

```
src/
├── core/ # 核心功能
│   ├── cell.ts # 单元格对象
│   ├── structure.ts # 表结构管理
│   ├── sheet.ts # 工作表树
│   ├── commit.ts # 提交对象
│   ├── table-git.ts # 主版本控制引擎
│   ├── diff-merge.ts # 差异比较和合并
│   └── conflict-resolver.ts # 冲突解决
├── formatters/ # 函数式格式化器
│   ├── types.ts # TableData/FormatterFunction 类型
│   ├── adapter.ts # TableDataAdapter (统一数据抽象)
│   ├── function-formatter.ts # FunctionFormatter/Registry
│   └── builtin.ts # csv/json/html 格式函数
├── types/ # 类型定义
│   └── index.ts
├── utils/ # 工具函数
│   ├── hash.ts # 哈希和工具函数
│   └── factory.ts # 便利创建函数
└── index.ts # 主入口文件
```

table-git

面向表格数据的 Git 风格版本控制工具包，支持单元格与表结构的细粒度历史追踪。

✦ 特性速览

- 单元格级与结构级版本控制
- 分支、提交、合并与差异对比
- 冲突检测与自动/手动解决
- CSV / JSON / HTML 等格式化输出
- TypeScript 声明文件开箱即用

📦 安装与使用

```
npm install table-git
```

TypeScript 项目直接导入即可：

```
import { createTableGit, createColumn } from 'table-git';

const repo = createTableGit('main');
repo.addColumn('default', createColumn('product_name', { dataType: 'string',
```

```
order: 0 }));  
repo.addCellChange('default', 1, 0, 'iPhone 15');  
repo.commit('初始化产品表', 'Alice', 'alice@example.com');
```

文档目录

- [概览](#)
- [快速开始](#)
- [API 参考](#)
- [格式化器体系](#)
- [高级主题](#)

测试 & 构建

```
npm test      # 运行测试  
npm run build # 生成 dist/
```

所有发布版本都包含编译输出与类型声明，可直接用于 Node.js 或前端构建环境。

贡献指南

欢迎通过 Issue 或 Pull Request 提交改进建议。请在提交前运行测试并附带说明性文档或用例。

许可证

MIT License