ICS 03.060
CCS A 11

Q/HXZQ

华 西 证 券 股 份 有 限 公 司 企 业 标 准

Q/HXZQ 007—2025

华西证券数据平台数据服务技术规范

Technical Specification For Data Services of HUAXI Data Platform

2025-12-01 发布　　2025-12-01 实施

华西证券股份有限公司　　发 布

目　　次

前　言

本文件按照GB/T　1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规范》的规定起草。

本文件由华西证券股份有限公司提出。

本文件由华西证券股份有限公司归口。

本文件起草部门：华西证券股份有限公司信息技术部。

本文件主要起草人：巩喜云、陈薇、刘光强、戢涛、黄红、罗朝新、常浩漾。

本文件及其所代替文件的历次版本发布情况为：

——2025 年首次发布为，Q/HXZQ 007—2025；

——本次为首次发布。

引　言

数据平台是证券期货行业企业数据治理的核心基础设施，承担企业数据汇聚、存储、计算、服务输出的核心职能，是数据价值落地的关键载体。

随着华西证券数字化转型深入，湖仓一体大数据中心建成投用，数据服务需求快速增长，为规范数据服务设计、开发、发布、调用、运维全流程，明确数据服务边界、性能、安全、权限要求，保障数据服务标准化、规范化、安全可控，制定本技术规范。

本文件为华西证券内部数据服务建设、系统对接、数据治理提供技术指导，同时符合证券期货行业数据模型、数据元相关监管规范，适用于公司所有数据服务相关工作。

华西证券数据平台数据服务技术规范

1 范围

本文件规定了华西证券湖仓一体数据平台数据服务的总体设计、服务方式、API开发、调用规范、安全权限、性能约束、运维管理、质量要求等内容。

本文件适用于华西证券内部数据服务建设、数据调用、系统对接、数据治理、运维管理相关所有工作，覆盖报表查询、管理分析、内部业务系统数据对接、监管数据支撑等场景。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 1.1—2020 标准化工作导则 第1部分：标准化文件的结构和起草规则

JR/T 0176.1—2019 证券期货业数据模型 第1部分：抽象模型设计方法

JR/T 0176.3—2021 证券期货业数据模型 第3部分：证券公司逻辑模型

JR/T 0304.1—2024 证券期货业基础数据元 第1部分：基础数据元

JR/T 0304.2—2024 证券期货业基础数据元 第2部分：基础代码

Q/HXZQ 001—2024 华西证券湖仓一体化大数据中心数据存储技术规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

数据平台 HUAXI Data Platform

基于湖仓一体大数据技术架构，整合数据湖、数据仓库、OLAP分析引擎能力，实现企业数据统一存储、计算、治理、服务输出的基础设施，支撑全公司数据需求与数字化应用。

3.2

数据服务 Data Service

数据平台将经过清洗、转换、建模后的可用数据（结果数据、集市数据、指标数据等），通过数据表推送、API调用、直连查询、文件导出等标准化方式，向业务系统、用户提供数据查询、分析、支撑的服务模式，是数据价值落地的核心载体。

3.3

API Application Programming Interface

数据服务对外暴露的标准化HTTP接口，支持JSON格式请求响应，提供单表分页查询能力，是内部系统数据交互的核心方式。

4 缩略语

下列缩略语适用于本文件。

API：应用程序编程接口（Application Programming Interface）
OLTP：联机事务处理（On-Line Transaction Processing）
OLAP：联机分析处理（On-Line Analytical Processing）
HTTP：超文本传输协议（Hypertext Transfer Protocol）
JSON：JavaScript对象表示法（JavaScript Object Notation）

5 总体设计

5.1 设计原则

数据服务遵循标准化、规范化、安全可控、性能适配、权责清晰的设计原则，确保数据服务统一、高效、安全、可运维。

5.2 服务定位

数据平台以StarRocks为核心OLAP存储引擎，聚焦分析型数据服务场景，为内部系统、管理分析、报表查询提供标准化数据获取通道；高并发OLTP联机事务、对客ToC业务禁止直接调用数据平台服务，需通过业务中台或数据前置缓存对接。

5.3 服务架构

数据服务采用“数据存储层-服务发布层-调用层”三层架构：
- 数据存储层：StarRocks库表，存储清洗后、可用的结果数据/集市数据；
- 服务发布层：API管理、权限控制、Token认证、流量管控；
- 调用层：内部业务系统、报表工具、管理平台等调用方。

6 数据服务方式

6.1 数据表推送

a) 定义：数据平台将定时/实时计算生成的结果数据、集市数据，批量推送到下游业务系统指定数据库表。
b) 适用场景：批量统计、报表数据、周期性管理指标等离线分析场景。
c) 要求：推送频率、数据格式、字段口径需双方对齐，推送数据需经过质量校验，确保一致性、准确性。

6.2 数据服务 API

a) 定义：基于 StarRocks 库表发布的标准化 HTTP 查询 API，支持分页查询。
b) 核心约束：
   - 仅支持单表点查、简单过滤，禁止复杂跨表、聚合、关联查询；
   - 单次请求分页返回，不支持全量数据导出；
   - 并发上限≤1000 QPS，避免影响平台整体性能；
   - 禁止对客 ToC、高并发 OLTP 场景直接调用。
c) 适用场景：内部管理系统、员工渠道、报表查询、轻量级业务数据对接。
d) 管理要求：API 按项目隔离，由项目管理员创建发布、运维负责人监控运维。

6.3 StarRocks 直连查询

a) 定义：下游申请只读账号，直接连接 StarRocks 数据库执行查询。
b) 约束：并发上限≤1000 Q，禁止高并发、对客场景使用。
c) 适用场景：内部分析、复杂报表、临时数据查询。
d) 管理：由运维负责人统一分配账号、管控权限、监控查询性能。

6.4 数据导出

a) 定义：下游提交导出申请，运维人员导出文件并上传至指定 FTP 服务器。
b) 适用场景：批量离线数据、跨部门数据共享。
c) 要求：导出需审批，文件格式统一，敏感数据脱敏处理。

7 数据服务 API 开发规范

7.1 API 创建要求

a) 基于 StarRocks 已建库表创建 API，需选择单表、明确查询字段、过滤条件。
b) 接口命名、字段命名遵循行业与公司标准，口径统一。
c) 分页参数默认，单次返回条数≤1000 条。

7.2 Token 权限管理

a) API 需绑定静态 Token，用于身份认证。
b) Token 需设置有效期、IP 白名单、访问权限，最小化授权。
c) Token 定期轮换，泄露立即吊销。

8 数据服务 API 调用规范

8.1 HTTP 调用示例

请求方式：POST
请求地址：http://
请求体：

```
{
  "ApiID": "$ApiID",
  "Option": [{"Id": 100, "Val": 0, "Val_": "$DPS_TOKEN"}],
  "Params": {"pageSize":0,"offset":0}
}
```

请求示例：

```
curl -X POST \
-H 'Content-Type: application/json' \
http:// \
-d   '{"ApiID":"$ApiID",
"Option": [{"Id":100,"Val":0,"Val_":"$DPS_TOKEN"}],
"Params": {"pageSize":0,"offset":0}}'
```

8.2 Java 调用示例

```
// 主函数
public static void main(String[] args) throws IOException {
        // 定义请求的 URL
        String baseUrl = "http://";

        // 创建 JSON 对象来表示 Option
        JSONObject option = new JSONObject();
        option.put("Id", 100); //定义 option 键值对
        option.put("Val", 0);
        option.put("Val_", "$DPS_TOKEN");

        // 创建 JSON 对象来表示整个 API 请求
        JSONObject apiRequest = new JSONObject();
        apiRequest.put("ApiID", "$ApiID"); //设置 ApiID
        apiRequest.put("Option", new JSONObject[]{option}); //将 Option 整体放入 ApiRequest
        apiRequest.put("Params", "{\"pageSize\":0,\"offset\":0}"); //将 Params 整体放入 ApiRequest

        // 执行 HTTP POST  请求
        performPostRequest(baseUrl, apiRequest.toString());
    }

    // 定义一个函数执行 HTTP POST 请求
    public static void performPostRequest(String url, String jsonInputString) throws IOException {
        // 创建一个 HTTP 客户端
        CloseableHttpClient client = HttpClients.createDefault();

        // 创建一个 HTTP POST 请求
        HttpPost httpPost = new HttpPost(url);

        // 创建一个实体协助发送 POST 数据，并设置实体
        StringEntity entity = new StringEntity(jsonInputString);
        httpPost.setEntity(entity);

        // 执行 POST 请求，并获取响应
        CloseableHttpResponse response = client.execute(httpPost);

        // 从响应中获取实体
        HttpEntity responseEntity = response.getEntity();

        // 如果响应实体存在，输出其内容
        if (responseEntity != null) {
```

```
            System.out.println(EntityUtils.toString(responseEntity));
        }

        // 关闭 HTTP 客户端
        client.close();
    }
```

9 安全与权限管理

a) 身份认证：所有 API 调用、直连查询需身份认证，禁止匿名访问。
b) 权限控制：最小权限原则，按用户/系统分配 API 访问权限、数据范围。
c) 数据脱敏：敏感数据（客户信息、交易数据）脱敏后输出。
d) 审计日志：所有调用、查询、导出操作留痕，日志留存≥6 个月。

10 运维管理与质量要求

a) 性能监控：实时监控 API 响应时间、并发量、错误率，异常告警。
b) 数据质量：接口数据口径一致、准确、完整，定期校验。
c) 变更管理：API 变更需评审、通知调用方，避免中断。

参考文献

[1] JR/T 0176.1—2019 证券期货业数据模型 第1部分：抽象模型设计方法
[2] JR/T 0176.3—2021 证券期货业数据模型 第3部分：证券公司逻辑模型
[3] JR/T 0304.1—2024 证券期货业基础数据元 第1部分：基础数据元
[4] JR/T 0304.2—2024 证券期货业基础数据元 第2部分：基础代码
[5] Q/HXZQ 001—2024 华西证券湖仓一体化大数据中心数据存储技术规范