**总体结构**

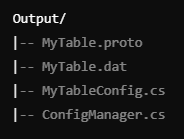
1. **核心工具链**：
   * **Excel 配表**：作为输入源，描述游戏功能相关的数据。
   * **打表工具**：负责将 Excel 数据解析成 .proto 和 .dat 文件，生成游戏代码（C# 脚本）。
   * **C# 配表脚本 (XXXConfig.cs)**：
     + 定义每张配表的管理逻辑。
     + 提供强类型的接口访问配表数据。
   * **配置管理器 (ConfigManager.cs)**：
     + 管理所有配表脚本。
     + 加载 .dat 文件并初始化数据。
     + 提供全局访问入口（通过单例或 DI 容器）。
   * **统一基类**：
     + 定义一个 BaseConfig，所有表的配置类（XXXConfig）继承自该基类。
     + 提供通用的加载逻辑。
   * **动态发现表配置类**：
     + 利用反射动态加载所有继承 BaseConfig 的类。
     + 自动调用 Load 方法加载数据。
   * **便捷的全局访问**：
     + ConfigManager 提供统一的访问接口，支持按类型获取特定的表配置。
2. **程序运行逻辑**：
   * **游戏启动时**：
     + ConfigManager 加载所有 .dat 文件，将数据分发到对应的 XXXConfig 中。
   * **运行时**：
     + 游戏逻辑通过 ConfigManager 或单例接口访问 XXXConfig 提供的配表数据。

**详细实现步骤**

**1. 打表工具的扩展**

**生成 .proto 和 .dat 文件**

当前工具已经能够解析 Excel 配表并生成 .proto 和 .dat 文件。需要确保每张 Excel 配表都有唯一的文件名，生成的文件结构类似：



**生成 C# 配表脚本**

为每张 Excel 表生成一个对应的 XXXConfig.cs，脚本功能包括：

* 定义与 .proto 文件对应的强类型数据结构。
* 提供加载 .dat 文件的功能。
* 处理表中数据的查询、过滤、校验等逻辑。