



浙江大学 Beamer 模板

——这里是副标题

光头哥

chenqiyuan1012@foxmail.com

<https://www.zju.edu.cn/>

计算机学院
浙江大学

2023 年 8 月 31 日

提纲

提纲

介绍

内置环境

DeSci 平台端到端演示

结论

参考文献



介绍

- 编译方式
 - 推荐安装完整版的 TeXLive
 - 使用 Xe^ATeX 编译
- 请参考 L^AT_EX 和 Beamer 用户文档
- 行内数学公式示例 $\sin^2 \theta + \cos^2 \theta = 1$
- 行间数学公式示例

$$y_1 = \int \sin x dx$$

- 基于“浙大蓝”颜色 <https://www.zju.edu.cn/>



内置环境

Slides with \LaTeX

Beamer offers a lot of functions to create nice slides using \LaTeX .

The basis

内部使用以下主题

- split
- whale
- rounded
- orchid

带数字列表

- ❶ This just shows the effect of the style
- ❷ It is not a Beamer tutorial
- ❸ Read the Beamer manual for more help
- ❹ Contact me only concerning the style file



完整演示流程 - 链上写链下读 + 数据库集成

演示步骤

- ① 启动环境 - 部署 10 个合约到 Hardhat
- ② 后端服务 - Go 服务连接 SQLite+ 事件监听
- ③ 链上交互 - 用户注册、发表研究成果 NFT
- ④ 链下同步 - 捕获事件并入库 SQLite
- ⑤ 数据库验证 - 查询验证数据完整性
- ⑥ API 验证 - RESTful 接口获取数据

[端到端架构图]

智能合约 → 事件监听 → SQLite → API

演示重点

完整展示链上写、链下读与数据库真实集成的 DeSci 平台

步骤 1: 环境启动与合约部署

启动命令

```
$ npm run start-hardhat
$ npm run deploy-contracts
```

验证要点

- Hardhat 网络 (Chain ID: 31337) 启动成功
- 10 个智能合约全部部署完成
- 合约地址输出并记录到配置文件

[Hardhat 部署日志]

```
DeSciRegistry: 0x5FbDB...
DatasetManager: 0xe7f17...
ResearchNFT: 0x9fE46...
...10 个合约部署完成
```

步骤 2: 后端服务启动与数据库连接

启动命令

```
$ cd backend && go run cmd/server/main.go
```

验证要点

- SQLite 数据库连接成功
- 区块链事件监听器启动
- HTTP API 服务器运行 (端口 8080)
- 合约地址配置加载完成

[Go 后端启动日志]

```
SQLite connected
Event listeners started
Server running on :8080
```


步骤 3: 链上交互与事件触发

演示脚本

```
$ npx hardhat run scripts/  
fullTestScenario.js --network  
localhost
```

模拟操作

- 用户注册 → UserRegistered 事件
- 上传数据集 → DatasetUploaded 事件
- 发表研究成果 → ResearchMinted 事件

[Hardhat 交易日志]

```
Transaction: 0xabc123...  
ResearchMinted(tokenId: 1)  
Gas Used: 234,567
```

步骤 4: 链下事件同步与数据库写入

事件监听验证

Go 后端实时捕获链上事件并入库 SQLite 数据库

同步过程

- 监听到 ResearchMinted 事件
- 提取事件参数: tokenId, author, contentHash
- 插入到 research_data 表
- 记录区块号和交易哈希

[事件同步日志]

```
Event Captured: ResearchMinted
Inserting to SQLite...
Database sync completed
```

步骤 5: SQLite 数据库验证

数据库查询

直接查询 SQLite 验证数据完整性

SQL 查询命令

```
SELECT * FROM research_data  
WHERE token_id = '1';
```

[SQLite 查询结果]

```
token_id: 1  
title: 区块链科研应用  
authors: ["0x123..."]  
block_number: 12345  
tx_hash: 0xabc123...
```

步骤 6: RESTful API 验证与数据一致性

API 测试

通过 REST API 获取刚才上链的数据

测试命令

```
curl -X GET
"http://localhost:8080/api/v1/
research/1"
```

[API 响应结果]

```
{
  "tokenId": "1",
  "title": "  区块链科研应用",
  "authors": ["0x123..."],
  "contentHash": "0xabc...",
  "blockNumber": 12345
}
```

演示总结 - 数据库真实集成验证

演示完成验证清单

- 环境部署 - 10 个智能合约成功部署
- 数据库连接 - SQLite 真实连接并建表
- 链上交互 - 用户注册、研究成果发表
- 事件同步 - 实时捕获并入库 SQLite
- 数据一致性 - 链上链下数据完全一致
- API 验证 - RESTful 接口获取数据库数据

核心价值体现

完整展示了链上写、链下读与数据库真实集成的 DeSci 平台架构!

[数据库集成架构]




智能合约 → 事件监听 → SQLite →
API

结论

- Easy to use
- Good results



参考文献

-  Yi Zhao, *An introduction to X*, Sep. 15, 2015
-  Er Qian, San Sun, *Phys. Lett. A* **xx**, 2xx (20xx)
-  Si Li, *Phys. Rev. C* **xx**, 5xx (20xx)

