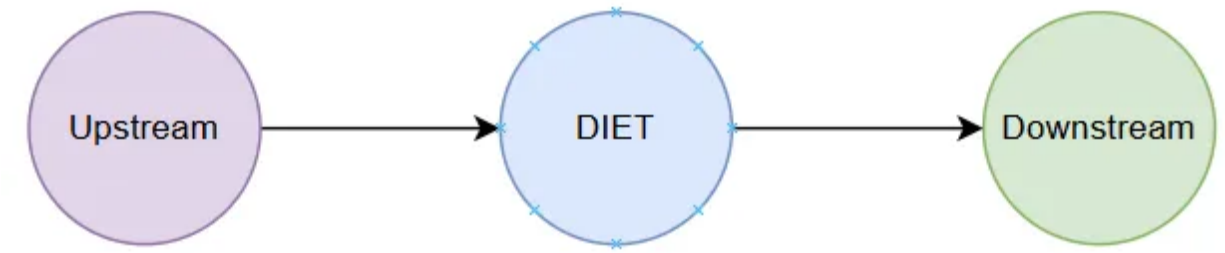
DIET (Data Ingestion and Extraction Tool) Guide

1. 概述

DIET (Data Ingestion and Extraction Tool) 是一个元数据驱动的ETL框架，旨在为部门内部各团队提供统一的数据接入和提取解决方案。它支持多种数据源和目标，包括文件系统(SFTP/HDFS)、API(RESTful)、数据库(JDBC)等，并通过配置化的方式实现灵活的数据处理流程。



2. 核心概念

2.1 主要组件

组件	描述
Feed	数据流的基本单位，代表一个完整的
Step	处理流程中的单个步骤，可组合成F
Flow	由多个Step组成的有序执行序列
Metadata	驱动框架运行的配置信息，存储在中

2.2 关键实体关系

Plain Text

```
Feed (1) → (*) Step
Step (*) → (1) Flow
Feed (*) ↔ (*) Configuration
```

3. 元数据配置详解

3.1 Feed基础配置 (Feed_Basic_Info)

字段名	类型	必填	描述
FEED_KEY	VARCHAR(50)	Y	唯一标识符
FEED_NAME	VARCHAR(100)	Y	父文件名
FEED_TYPE	ENUM	Y	类型: ZIP/GZ/CSV/EXCEL
SOURCE_TYPE	ENUM	Y	源类型: SFTP/API/JDBC/HDFS
SOURCE_APP_NAME	VARCHAR(20)	Y	上游应用
SOURCE_CONTACACT	VARCHAR(200)	Y	上游联系方式
DESTINATION_APP_NAME	VARCHAR(20)	Y	下游应用
DESTINATION_CONTACT	VARCHAR(200)	Y	下游联系方式
DESTINATION_TYPE	ENUM	Y	目标类型: DB/HDFS/API
ACTIVE_FLAG	BOOLEAN	Y	是否激活
SCHEDULE_CRON	VARCHAR(50)	N	调度表达式
HOLIDAY_PROFILE	VARCHAR(50)	N	节假日配置ID

3.3 Feed列信息表（Column_Info)

字段名	类型	必填	描述
SUB_FEED_ID	INT	Y	外键，关联 Sub_feed_info

Column_ID	INT	Y	唯一标识符
COLUMN_NAME	VARCHAR(100)	Y	列名
COLUMN_ORDER	INT	Y	列顺序号
COLUMN_TYPE	VARCHARE(50)	Y	列字段类型：int/varchar/date/datetime/numeric

3.4 子Feed配置表（Sub_Feed_Info)

字段名	类型	必填	描述
SUB_FEED_ID	INT	Y	唯一标识符
FEED_KEY	VARCHAR(50)	Y	外键，关联Feed_Basic_Info
SUB_FEED_NAME	VARCHAR(100)	Y	子文件名
CONTROL_FILE_NAME	VARCHAR(100)	Y	控制文件名称
FEED_TYPE	ENUM	Y	文件类型：dat/txt/csv/json/xls/parquet

3.5 步骤配置 (FEED_STEPS)

字段名	**类型**	**描述**	**示例**
STEP_ID	VARCHAR(50)	步骤唯一ID	VALI
STEP_NAME	VARCHAR(100)	步骤名称	"文件
STEP_TYPE	ENUM	步骤类型	VALI
STEP_ORDER	INT	执行顺序	2
CONFIG_REF	VARCHAR(50)	关联的配置ID	FILE

RETRY_COUNT	INT	重试次数	3
FAILURE_ACTION	ENUM	失败行为: STOP/SKIP/ALERT	STO

3.6 源配置示例 (SFTP_SOURCE_CONFIG)

SQL

```sql
CREATE TABLE SFTP_CONFIG (
    CONFIG_ID VARCHAR(50) PRIMARY KEY,
    HOST VARCHAR(100) NOT NULL,
    PORT INT DEFAULT 22,
    USERNAME VARCHAR(50),
    PASSWORD_REF VARCHAR(100), -- 引用密钥管理服务
    DIRECTORY_PATH VARCHAR(255),
    FILE_PATTERN VARCHAR(100),
    ENCRYPTION_TYPE ENUM('NONE','PGP','AES'),
    COMPRESSION_TYPE ENUM('NONE','ZIP','GZIP')
);
```

3.7 验证规则配置 (VALIDATION_RULES)

JSON

```json
{
  "ruleId": "DATE_VALIDATION_001",
  "ruleType": "DATE",
  "params": {
    "dateFormat": "yyyy-MM-dd",
    "position": "HEADER",
    "lineNumber": 1,
    "columnRange": "10-19",
    "expectedValue": "${COB_DATE}"
  },
  "errorAction": "REJECT"
}
```

4. 典型流程配置

4.1 SFTP文件处理流程

mermaid

```
graph TD
    A[Feed Monitor] --> B[File Unzip]
    B --> C[File Validation]
    C --> D[Data Loading]
    D --> E[File Archival]
```

配置示例：

SQL

```
INSERT INTO FEED_STEPS VALUES
('SFTP_SALES_001', 'MONITOR', 'MONITOR', 1, 'SFTP_MONITOR_CFG', 3, 'STOP'),
('SFTP_SALES_001', 'UNZIP', 'PROCESS', 2, 'UNZIP_CFG', 2, 'SKIP'),
('SFTP_SALES_001', 'VALIDATE', 'VALIDATION', 3, 'SALES_VALIDATION_CFG', 1,
```

4.2 API数据加载到大数据平台

mermaid

```
graph TD
    A[Data Readiness Check] --> B[API Loading]
    B --> C[HDFS Upload]
    C --> D[EAP Transform]
    D --> E[Data Archival]
```

配置示例：

JSON

```
{
  "apiConfig": {
    "endpoint": "https://api.sales.com/v1/data",
    "authType": "OAUTH2",
    "pagination": {
      "type": "OFFSET",
      "pageSize": 1000
    },
    "responseFormat": "JSON"
  },
  "hdfsConfig": {
    "path": "/data/lake/sales/${COB_DATE}",
```

```
    "format": "CSV",
    "partitionBy": ["region"]
  }
}
```

5. 使用指南

5.1 新Feed配置流程

1. 注册基础信息

SQL

```
INSERT INTO FEED_MASTER VALUES (
  'API_MONTHLY_INVENTORY',
  'Monthly Product Inventory',
  'INGESTION',
  'API',
  'HDFS',
  TRUE,
  '0 0 5 1 * ?',
  'GLOBAL_HOLIDAYS'
);
```

2. 定义处理步骤

SQL

```
INSERT INTO FEED_STEPS VALUES
('API_MONTHLY_INVENTORY', 'READINESS_CHECK', 'VALIDATION', 1, 'INV_READINES
('API_MONTHLY_INVENTORY', 'API_LOAD', 'PROCESS', 2, 'INV_API_CFG', 3, 'ALER
```

3. 配置API源

JSON

```
{
  "configId": "INV_API_CFG",
  "httpMethod": "GET",
  "queryParams": {
    "reportDate": "${COB_DATE}",
    "detailLevel": "FULL"
  },
  "pagination": {
```

```
    "type": "CURSOR",
    "nextPageField": "pagination.next"
  }
}
```

5.2 执行流程

通过Autosys调用：

Bash

```
#!/bin/bash
# diet_launcher.sh
FEED_KEY=$1
COB_DATE=$2
STEP_NAME=$3
java -jar diet-core.jar \
  --feed=$FEED_KEY \
  --date=$COB_DATE \
  --step=$STEP_NAME \
  --config=/path/to/config.properties
```

典型作业定义：

jil

```
insert_job: DAILY_SALES_LOAD
job_type: CMD
command: /opt/diet/bin/launcher.sh SFTP_DAILY_SALES ${COB_DATE} ALL
machine: diet-worker-01
date_conditions: 1
days_of_week: mo,tu,we,th,fr
```

6. 高级功能

6.1 数据质量检查

SQL

```
INSERT INTO DQ_RULES VALUES (
  'SALES_AMOUNT_CHECK',
  'COLUMN_RANGE',
  'SALES_FACT.AMOUNT',
  '{
```

```
    "min": 0,
    "max": 1000000,
    "threshold": 99.5,
    "action": "QUARANTINE"
  }'
);
```

6.2 错误处理机制

错误处理配置示例：

YAML

```yaml
errorHandling:
  maxRetries: 3
  retryInterval: 300000 # 5分钟
  notifications:
    - type: EMAIL
      recipients: ["team@company.com"]
    - type: SLACK
      channel: "#data-alerts"
  escalation:
    afterAttempts: 3
    contact: "oncall-engineer@company.com"
```

7. 监控与维护

7.1 关键监控指标

指标名称	类型	描述
diet.step.duration	Timer	步骤执行时间
diet.feed.success	Counter	成功运行次数
diet.validation.errors	Gauge	验证错误数
diet.records.processed	Counter	处理记录数

7.2 维护操作

元数据版本控制：

SQL

```sql
CREATE TABLE METADATA_VERSION (
  config_id VARCHAR(50),
  version INT,
  effective_date TIMESTAMP,
  config_content JSON,
  PRIMARY KEY (config_id, version)
);
```

(注：此处应包含完整的ER图，展示所有配置表及其关系)

附录B：典型错误代码

代码	含义	建议操作
DIET-4001	源文件未找到	检查SFTP连接
DIET-5003	API认证失败	验证令牌/证书
DIET-3008	验证规则不匹配	检查数据内容和
DIET-2002	目标存储空间不足	清理目标系统空