

Data: 02/10/2025

Sessão de Teste

1. Qual é o objetivo desta sessão?

Re-testar os endpoints de **Portfolio** e **History** da Smart Wallet no contexto completo da **Trilha A**, após ter executado:

- Criação de Smart Wallet
- Criação e processamento de sessão KYC (resultou em status FAILED)
- Tentativa de operação Fiat (bloqueada por permissão)

Objetivos específicos:

- Verificar se Portfolio mostra informações relacionadas a KYC ou Fiat
- Verificar se History registra eventos de KYC ou tentativas de Fiat
- Comparar resultados com testes anteriores (wallet vazia)
- Identificar integrações entre diferentes funcionalidades da API
- Validar se endpoints refletem estado completo da conta

2. Qual abordagem você vai usar?

Ferramenta: Postman para testes de API REST

Metodologia:

1. Consultar portfolio da Smart Wallet após execução de KYC
2. Consultar histórico da Smart Wallet após execução de KYC
3. Comparar respostas com testes anteriores (Daily Board #02)
4. Analisar se há diferenças ou integrações
5. Documentar escopo real dos endpoints

Endpoints testados:

- GET /wallets/{walletAddress}/portfolio
- GET /wallets/{walletAddress}/history

3. Há algo que precisa ser configurado antes de começar?

Configurações já realizadas:

- Smart Wallet criada e operacional
- Variável `smart_wallet_address` configurada no Postman
- API Key configurada
- Sessão KYC criada (sessionId: a7fd2ad4-12b4-4464-bbca-0fbbafd7ef98)
- Sessão KYC processada (status: FAILED)
- Tentativa de acesso a Fiat realizada (bloqueada com 403)

Contexto dos testes anteriores:

- Teste #9 (Daily Board #02): Portfolio com wallet vazia
- Teste #10 (Daily Board #02): History com wallet sem transações

Objetivo da comparação: Identificar se operações off-chain (KYC, Fiat) aparecem nesses endpoints que foram testados apenas com wallet vazia anteriormente.

4. Você conseguiu atingir o objetivo da sessão?

- Sim

Detalhamento:

Conseguir re-testar ambos endpoints e realizar comparação completa com testes anteriores.

Resultados obtidos:

- Portfolio consultado após KYC: arrays vazios (idêntico ao teste anterior)
- History consultado após KYC: sem transações (idêntico ao teste anterior)
- Nenhuma informação sobre KYC aparece em nenhum endpoint
- Nenhum evento relacionado a KYC registrado no histórico
- Nenhuma referência a tentativa de operação Fiat

Descoberta principal: os endpoints Portfolio e History são **exclusivamente focados em ativos e transações blockchain**. Eles não integram ou mostram informações sobre:

- Status de verificação KYC
- Eventos de compliance ou verificação de identidade
- Operações Fiat (depósitos, saques, cotações)
- Metadados off-chain ou dados administrativos

Estatísticas:

- Testes executados: 2
- Comparações realizadas: 2
- Problemas técnicos encontrados: 0
- Observações arquiteturais: 1

5. Problemas encontrados

Não foram identificados problemas técnicos.

Os endpoints funcionam exatamente como esperado para seu propósito específico. A ausência de informações sobre KYC e Fiat não é um bug, mas sim uma decisão arquitetural de separação de responsabilidades.

Observação Arquitetural (não é problema):

A API mantém clara separação entre:

Dados On-Chain (Blockchain):

- Portfolio: tokens ERC-20, NFTs ERC-721/1155
- History: transfers, swaps, transações on-chain

Dados Off-Chain (Compliance/Administrativo):

- KYC: verificação de identidade, documentos, status
- Fiat: operações financeiras tradicionais, conversões

Esta separação é arquiteturalmente correta e importante para:

- Segurança de dados sensíveis (KYC/Fiat ficam isolados)
- Compliance regulatório (dados pessoais protegidos)
- Performance (endpoints blockchain não carregam dados administrativos)
- Clareza de responsabilidades (cada endpoint tem escopo bem definido)



6. Observações adicionais

Detalhamento dos Testes

Teste #21: Get Portfolio - After KYC

Request:

```
GET /wallets/0x34378c28a87ac84266211aa9b1c77caca241a659/portfolio
x-api-key: [API_KEY]
```

Response:

```
Status: 200 OK
{
  "tokens": [],
  "nfts": [],
  "portfolio": []
}
```

Comparação com Teste #9 (antes de KYC):

- Response: **Idêntico**
- Estrutura: **Sem mudanças**
- Conclusão: Portfolio não é afetado por KYC

Análise:

- Endpoint funciona corretamente
- Foco exclusivo em ativos blockchain
- Arrays vazios são corretos (wallet sem tokens/NFTs)
- Não há integração com sistema de KYC
- Design intencional: Portfolio = ativos financeiros blockchain apenas



Teste #22: Get History - After KYC

Request:

```
GET /wallets/0x34378c28a87ac84266211aa9b1c77caca241a659/history
x-api-key: [API_KEY]
```

Response:

```
Status: 200 OK
{
  "nextLastId": null,
  "transactions": []
}
```

Comparação com Teste #10 (antes de KYC):

- Response: **Idêntico**
- Estrutura: **Sem mudanças**
- Conclusão: History não registra eventos de KYC

Análise:

- Endpoint funciona corretamente
- Foco exclusivo em transações blockchain
- Array vazio é correto (wallet sem transações on-chain)
- Eventos de KYC não aparecem aqui
- Design intencional: History = transações on-chain apenas



Sugestões para Melhorar Developer Experience

1. Documentar Claramente o Escopo dos Endpoints

Na documentação de Portfolio:

```
## GET /wallets/{address}/portfolio

### Escopo
Retorna **apenas ativos blockchain**:
- Tokens ERC-20
- NFTs (ERC-721, ERC-1155)
- Saldos em múltiplas chains

### NÃO inclui
```

- Status de KYC
- Saldo em moeda fiduciária (fiat)
- Operações pendentes off-chain
- Metadados administrativos

Para obter informações sobre KYC

Use: GET /kyc/individual-verification-sessions/standard/{sessionId}

Para obter informações sobre Fiat

Use: GET /fiat/orders (se disponível)

Na documentação de History:

GET /wallets/{address}/history

Escopo

Retorna ****apenas transações on-chain****:

- Transferências de tokens
- Swaps de tokens
- Transferências de NFTs
- Interações com smart contracts

NÃO inclui

- Eventos de verificação KYC
- Depósitos ou saques fiat
- Cotações ou ordens off-chain
- Eventos administrativos

Para histórico completo da conta

Combine com:

- GET /kyc/.../sessions para eventos KYC
- GET /fiat/orders para operações fiat (se disponível)

2. Criar Endpoint de Status Consolidado

Sugestão de novo endpoint:

GET /account/overview

Response:

```
{
  "wallets": [
    {
      "address": "0x...",
      "chainId": 137,
      "hasTokens": false,
      "hasNFTs": false
    }
  ],
  "kyc": {
    "status": "FAILED",
    "sessionId": "xxx",
    "lastUpdated": "2025-10-02T06:06:07.357Z"
  },
  "fiat": {
    "enabled": false,
    "reason": "PERMISSION_REQUIRED"
  },
  "features": {
    "wallet": true,
```

```
"kyc": true,  
"fiat": false,  
"swaps": false,  
"transfer": true  
}  
}
```

3. Adicionar Links de Navegação na Documentação

Criar fluxo de navegação claro:

Smart Wallet → Endpoints relacionados:

- └ Portfolio (ativos blockchain)
- └ History (transações blockchain)
- └ KYC (verificação de identidade)
- └ Fiat (operações financeiras tradicionais)

Para visão completa da conta, consulte:

- Portfolio: ativos disponíveis
- History: movimentações realizadas
- KYC Status: verificação de identidade
- Fiat Orders: operações fiat/crypto

4. Exemplos Práticos na Documentação

Adicionar seção "Use Cases":

Casos de Uso

Ver saldo total do usuário

1. GET /wallets/{address}/portfolio → tokens e NFTs
2. (se Fiat habilitado) GET /fiat/balance → saldo fiat

Ver histórico completo de atividades

1. GET /wallets/{address}/history → transações blockchain
2. GET /kyc/.../sessions → eventos de verificação
3. (se Fiat habilitado) GET /fiat/orders → operações fiat

Verificar se usuário pode fazer operação fiat

1. GET /kyc/.../sessions → verificar status APPROVED
2. Verificar se Fiat está habilitado na conta



Possíveis Melhorias de UX (Opcionais)

Se a equipe NotusLabs quiser melhorar integração visual:

Opção 1: Flags Opcionais nos Endpoints

GET /wallets/{address}/history?includeOffChain=true

Response:

```
{  
  "nextLastId": null,  
  "transactions": [],  
  "offChainEvents": [  
    {  
      "type": "KYC_CREATED",  
      "timestamp": "2025-10-02T05:59:38.348Z",  
    }  
  ]  
}
```

```
"status": "PENDING"
},
{
  "type": "KYC_PROCESSED",
  "timestamp": "2025-10-02T06:06:07.357Z",
  "status": "FAILED"
}
]
```

Opção 2: Metadata na Wallet

GET /wallets/{address}

Response:

```
{
  "wallet": { ... },
  "relatedData": {
    "hasKycSession": true,
    "kycStatus": "FAILED",
    "fiatEnabled": false,
    "lastActivity": "2025-10-02T06:06:07.357Z"
  }
}
```

Opção 3: Endpoint de Timeline Unificado

GET /account/timeline

Response:

```
{
  "events": [
    {
      "type": "WALLET_CREATED",
      "timestamp": "2025-10-02T02:29:02.982Z",
      "details": { "address": "0x..." }
    },
    {
      "type": "KYC_SESSION_CREATED",
      "timestamp": "2025-10-02T05:59:38.348Z",
      "details": { "sessionId": "..." }
    },
    {
      "type": "KYC_PROCESSED",
      "timestamp": "2025-10-02T06:06:07.357Z",
      "details": { "status": "FAILED" }
    }
  ]
}
```



Lições Aprendidas sobre Arquitetura da API

1. Separação Clara de Domínios:

- Blockchain: Portfolio, History
- Compliance: KYC

- Finance: Fiat
- Cada um tem endpoints independentes

2. Sem Cross-Referências Automáticas:

- Endpoints não referenciam outros domínios
- Desenvolvedor precisa consultar múltiplos endpoints
- Integração é responsabilidade do cliente

3. Design Orientado a Propósito:

- Cada endpoint tem escopo muito específico
- Não há endpoints "catch-all" ou "dashboard"
- Abordagem modular e desacoplada

4. Benefícios desta Abordagem:

- Segurança (dados sensíveis isolados)
- Performance (queries focadas)
- Escalabilidade (serviços independentes)
- Clareza (responsabilidades bem definidas)

5. Trade-offs:

- Desenvolvedores precisam fazer múltiplas chamadas
- Mais complexidade na integração client-side
- Necessidade de agregar dados manualmente



Recomendações Finais

Para NotusLabs:

Curto Prazo:

1. Documentar claramente o escopo de cada endpoint
2. Adicionar seção "O que este endpoint NÃO retorna"
3. Criar guia de integração mostrando como combinar endpoints

Médio Prazo: 4. Considerar endpoint de overview/dashboard 5. Adicionar exemplos de casos de uso comuns 6. Melhorar navegação entre documentações relacionadas

Longo Prazo: 7. Avaliar criação de SDK que abstraia múltiplas chamadas 8. Considerar GraphQL para queries flexíveis 9. Criar ambiente de testes integrado

Para Desenvolvedores Integrando:

- Não espere que Portfolio/History mostrem dados de KYC/Fiat
- Planeje múltiplas chamadas para visão completa
- Implemente cache para otimizar performance

- Agregue datos no lado do cliente