

Web3后端学员面试阶段标准

总体要求

技能可量化 —— 能够验证学员是否真的具备完成实际任务的能力。

项目可展示 —— 学员需要有拿得出手的Demo或代码库，作为作品展示给面试官。

Web3 求职面试标准（后端+合约方向）

基础知识标准

学员必须能够清晰回答以下问题（口述/笔试均可）：

- 区块链核心概念：区块、交易、Merkle Tree、共识机制（PoW、PoS）、Gas Fee 原理。

补充：

- BTC UTXO账户模型的理解。
- 以太坊账户模型的理解，合约账户、EOA账户认识。
- 以太坊交易的数据结构及EIP1559/712协议的理解。
- 以太坊的账户是如何生产的，公私钥的关系。
- 以太坊区块链浏览器一般会展示哪些信息，大概能知道。
- 以太坊交易nonce的作用，nonce异常交易如何处理。
- 以太坊交易inputdata一般放什么数据。
- 以太坊的地址多长，有几种地址创建方式。
- 以太坊的事件包含哪些内容，浏览器上事件如何分析。
- 以太坊和比特币快速交易的方式分别是什么。
- 作为扩展，对solana的共识机制POH，账户模型以及交易流程是否熟悉。
- 钱包 & 签名：解释私钥、公钥、助记词，能用代码生成签名 & 验证签名。
- 智能合约基础：合约生命周期、事件、modifier、storage vs memory 的区别。

补充：

- calldata是什么含义，什么情况下使用calldata。
- 数组与map存储的有啥区别，在存储槽中是连续的么。
- 合约安全是如何保障的，日常开发有哪些考虑。

- solidity gas费优化是如何做的，有哪些措施。
- 什么是ABI编码,怎么通过ABI编码调用合约方法。
- call与delegatecall的区别。
- topic的理解，indexed可以定义多少个。
- erc20和erc721区别，多了哪些函数。
- bool类型占用多少字节，合约存在大量“bool”数据，如何做优化。
- 发行图文并茂的erc721
- 常见安全问题：重入攻击、整数溢出、权限控制、随机数等问题。
- golang基础知识

补充：

- golang基本数据类型有哪些。
- interface接口类型是如何实现的，作为value，其原理是什么。
- 数组和切片(slice)的区别。
- map的底层实现。
- channel的底层实现。
- select case的用法，有哪些运用场景。
- goroutine是什么，启动一个goroutine需要多少内存。
- 并发编程一般用什么包，sync.Group的作用是什么。
- gmp实现底层原理。
- golang的垃圾回收机制原理。
- golang的内存管理模型。
- gorm中事务需要手动处理么。
- **uniswap的理解**
- 补充：
- 谈谈你对uniswap的认识。
- uniswap v2和v3的区别。
- uniswap增加了流动性k值会变么。
- uniswap的event事件有哪些。
- 什么是最优路径，谈谈你的理解。
- 闪电贷的流程是否清楚。
- 前端需要会 the graph 同步链上数据

- 后端需要用 go-ethereum 同步链上数据
- sepolia测试网上面，发行erc20并通过脚本在uniswap V3创建资金池，添加流动性
- 项目的理解
- 补充：
- 你的项目如何支持高并发，高可用的。
- 你的项目接口响应速度是如何做优化的。
- 扫链过程中出现网络分叉一般怎么处理。
- 交易过程中，是如何保证安全性的。
- 你项目的私钥一般是如何存储的。
- 你项目是如何支持不同的链信息汇聚的。
- 你们公司的链节点是自己搭的还是用的第三方的。
- 简历扩展知识(如果简历有体现)

补充：

- defi借贷协议相关，你熟悉的借贷协议有哪些，如aave，请大概描述其中的一些核心的业务。
- 跨链相关，你们公司使用的跨链桥方案是什么，如LayerZero，请说出LayerZero的架构，如何优化gas等。
- solana、sui链，说下你如何在这些链上做交易的，这些高性能链的共识原理理解。
- 多链聚合，如何实现的聚合业务，安全性和实效性如何保障。
- 交易所业务，对订单撮合、清算结算、永续合约等业务理解。

✓ **通过标准：**能用简洁清楚的语言解释上述概念，80% 以上答对。

2 编程能力标准

学员需要完成以下任务：

- Go 语言后端：
 - 使用Gin写一个简单RESTful API，能调用数据库（MySQL/Postgres）+ Redis。
 - 使用ethclient读取某个合约的事件日志，写入数据库，并对外提供查询接口。
- Solidity 合约：
 - 能独立编写并部署一个ERC20合约（支持转账 & 铸造）。
 - 能写一个简单的Staking合约（用户存入代币 → 获得奖励）。
 - 能用Hardhat编写单元测试，覆盖率 > 80%。

✔ **通过标准：**完成Demo并能清楚解释每个模块的设计。

3 项目作品标准

学员至少要完成1~2个可展示的完整小项目，并上传到GitHub/GitLab：

- **必选项目（至少 1 个）：**
 - Meme Token（含税收 / LP 注入机制）。
 - Staking / Farming 合约 + 后端服务（存款、取款、奖励分发，带事件监听）。
- **可选项目（加分项）：**
 - NFT Market Demo（上架、购买、转让）。
 - DEX Swap Demo（简化版 Uniswap V2）。

✔ **通过标准：**代码完整，Readme写清楚架构与运行方式，能运行成功。

4 工具链掌握标准

学员需要熟练使用以下工具：

- Hardhat（or Foundry）：能完成部署、单元测试、mainnet fork测试。
- Go-Ethereum：会用abigen生成Go绑定代码，能编写链上交互逻辑。
- Git：熟悉基本分支管理& PR流程。
- Postman/Insomnia：会调试合约后端API。

✔ **通过标准：**能在1小时内完成一次合约部署 + 后端API接口调试。

5 面试软技能标准

- 能用英文或中文清晰描述自己做过的项目（合约功能、后端架构、测试方式）。
- 能回答“如果要在生产环境上线这个合约/后端服务，你会怎么保证安全？”
- 面试模拟时，遇到不会的题，能保持逻辑思考，不直接沉默。

补充：

- 面试过程中，问到项目的团队规模，项目周期的时候，心里大概要有个数。
- 为什么从上一家公司离职，需要提供一个合适的理由。

✔ **通过标准：**表达流畅，能展现思路，具备基本工程师的职业沟通能力。

最终面试阶段标准

达到以下条件，才能进入面试阶段：

1. 区块链核心知识答题 $\geq 80\%$ 。
2. 完成至少**1个合约项目 + 1个后端项目**，可运行并有完整Readme。
3. 能演示合约部署 + API调用的流程，全程不依赖讲师提示。
4. 能清楚解释自己代码中的关键点（合约逻辑 / 后端监听 / 数据结构）。
5. 能进行一次30分钟的模拟面试，并达到及格线。

Web3面试模拟题清单

初级（基础认知 + 入门编程）

适合：刚完成第3阶段（Solidity基础）或第4阶段（后端实战）的学员。

区块链基础

6. 区块链的区块里主要包含哪些数据？
7. 公钥和私钥的关系是什么？
8. 什么是Gas？为什么会出现Gas费高的情况？
9. 请解释一下51%攻击。

Go 后端

10. Go 语言里Goroutine和线程的区别。
11. 请写一个最简单的RESTful API：GET /ping → 返回 { "message": "pong" }。
12. Gorm和原生SQL的区别是什么？
13. 你会如何在Go里处理错误（error）？

Solidity 合约

14. memory、storage、calldata 的区别。
15. view 和 pure 函数的区别。
16. 请写一个最简单的 ERC20 代币合约（mint + transfer）。
17. 什么是事件（event），它在DApp中的作用是什么？

 **考察点：**基本概念是否清晰，能否写出简单的API和合约。

中级（实战能力 + 工程思维）

适合：完成第5阶段（合约实战）学员，准备开始做项目。

区块链基础

- 18. 什么是Merkle Tree？它在区块链中起到什么作用？
- 19. PoW和PoS的主要区别是什么？
- 20. 请描述一笔交易从发起到上链的过程。

Go 后端

- 21. 请解释channel在Go并发编程中的作用，并举例。
- 22. 使用Go-ethereum (ethclient) 如何读取某个合约的事件？
- 23. 如果API/RPC不稳定，你会如何保证后端服务的可用性？
- 24. 请解释context.Context在Go网络编程中的用途。

Solidity 合约

- 25. 如何防止合约的重入攻击？
- 26. require / assert / revert 的区别是什么？
- 27. modifier的作用，请举例。
- 28. 你知道哪些合约升级方案？（如 Proxy Pattern）
- 29. 请列出常见的Gas优化方法。

✅ **考察点：**能否独立写常见合约逻辑，理解Go后端链上交互，有初步的安全意识。

高级（综合能力 + 项目经验）

适合：准备第 6 阶段（真实项目）的学员。

区块链基础

- 30. 区块链为什么具有不可篡改性？请结合哈希和共识解释。
- 31. 如果一个区块链节点发生分叉，网络会如何达成一致？
- 32. 解释什么是随机数攻击，为什么链上随机数生成困难？

Go 后端

- 33. 设计一个服务：监听以太坊区块，并把结果写入数据库，你会如何划分模块？
- 34. 如果你的后端需要处理上千 TPS 的交易监听，如何优化？（缓存、并发、消息队列）
- 35. 在Gin项目中，如何设计中间件来做身份认证？

36. 请解析Go语言内部的并发模型（GMP）。

Solidity 合约

37. 设计一个Staking合约，支持：用户存入 ERC20 → 获得奖励 → 取款。

38. 如何在 Hardhat 或者 Foundry 里做 mainnet fork 测试？举一个场景。

39. 请解释代理合约（Proxy + Implementation）模式的工作原理。

40. 如果一个合约需要与 Chainlink 预言机交互，你会如何设计？

综合实战题

41. 写一个 Meme Token（带税收机制 + 自动注入 LP）。

42. 写一个 NFT 市场Demo（用户上架 → 购买 → 下架）。

43. 如果要在生产环境上线你的合约，你会从安全和性能角度做哪些检查？

44. 中心化交易所CEX和去中心化交易所DEX的区别在哪里，各有什么样的优缺点？

✅ **考察点：**能否完成 DApp 全链路（合约 + 后端 + 数据服务），具备工作面试的水准。