自研配置中心微服务开发设计

（第一版）

郑琳

2023 年 8 月 5 日

目录

一、总体技术方案

1.1 总体架构设计

1.1.1 系统逻辑分层

为满足体育赛事大数据自动投注系统（以下简称投注系统）的模块化设计及以后的扩展需求，整个系统将被划分为公共组件、基础服务、业务模块即应用模块四层。

图 1-1 系统逻辑分层

1.1.2 系统物理分层

根据四层结构的设计，所有模块将被划分在不同的层面实现。

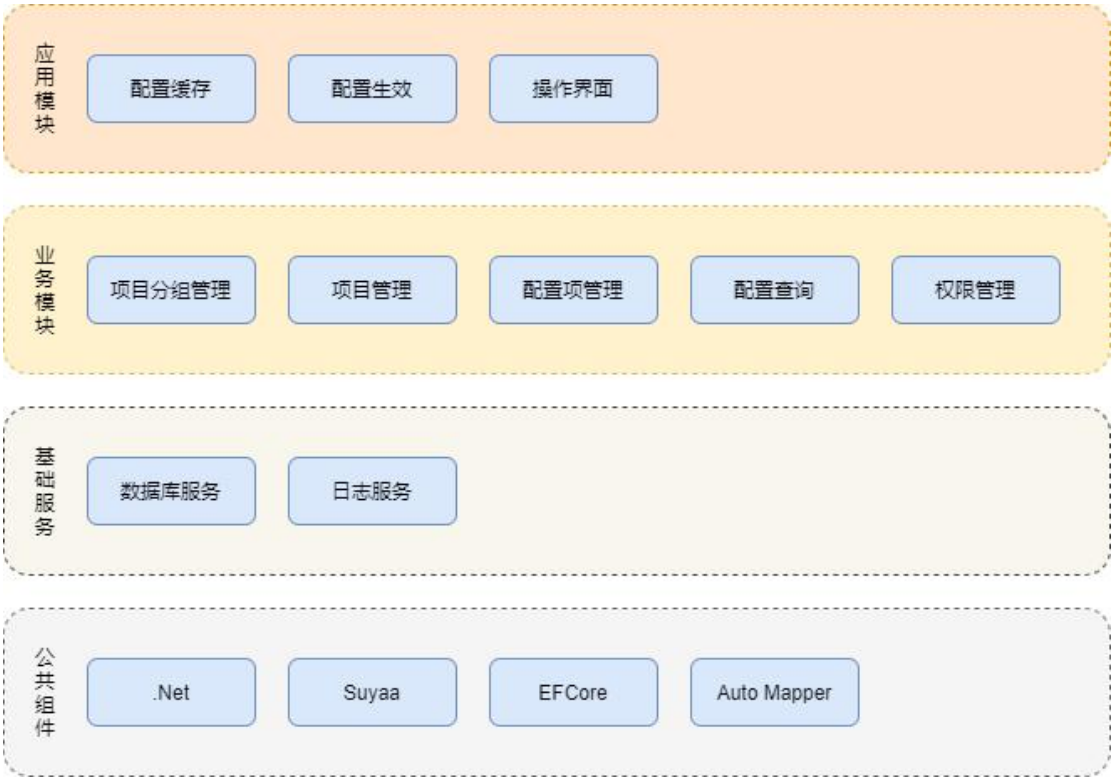

图 1-2 系统物理分层

1.2 网络架构设计

配置中心需要随时在不同客户端上进行配置修改，因此无需安装的 B/S 前后端分离架构优势就比较明显。

通过 B/S 结构的划分，我们将业务模块中的功能继承到服务端中，通过 API 将功能暴露给外部，Browser UI 端通过 API 进行基础信息的管理及配置的更新，SDK 端则访问 API 进行配置的查询与缓存处理。

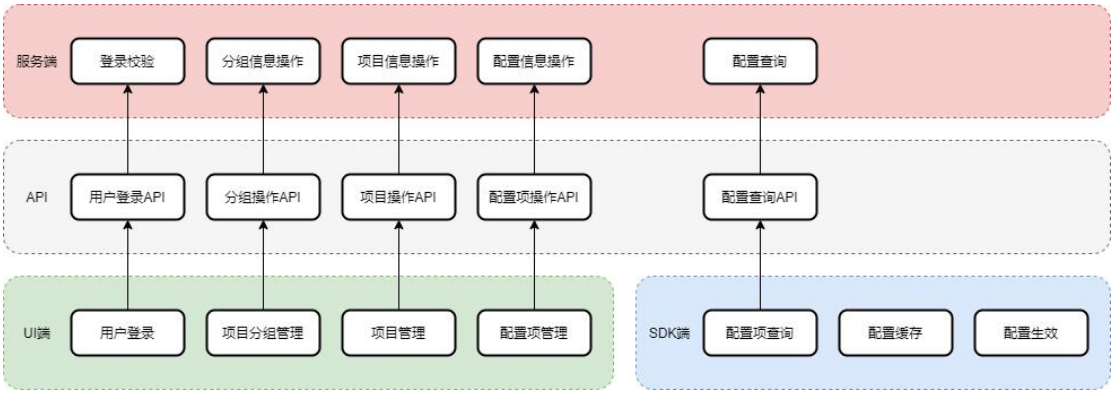

图 1-3 网络架构设计

1.2.1 服务端（Server）

配置中心的服务端负责提供以下服务：

- 登录信息校验
- API 身份安全校验
- 分组信息管理（添加、删除、修改和查询）
- 项目信息管理（添加、删除、修改和查询）
- 配置信息管理（添加、删除、修改和查询）

1.2.2 界面端（Browser UI）

配置中心的界面端负责提供以下服务：

- 用户登录
- 分组信息操作界面（添加、删除、修改和查询）
- 项目信息操作界面（添加、删除、修改和查询）
- 配置信息操作界面（添加、删除、修改和查询）

1.2.3 开发组件端（SDK）

配置中心的开发组件端负责提供以下服务：

- 配置项查询（轮询）
- 配置缓存操作（新建、更新、读取）
- 配置生效

1.3 开发语言及运行环境

1.3.1 服务端（Server）

开发语言：C#

运行环境：.Net 6

操作系统：Debian 12

数据库：Sqlite

技术栈：Asp.Net Core/EFCore/Auto Mapper/Suyaa(自研)

1.3.2 界面端（Browser UI）

开发语言：Html/Css/Javascript

运行环境：Chrome/Edge

操作系统：Windows 7/Windows 7+

技术栈：Vue

1.3.3 开发组件端（SDK）

开发语言：C#

运行环境：.Net Standard 2.1

操作系统：Windows 7/Windows 7+/Linux/MacOS

技术栈：Suyaa(自研)

二、基础数据模块设计方案

基础数据模块是为其他功能提供基础支持的模块。

基础数据模块主要包含项目基础信息的建立和用户权限控制两大块内容。

为尽量降低开发成本，用户信息直接采用配置文件方式进行定义。

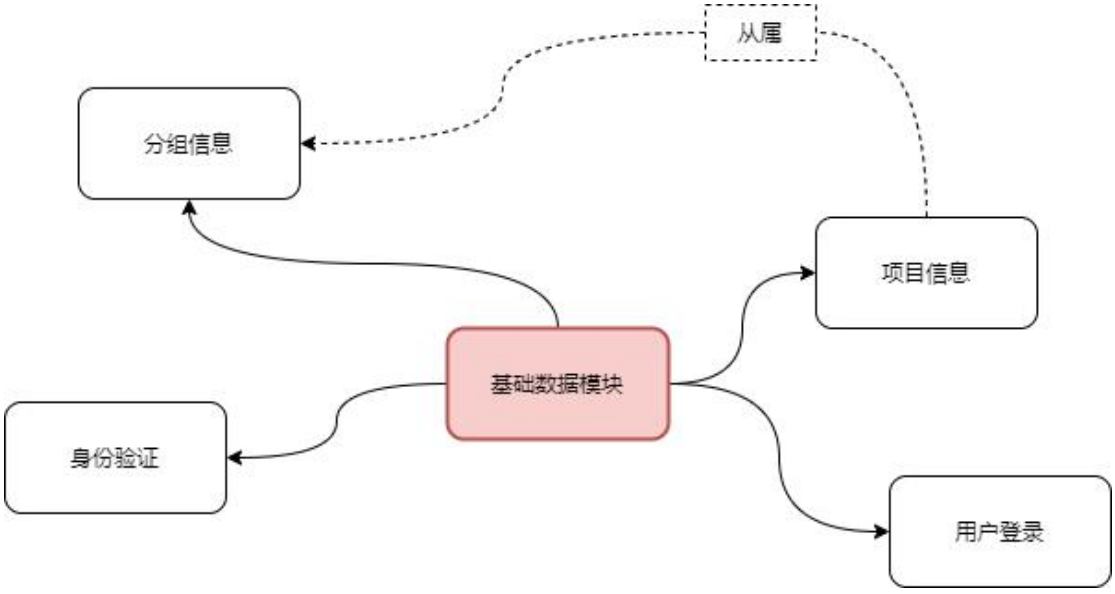

图 2-1 基础数据模块

2.1 数据建模

2.1.1 项目分组（project_catalog）

编号	字段名	数据类型	备注
1	id	varchar(50)	唯一标识
2	name	varchar(128)	名称
3	decription	varchar(512)	描述

2.1.2 项目信息（project）

编号	字段名	数据类型	备注
1	id	varchar(50)	唯一标识，主键
2	name	varchar(128)	名称

3	description	varchar(512)	描述
4	security_key	varchar(50)	安全密钥
5	catalog_id	varchar(50)	项目分组 Id

2.2 应用层级

所属模块：base/UI

2.3 开发项

编号	开发项	开发类型	备注
1	用户登录	API	提供用户登录接口
2	用户身份校验	Core	提供内部用户身份校验接口
3	项目分类信息管理	API	提供分类信息的列表、添加、修改、删除
4	项目信息管理	API	提供项目信息的列表、添加、修改、删除
5	用户登录界面	UI	设计用户登录界面，对接登录接口
6	分类操作界面	UI	设计分类操作界面，对接分类管理接口
7	项目操作界面	UI	设计项目操作界面，对接项目管理接口

三、配置存储模块设计方案

3.1 业务流程

配置存储模块支持以下功能：

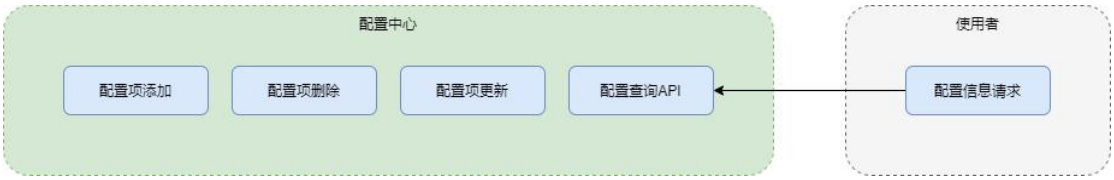

图 3-1 配置存储模块

- 配置的列表、添加、修改和删除；
- 可通过项目 Id、项目密钥信息，在无需登录的情况下，提供配置的独立查询接口。

3.2 数据建模

3.2.1 配置（config）

编号	字段名	数据类型	备注
1	id	varchar(50)	唯一标识
2	name	varchar(128)	配置名称
3	value	text	配置值
4	description	varchar(512)	描述
5	project_id	varchar(50)	项目 Id

3.3 应用层级

所属模块：cfg/UI

3.4 开发项

编号	开发项	开发类型	备注
1	配置管理	API	提供配置信息的列表、添加、修改和删除

2	配置独立查询	API	提供配置信息的查询
3	配置管理界面	UI	设计项目操作界面，对接配置管理接口

四、配置缓存模块设计方案

4.1 业务流程

配置缓存模块业务流程如下：

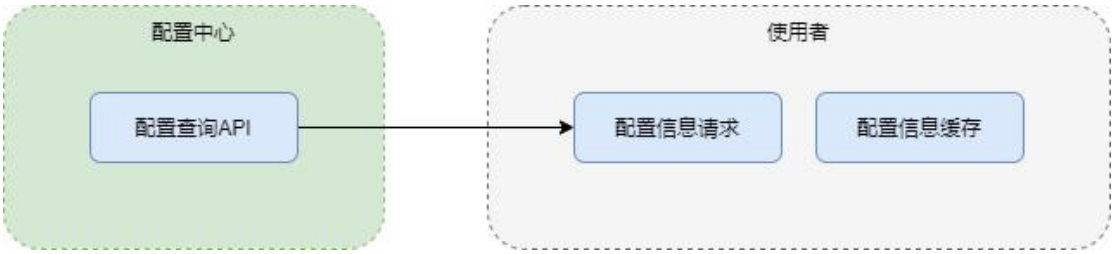

图 4-1 配置缓存模块上下文关系

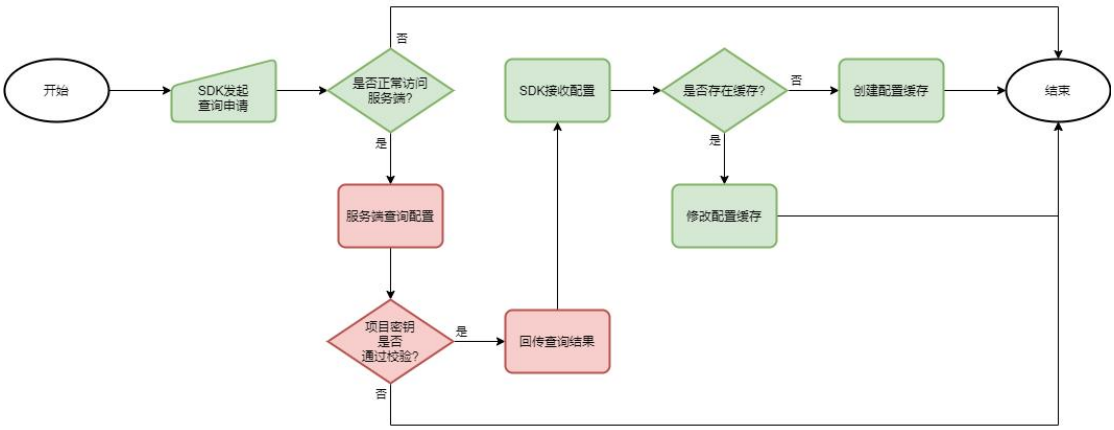

图 4-2 配置缓存流程

- SDK 端发起查询请求，如果查询失败，则退出操作流程，反之继续下一步；
- 服务端接收到查询请求后，先进行密钥校验，校验未通过，则退出操作流程，反之继续下一步；
- 服务端返回查询结果；
- SDK 接收到查询结果后，判断本地是否存在缓存文件，如不存在，则创建缓存文件，反之，则进行缓存文件的更新。

4.2 应用层级

所属模块：SDK

4.3 开发项

编号	开发项	开发类型	备注
1	查询接口对接	SDK	发起查询并获取查询结果
2	缓存操作	功能	缓存文件的新建与更新

五、配置应用模块设计方案

5.1 业务流程

配置存储模块支持以下功能：

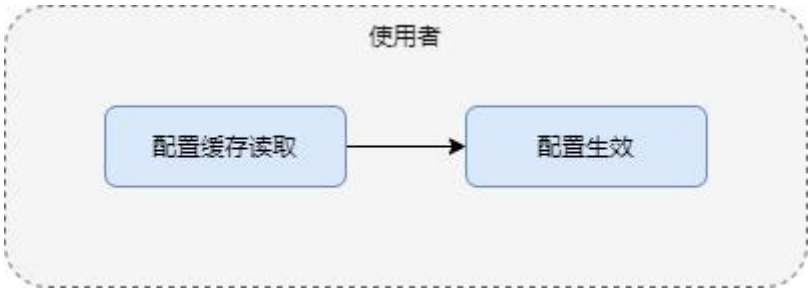

图 5-1 配置应用模块

- 从缓存文件中读取配置信息；
- 将配置信息对接 Asp.Net Core 配置集合。

5.2 应用层级

所属模块：SDK

5.3 开发项

编号	开发项	开发类型	备注
1	配置读取	功能	从配置缓存文件中读取信息
2	配置生效	功能	将配置信息对接 Asp.Net Core 框架