IEC客户端和服务端交互

1. 服务端

string IECDataProcess(const Document& docObj, std::shared\_ptr<ICfgSvrDatabaseClient> pDBInterface)为 IEC模块接口总入口

{DB\_QSUBTYPE\_POU\_NEW\_CHECK, PouNewCheck},新建POU检查DB检查接口

{DB\_QSUBTYPE\_POU\_PRE\_ADD, PouPreAdd},准备添加POU，在数据库上将UUID和名称登记上，保存时正式添加并补充其他信息

{DB\_QSUBTYPE\_POU\_PRE\_MOD, PouPreMod},准备编辑POU，在数据库上设置对应的编辑标志

{DB\_QSUBTYPE\_POU\_GET\_ALL, PouGetAll},获取所有POU的基本信息

{DB\_QSUBTYPE\_POU\_GET\_UUID, PouGetUUID},获取指定POU的基本信息

{DB\_QSUBTYPE\_POU\_GET\_DATA, PouGetData},获取指定POU的组态逻辑信息

{DB\_QSUBTYPE\_POU\_SAVE,PouSave},保存POU的基本信息(目前包含组态信息)

{DB\_QSUBTYPE\_POU\_SAVE\_DATA, PouSaveData},保存POU的逻辑组态信息

{DB\_QSUBTYPE\_POU\_MODIFY\_NAME\_DATA, PouModifyName},修改POU名称

{DB\_QSUBTYPE\_POU\_FOLDER\_NEW\_CHECK, PouFolderNewCheck},新建POU文件夹检查DB检查接口

{DB\_QSUBTYPE\_POU\_FOLDER\_ADD, FolderAdd},添加文件夹基本信息

{DB\_QSUBTYPE\_POU\_FOLDER\_MOD, FolderMod},准备编辑文件夹，在数据库上设置对应的编辑标志以及修改其他信息

{DB\_QSUBTYPE\_POU\_FOLDER\_GET\_ALL, FolderGetAll},获取所有POU文件夹的基本信息

{DB\_QSUBTYPE\_POU\_FOLDER\_GET\_UUID, FolderGetUUID},获取指定POUFolder的基本信息

{DB\_QSUBTYPE\_POU\_FOLDER\_SAVE, FolderSave},保存Folder的基本信息

{DB\_QSUBTYPE\_POU\_FOLDER\_MODIFY\_NAME\_DATA, FolderModifyName},修改文件夹名称

{DB\_QSUBTYPE\_IEC\_CLEAR\_STATUS\_DATA, IECClearStatusData},清空工程本身的离线组态状态数据

以上为数据库指令接口及说明

1. 客户端
2. virtual bool Load() override;从数据库加载数据，获取文件夹及pou数据
3. 从数据库获取文件夹数据
4. 从数据库获取pou数据
5. 组织文件夹和pou关系树
6. virtual bool Save() override;保存数据到数据库，文件夹及pou数据
7. 保存文件夹数据，只保存修改过的以及新增的
8. 保存pou数据，只保存修改过的pou数据

3、virtual void LoadAfter() override;所有数据加载完成后处理业务接口

1、处理关联变量

4、virtual bool Sync() override;同步数据接口

1、先保存pou数据

2、获取数据库所有文件夹数据，和本地数据比较，删除以及删除的文件夹，添加新增的文件夹，处理修改过的文件夹

3、文件夹节点移动

4、获取数据库所有pou数据信息

5、删除已经删除的pou，添加新增的pou，修改已经被修改的pou

5、void UpdateData(const map<string, ChangeTypeInfo>& mapChangeType, const bool& isNeedFindType = true);类型更新接口

1、调用所有pou VarTypeUpdateData（）类型更新接口，调用所有单元件类型更新接口