确保原有的数据列表对象不变，多数采用的是不添加表头的，且在模版列名行下一行，以单元格下插入模式的数据更新方式。

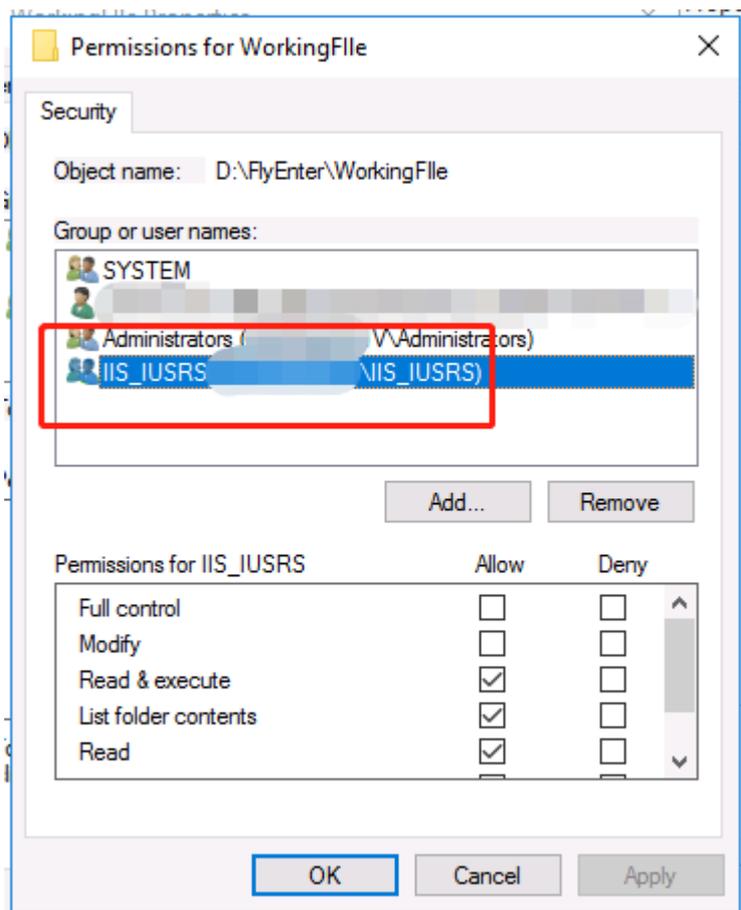
[运行记录中报表文件下载失败](#)

		No	Task Name	Sort	Start Time	End Time	Status	Ops
1			Create Working DB	-100	08-23 10:31	10:31	Completed	Log More Ops
2			Grant User Access	-60	08-23 10:31	10:31	Completed	Log More Ops
3		RP202408-0010	Validate Status Row Count	10	08-23 10:31	10:31	Completed	Log More Ops
4		RP202408-0011	Validate Inv Row Count	15	08-23 10:31	10:31	Completed	Log More Ops
5		APIW202408-0033	Data Warning - 数据异常	20	08-23 10:31	10:31	Ignore	Log More Ops
6		GE202408-0016	创建报表	25	08-23 10:31	10:31	Completed	Log More Ops
7		RP202408-0012	获取汇总值	30	08-23 10:31	10:31	Completed	Log More Ops
8		MAL202408-0009	报表发送	35	08-23 10:31	10:31	Completed	Log More Ops
9		APIW202408-0034	发送群通知	40	08-23 10:31	10:31	Completed	Log More Ops

报表文件创建后，会先存在于系统设置-平台设置-共享服务-工作目录-工作子目录中。

如果该工作目录为 Windows 环境下的本地目录（非网络共享目录）。在报表创建成功后，点击左侧报表图标，应该可以直接下载该报表文件，但如果提示：文件没找到，或已经被删除，且文件确认存在的话，需要先配置工作目录的访问权限。

部署于 IIS 目录下面的站点服务，访问本地其它目录时，需要添加如下账号 IIS\_IUSRS。



默认授权即可，这时候页面文件就可以正常下载了。

### 6.3. 企业微信审批

无法获取金额字段 错误描述：按企业微信后台导出的审批记录，无法获取金额字段类型的数据 问题原因：企业微信后台导出的金额字段，会默认增加“（元）”这样的币别，但实际的字段名中是没有的，所以对应不上。 解决方案：按表单中字段名进行匹配，及不含额外的（元）后缀。同时，接口返回的数据中，不含金额大写内容。

	K	L
	总金额 (元)	金额大写
1856.00		壹仟捌佰伍拾陆圆
275.00		贰佰柒拾伍圆
219.38		贰佰壹拾玖圆叁角捌分
76.00		柒拾陆圆
253.00		贰佰伍拾叁圆