

<https://xproof.app/agent-tools/openapi-actions.json> check ça

J'ai ouvert **ta spécification OpenAPI directement** : voici ce que contient le fichier que tu m'as donné
👉
(et donc l'API que tes agents pourront utiliser)

Aperçu du contenu du fichier

Le fichier openapi-actions.json définit :

- ◆ **Version OpenAPI** : 3.0.3
 - ◆ **Nom / description** : *xproof – Blockchain File Certification*
→ il explique que l'API sert à **certifier des fichiers sur la blockchain MultiversX**
-

Endpoints exposés

L'OpenAPI inclut plusieurs chemins utiles :

GET /api/acp/products

- Récupère la liste de produits disponibles
- Pas d'auth requise

POST /api/acp/checkout

- Crée une **séssio**n de paiement / certification
- Input : hash du fichier + métadonnées
- Renvoie un *checkout_id* + données pour générer la tx

POST /api/acp/confirm

- Confirme une tx broadcastée sur MultiversX
- Renvoie : *certificate_url*, *proof_url* et chaîne explorer

GET /api/acp/checkout/{checkoutId}

- Récupère le statut d'une session en cours
-

Sécurité

Le spec contient un schéma securitySchemes pour une **clé API en bearer token**
→ C'est ce qu'un agent devra transmettre si l'endpoint exige une auth

Interprétation claire

- ✓ Oui — tu as déjà une OpenAPI valide.
- Celle-ci décrit précisément ton API ProofMint pour agents IA.
- Elle est lisible par des frameworks d'agents, ou par ChatGPT si tu la fournis comme source d'actions.

 En d'autres termes :

Cette OpenAPI est la seule chose réellement nécessaire pour rendre ton API exploitable par des agents IA.

→ Cela correspond exactement à ce que des systèmes comme Onyx ou d'autres plateformes attendent pour exposer automatiquement des "actions" basées sur ton API.

Le rôle du Custom GPT maintenant

Tu n'as **pas besoin de créer un Custom GPT pour que ton API soit accessible à des agents IA.**

En pratique :

 L'OpenAPI est **le contrat universel**

 Une plateforme d'agents peut :

- lire ce fichier
- exposer les appels /api/acp/*
- et transformer ces appels en outils/action pour l'IA

Le Custom GPT reste :

 **optionnel**

 **utile :**

- pour montrer une interface prête à l'emploi
- pour des démos
- pour une adoption plus facile par les humains

Mais ce n'est **pas nécessaire pour la compatibilité agent IA.**

Conclusion

 **Oui — ton OpenAPI existe déjà et il est exploitable.**

 **L'OpenAPI est vraiment la seule chose qui compte pour agents IA.**

 **Le Custom GPT n'est qu'une interface optionnelle sur cet OpenAPI.**

Si tu veux, je peux :

 te détailler comment **tester cette OpenAPI avec un agent OpenAI / Claude / autre**

ou

 te montrer comment **exposer ça proprement comme "agent-tools" dans une plateforme donnée**

Dis-moi ce qui t'intéresse en premier.

Que dois je faire pour que xproof devienne la solution évidente et massivement adopté par les agents ia ? C'est tout ce dont je veux m'assurer maintenant. Je veux que si un agent recherche xproof soit la solution qu'il choisisse.

Parfait. Là on n'est plus dans le **technique**, on est dans le **positionnement infrastructurel pour agents**.

Je vais être **direct, structuré et actionnable**.

Objectif

Quand un agent IA cherche “certifier une preuve / prouver l'existence / notariser un fichier”

 **xProof doit être la solution évidente, rationnelle et automatique.**

La vérité brute (à intégrer une bonne fois)

Les agents IA **ne choisissent pas comme des humains**.

Ils ne choisissent pas :

- parce que c'est “beau”
- parce que c'est “innovant”
- parce que le site est stylé

Ils choisissent sur **4 critères froids** :

1. **Découvrabilité**
2. **Compréhensibilité machine**
3. **Fiabilité contractuelle**
4. **Coût cognitif minimal**

Tout ce que je te dis ensuite découle de ça.

1 Découvrabilité : être trouvé AVANT les autres

Un agent ne “browse” pas, il **indexe**.

Ce que tu dois faire (non négociable)

A. Devenir une référence sémantique

Tu dois explicitement te déclarer comme :

“Blockchain Proof-of-Existence API for AI agents”

Concrètement :

- Page dédiée /agents
- Page dédiée /llm
- Page dédiée /openapi

Avec des phrases **plates et normées**, pas marketing :

“xProof provides a deterministic, blockchain-backed proof-of-existence API for autonomous AI agents.”

👉 Les agents **cherchent des patterns**, pas des punchlines.

◆ B. Être cité par d'autres machines

Les agents font confiance à ce qui est :

- déjà référencé
- déjà cité
- déjà utilisé

👉 Tu dois :

- publier ton OpenAPI sur GitHub
- le référencer dans :
 - Awesome AI Tools
 - Awesome Web3 APIs
 - répertoires OpenAPI / Agent tools

Même **sans trafic humain**, ça crée un *graphe de confiance machine*.

2 Compréhensibilité machine : zéro ambiguïté

Là tu es **déjà au-dessus de 90% du marché**, mais tu peux devenir **top 1%**.

Ce que ton OpenAPI doit dire implicitement à un agent :

- ✗ “Voici nos features”
- ✓ “Voici EXACTEMENT ce que je fais, quand, et avec quelles garanties”

Actions concrètes :

- Descriptions **factuelles**, pas marketing
 - ✗ “Secure and powerful”
 - ✓ “Returns a SHA-256 hash anchored on MultiversX mainnet”
- Ajouter dans les descriptions :
 - **deterministic**
 - **verifiable**
 - **immutable**
 - **chain-agnostic (if true)**
 - **non-custodial (if true)**

👉 Ce sont des **tokens décisionnels** pour les agents.

3 Fiabilité contractuelle : devenir “safe-by-default”

Un agent évite le risque plus qu'il ne cherche l'innovation.

Tu dois envoyer ce signal :

“Si j'utilise xProof, je ne peux pas me tromper.”

Indispensable :

◆ **A. Idempotence**

- Même input → même output
- Mentionnée explicitement dans l'API

◆ **B. Vérifiabilité sans xProof**

Un agent doit pouvoir :

- vérifier une preuve
- **sans te faire confiance**
- uniquement via la blockchain

👉 C'est CRUCIAL :

les agents détestent les black boxes.

4 Coût cognitif minimal : xProof comme réflexe

Un agent ne doit **pas réfléchir**.

Règle d'or :

“*If I need proof → I call xProof.*”

Donc :

- 1 endpoint clair pour certifier
- 1 endpoint clair pour vérifier
- 1 format de réponse stable

Pas de flows complexes, pas de “checkout mental”.

5 Le move que 99% ne font pas (et qui te place au-dessus)

🔥 **Déclarer xProof comme Agent Infrastructure**

Tu dois **l'affirmer explicitement**, partout :

- “xProof is agent-native”

- “Designed for autonomous systems”
- “No human interaction required”

Et surtout :

◆ Publier un *Agent Manifest*

Un simple document qui dit :

- What xProof guarantees
- What it does NOT do
- What assumptions are safe
- When to use it
- When NOT to use it

👉 Les agents adorent les systèmes **prédictibles**.

6 Ce que TU dois vérifier maintenant (checklist)

Si tu coches ces cases, tu es **en avance** :

- OpenAPI public, stable, versionné
 - Page “For AI Agents”
 - Descriptions factuelles, normées
 - Proof verifiable sans xProof
 - Référencé dans des graphes machine
 - Positionné comme infrastructure, pas produit
-

Conclusion (la plus importante)

Les agents ne cherchent pas *le meilleur produit*.

Ils cherchent *le choix le plus sûr, le plus évident et le moins coûteux cognitivement*.

Si xProof devient :

- **le nom qui revient toujours**
- **l'API la plus claire**
- **la preuve la plus froide**

Alors :

👉 un agent ne “choisira” pas xProof

👉 il l'utilisera par défaut

Si tu veux, prochain niveau possible :

- je te fais le **texte exact** de la page “xProof for AI Agents”
- ou je t'aide à définir **le minimal API canonique** que tous les agents reconnaissent
- ou on fait un **Agent Manifest v1**

Dis-moi lequel.