

# DataManager

主要是写了

datamanager.h

datamanager.cpp

localtest.qml (放进项目后可以跑这个测试一下)

记得在main.cpp里加两行:

```
#include "DataManager.h"
```

.....

```
engine.rootContext()->setContextProperty("dataManager", dataManager); //在viewModel后面
```

.....

方法	作用
saveData(storyData: QVariantMap, fileName: QString)	将 storyData 转成 JSON 并保存到本地
loadData(fileName: QString)	从本地读取 JSON 并返回 QVariantMap
clearData(fileName: QString)	删除指定 JSON 文件

所有文件都保存在:

```
./data/<fileName>
```

例如:

```
./data/story.json
```

```
./data/scene_23.json
```

## 📌 2. 如何调用 (示例)

在任意 QML 文件中:

```
var story = {
```

```
"storyId": "abc-001",
"title": "测试故事",
"scenes": []
}
dataManager.saveData(story, "story.json")
```

加载：

```
var loaded = dataManager.loadData("story.json")
console.log(JSON.stringify(loaded, null, 2))
```

删除：

```
dataManager.clearData("story.json")
```

## 📌 3. QVariantMap = QML 中的 JS Object

在 QML 中：

```
var obj = {
    "id": 1,
    "title": "Hello"
}
dataManager.saveData(obj, "a.json")
```

这个 JS Object 在内部会自动变成 QVariantMap。

你不需要做任何转换，只用写普通 JS 对象即可。

## 📌 4. 后端返回内容 → 本地保存内容的映射规则

后端可能会返回以下数据种类：

- **创建项目**返回 projectId
- **生成分镜**返回 shots
- **生成单帧**返回 prompt / finalImageUrl 等
- **生成视频**返回 videoPath

所有这些数据最终都要合并进一个 storyData：

## ✓ storyData 标准结构 (推荐)

```
{  
  "storyId": "string",  
  "title": "string",  
  "description": "string",  
  
  "author": {  
    "id": "string",  
    "name": "string"  
  },  
  
  "createTime": "string",  
  "updateTime": "string",  
  
  "scenes": [  
    {  
      "sceneId": 1,  
      "title": "场景标题",  
      "summary": "描述",  
      "frames": [  
        {  
          "frameId": "1-1",  
          "prompt": "提示词",  
          "aspectRatio": "16:9",  
          "thumbUrl": "http://...",  
          "finalImageUrl": "http://...",  
          "isApproved": true  
        }  
      ]  
    }  
  ],  
  
  "video": {  
    "exported": false,  
    "videoUrl": "",  
    "thumbnail": ""  
  }  
}
```

## 📌 5. 前端应该在什么时候调用 saveData?

建议的流程如下：

### ✓ 1) 创建故事成功 (后端返回新 projectId)

保存：

```
storyData.storyId = result.projectId  
dataManager.saveData(storyData, "story.json")
```

### ✓ 2) LLM 返回分镜 (sequences, shots)

合并、保存：

```
storyData.scenes = result.scenes  
dataManager.saveData(storyData, "story.json")
```

### ✓ 3) 修改 prompt 或重新生成某帧

更新：

```
storyData.scenes[0].frames[1].prompt = newPrompt  
dataManager.saveData(storyData, "story.json")
```

### ✓ 4) 生成图片 / 视频成功

更新字段：

```
storyData.scenes[1].frames[0].finalImageUrl = url  
dataManager.saveData(storyData, "story.json")
```

### ✓ 5) 视频导出成功

```
storyData.video = {  
  "exported": true,  
  "videoUrl": url,  
  "thumbnail": thumb}
```

```
}
```

```
dataManager.saveData(storyData, "story.json")
```