

Oxcafe.fun – MEME 代币发射平台

一个完整的去中心化 MEME 代币创造与交易平台，采用 **Bonding Curve** 机制实现公平价格发现、自动毕业系统和智能流动性管理。

—核心特性—

Bonding Curve 交易机制

- **动态定价算法**: 价格随购买量增长，防止操控
- **公平价格发现**: 早期支持者获得更低价格
- **自动流动性**: 无需人工做市，算法自动提供流动性

自动毕业系统

- **市值毕业门槛**: 达到 10 ETH 市值自动毕业
- **智能流动性管理**: 自动添加流动性到 Uniswap V2
- **永久锁定**: 毕业后流动性永久锁定，防止 Rug Pull
- **完全去中心化**: 毕业后放弃所有权限，实现真正去中心化

Vanity 地址生成

- **个性化地址**: 生成以 “Oxcafe” 开头的合约地址
- **CREATE2 技术**: 预计算地址，确保唯一性
- **高速生成**: 本地算法，平均 10,000+ 次/秒计算速度

费用分配机制

- **平台费用**: 2% 交易手续费
- **创建者分成**: 3% 交易收益分成
- **可持续发展**: 为平台和创建者提供持续收益

-项目目录-

0xcafedotfun (Meme Token 发射平台)

0xcafe.fun/	
├── src/	# 智能合约层
│ ├── MemeToken.sol	# ERC20 代币合约
│ ├── MemeFactory.sol	# CREATE2 工厂合约
│ ├── MemePlatform.sol	# 平台管理合约
│ ├── BondingCurve.sol	# 交易和毕业逻辑
│ ├── LiquidityManager.sol	# 流动性管理合约
│ └── FeeManager.sol	# 费用管理合约
├── frontend/	# Web 应用层
│ ├── src/app/	# Next.js 14 页面路由
│ ├── src/components/	# React UI 组件
│ ├── src/hooks/	# 自定义 React Hooks
│ └── src/config/	# Web3 配置
├── backend/	# 服务层
│ ├── src/services/	# 核心服务
│ │ ├── liquidityMonitor.ts	# 流动性监控服务
│ │ ├── blockchain.ts	# 区块链交互服务
│ │ └── cache.ts	# 缓存服务
│ ├── src/routes/	# API 路由
│ └── src/clients/	# 外部客户端
├── script/	# 部署脚本
├── shell/	# 工具脚本
└── test/	# 测试文件

0xcafe-uniswapV2-dex (基于 uniswap v2 实现 DEX)

 核心架构																	
 智能合约																	
<div>contracts/</div> <table><tr><td>├──  UniswapV2Factory.sol</td><td># 工厂合约 - 创建交易对</td></tr><tr><td>├──  UniswapV2Router02.sol</td><td># 路由合约 - 用户交互入口</td></tr><tr><td>├──  UniswapV2Pair.sol</td><td># 交易对合约 - 管理流动性</td></tr><tr><td>├──  MockWETH.sol</td><td># WETH 合约 - 本地测试用</td></tr><tr><td>└──  MockERC20.sol</td><td># 测试代币合约</td></tr></table>		├──  UniswapV2Factory.sol	# 工厂合约 - 创建交易对	├──  UniswapV2Router02.sol	# 路由合约 - 用户交互入口	├──  UniswapV2Pair.sol	# 交易对合约 - 管理流动性	├──  MockWETH.sol	# WETH 合约 - 本地测试用	└──  MockERC20.sol	# 测试代币合约						
├──  UniswapV2Factory.sol	# 工厂合约 - 创建交易对																
├──  UniswapV2Router02.sol	# 路由合约 - 用户交互入口																
├──  UniswapV2Pair.sol	# 交易对合约 - 管理流动性																
├──  MockWETH.sol	# WETH 合约 - 本地测试用																
└──  MockERC20.sol	# 测试代币合约																
<div> 前端应用</div> <div>frontend/</div> <table><tr><td>├── src/app/</td><td># Next.js 页面</td></tr><tr><td>│ ├── swap/</td><td># 代币交换页面</td></tr><tr><td>│ └── add-liquidity/</td><td># 添加流动性页面</td></tr><tr><td>├── src/components/</td><td># React 组件</td></tr><tr><td>├── src/config/</td><td># 配置文件</td></tr><tr><td>│ ├── addresses.ts</td><td># 合约地址配置</td></tr><tr><td>│ └── abis.ts</td><td># 合约 ABI</td></tr><tr><td>└── src/hooks/</td><td># 自定义 Hooks</td></tr></table>		├── src/app/	# Next.js 页面	│ ├── swap/	# 代币交换页面	│ └── add-liquidity/	# 添加流动性页面	├── src/components/	# React 组件	├── src/config/	# 配置文件	│ ├── addresses.ts	# 合约地址配置	│ └── abis.ts	# 合约 ABI	└── src/hooks/	# 自定义 Hooks
├── src/app/	# Next.js 页面																
│ ├── swap/	# 代币交换页面																
│ └── add-liquidity/	# 添加流动性页面																
├── src/components/	# React 组件																
├── src/config/	# 配置文件																
│ ├── addresses.ts	# 合约地址配置																
│ └── abis.ts	# 合约 ABI																
└── src/hooks/	# 自定义 Hooks																

-功能过程演示-

1. 创建 Token

(Token 地址=0xcafe9328d9ca0e574704967e88eed656e69711e5)

- 通过 CREATE2 实现“0xcafe”开头 Token 合约地址（在前端计算）

- 初始化 Token 信息（用于 Bonding Curve 交易机制计算）

- **初始价格:** 0.0000001 ETH
- **目标价格:** 0.001 ETH
- **目标供应量:** 100,000,000 代币（仅用于价格计算）
- **毕业条件:** 市值达到 10 ETH
- **费用结构:** 平台费用 2%，创建者分成 3%

0xcafe.fun

创建代币

交易市场

9,992.2 ETH

0x3C...93BC

基础信息

代币名称

Meme Token

代币符号

MEME

小数位数

18

目标供应量

100,000,000 代币

Bonding Curve 参数 & 毕业机制

初始价格: 0.0000001 ETH

目标价格: 0.001 ETH

目标供应量: 100,000,000 代币 (仅用于价格计算)

毕业条件: 市值达到 10 ETH

费用结构: 平台费用 2%, 创建者分成 3%

代币描述

MEME 代币描述

代币图标

https://example.com/token-image.png

可选: 输入图片URL, 留空将使用默认图标

Vanity 地址生成

cafe前缀地址生成

使用本地CREATE2算法生成以 'cafe' 开头的个性化代币合约地址, 计算速度更快, 让你的代币更具辨识度。

生成 cafe 开头地址

生成成功!

地址:

0xcafe9328d9ca0e574704967e88eed656e69711e5

Salt:

0x5c7656afc0c73c64112cfc30f15cf9b368c76d98094f70449dcec42033424f3a

尝试次数: 29,086

将使用此地址

合约信息

平台:

0xe6e340d132b5f46d1e472debc681b2abc16e57e

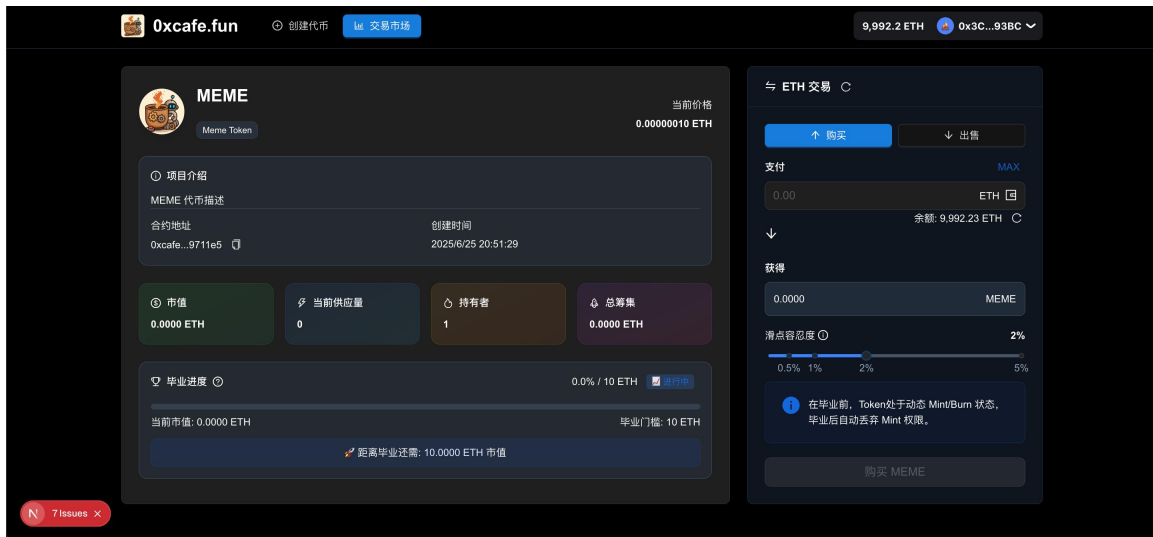
工厂:

0x67d269191c92caf3cd7723f116c85e6e9bf55933

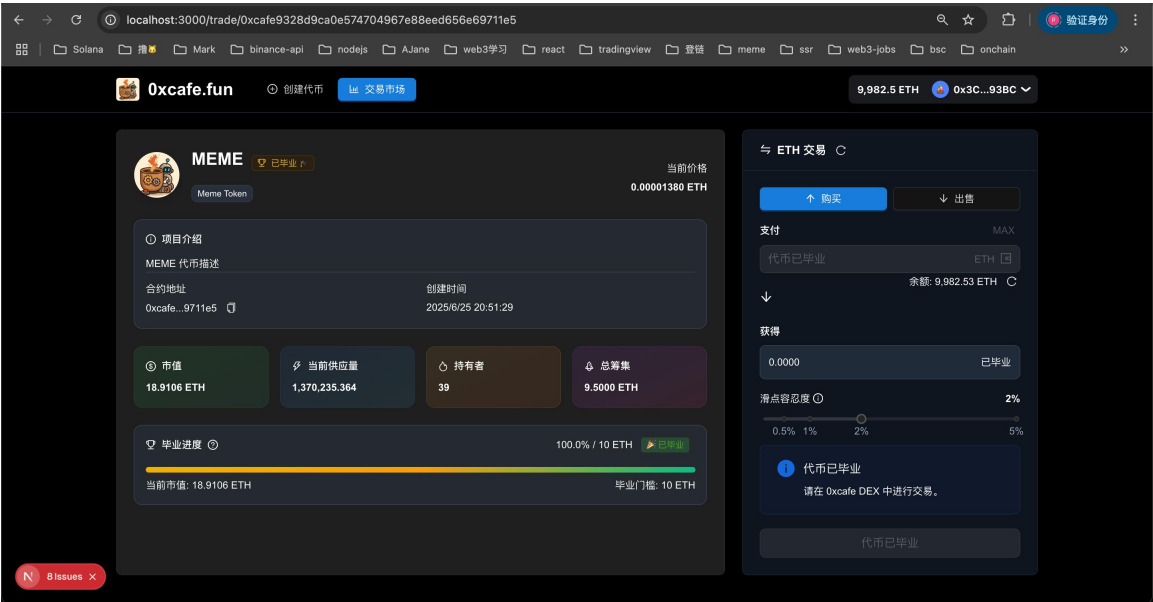
网络: Anvil

2. 购买 Token（市值达到 10ETH 自动毕业）

– 当前购买 Token Swap 使用 Bonding Curve 交易机制



3. Token 自动毕业（执行添加流动性并丢弃 Minter 权限）



4. 流动性自动添加成功

- nodejs 后端服务通过监听事件触发，去执行合约中添加流动性并丢弃 Minter 铸币权限操作
- 此时在 Oxcafe DEX 中 Token Swap 转换为 AMM 模式

