

TRUSTFUL AGENTS

Governance API

Developer Documentation

Version 1.3
January 2026

Contents

Table of Contents	Error! Bookmark not defined.
1. Introduction	5
1.1 Base URL.....	5
1.2 Authentication	5
1.3 Rate Limiting	5
1.4 Response Format.....	5
2. Authentication Routes.....	6
2.1 GetNonce	6
2.2 Login	6
2.3 Logout.....	7
2.4 GetProfile	7
2.5 UpdateProfile	7
3. Claims Routes	9
3.1 ListClaims.....	9
3.2 GetClaimDetails	9
3.3 GetVotes	10
3.4 GetMyVote	11
3.5 PrepareVoteTransaction.....	11
3.6 PrepareFinalizeTransaction.....	11
3.7 CalculateRequiredDeposit.....	12
3.8 GetMember'sCouncils	12
3.9 GetMember'sPendingClaims	12
3.10 GetClaimMetadata	12
3.11 SaveClaimMetadata	13
3.12 GetClaimMessages	13
3.13 PostClaimMessage	14
4. Councils Routes.....	15
4.1 ListCouncils.....	15
4.2 GetCouncilDetails.....	15
4.3 GetCouncilMembers.....	15
4.4 GetCouncilAgents	16
4.5 CheckIfCouncilCanClose.....	16
4.6 ProposeCreateCouncil	16
4.7 ProposeCloseCouncil.....	17
4.8 ProposeAddMember	17
4.9 ProposeRemoveMember.....	17

4.10 Update Member Metadata	18
5. Safe Routes	19
5.1 Get Safe Info	19
5.2 Get Pending Transactions	19
5.3 Get Transaction	19
5.4 Propose Transaction	19
5.5 Record Signature	20
5.6 Record Execution	20
6. Agents Routes	21
6.1 Get Agent Council Assignment	21
6.2 Propose Agent Reassignment (Body)	21
6.3 Propose Agent Reassignment (Path)	21
7. Pending Transactions Routes	23
7.1 List Pending Transactions	23
7.2 Get Pending Transaction	23
7.3 Store Pending Transaction	23
7.4 Reject Pending Transaction	23
7.5 Sync Pending Transactions	24
8. Provider Agents Routes	25
8.1 Get Agent Metadata	25
8.2 Get Agents by Owner	25
8.3 Create Agent Metadata	25
8.4 Update Agent Metadata	26
9. Validation Routes (A2A Protocol)	27
9.1 Get Agent Card	27
9.2 Get Agent Image	27
9.3 Get Validation Status	28
9.4 Get Trust Info	28
9.5 API Health (v1)	28
10. Health Routes	30
10.1 System Health	30
Appendix A: HTTP Error Codes	31
Appendix B: Type Definitions	32
Address Types	32
Claim Status Enum	32
Vote Enum	32
Action Types	32
Author Roles	32
Appendix C: Complete Endpoint Matrix	33

1. Introduction

The Trustful Agents Governance API provides a REST interface for interacting with the Trustful Agents protocol. It serves as the backend for the governance, provider, claimer, and council dashboards, providing endpoints for authentication, council management, claims processing, and agent validation.

1.1 Base URL

```
Production: https://api.trustful-agents.ai  
Testnet:     https://api-sepolia.trustful-agents.ai
```

1.2 Authentication

The API uses Sign-In with Ethereum (SIWE) for authentication. Protected endpoints require a Bearer token obtained through the login flow:

1. Request a nonce from GET /auth/nonce
2. Sign a SIWE message with your wallet
3. Submit the signed message to POST /auth/login
4. Use the returned token in the Authorization header

```
Authorization: Bearer <session_token>
```

1.3 Rate Limiting

General endpoints: 100 requests per 15 minutes per IP

Authentication endpoints: 20 requests per 15 minutes per IP

1.4 Response Format

All responses are JSON. Successful responses return the data directly. Error responses follow this format:

```
{  
  "error": "Error message description",  
  "details": [...] // Optional validation details  
}
```

2. Authentication Routes

Authentication uses Sign-In with Ethereum (SIWE) to establish sessions. Only Safe multisig owners can log in to governance-protected endpoints.

2.1 Get Nonce

GET /auth/nonce

Generates a random nonce for constructing a SIWE message. The nonce is used to prevent replay attacks.

Parameters

None

Response

Field	Type	Description
nonce	string	Random nonce string for SIWE message

Example Response

```
{  
  "nonce": "abcd1234efgh5678ijkl"  
}
```

2.2 Login

POST /auth/login

Verifies a signed SIWE message and creates a session. Only addresses that are owners of the governance Safe multisig can successfully authenticate.

Request Body

Parameter	Type	Required	Description
message	string	Yes	The SIWE message that was signed
signature	string	Yes	Hex-encoded signature (0x-prefixed)

Response

Field	Type	Description
token	string	Session token for authenticated requests
address	string	Authenticated wallet address
expiresAt	string	ISO 8601 timestamp when session expires

Errors

Code	Description
401	Authentication failed - invalid signature or SIWE message
403	Access denied - address is not a Safe multisig owner

Example Request

```
POST /auth/login
Content-Type: application/json

{
  "message": "trustful-agents.ai wants you to sign in...",
  "signature": "0x1234abcd..."
}
```

Example Response

```
{
  "token": "session_abc123xyz",
  "address": "0x1234...5678",
  "expiresAt": "2026-01-31T12:00:00.000Z"
}
```

2.3 Logout

POST /auth/logout

Invalidates the current session token.

Headers

Parameter	Type	Required	Description
Authorization	string	Yes	Bearer <session_token>

Response

Field	Type	Description
success	boolean	Always true on success

2.4 Get Profile

GET /auth/me

Returns the authenticated user's profile information.

Response

Field	Type	Description
address	string	Wallet address
name	string null	Display name if set
email	string null	Email address if set
isSafeSigner	boolean	Whether user is a Safe owner

2.5 Update Profile

PUT /auth/me

Updates the authenticated user's profile information.

Request Body

Parameter	Type	Required	Description
name	string	No	Display name (max 255 chars)
email	string	No	Email address

Response

Field	Type	Description
address	string	Wallet address
name	string null	Updated display name
email	string null	Updated email address

3. Claims Routes

Claims routes provide read access to claim data from the ClaimsManager smart contract, plus database-stored metadata and messages.

3.1 List Claims

GET /claims

Returns a list of claims with optional filtering. Without filters, returns the most recent 50 claims.

Query Parameters

Parameter	Type	Required	Description
councilId	bytes32	No	Filter by council ID
claimant	address	No	Filter by claimant address
agentId	number	No	Filter by agent ID
status	string	No	Filter by status (Filed, Approved, Rejected, etc.)
pending	boolean	No	If true with councilId, returns only pending claims

Response

Field	Type	Description
claims	Claim[]	Array of claim objects
count	number	Total number of claims returned

3.2 Get Claim Details

GET /claims/:claimId

Returns detailed information about a specific claim including voting progress.

Path Parameters

Parameter	Type	Required	Description
claimId	number	Yes	The claim ID

Response

Field	Type	Description
claimId	string	Claim identifier
agentId	string	Agent token ID
claimant	string	Claimant wallet address
claimedAmount	string	Amount claimed in USDC (wei)
approvedAmount	string	Approved amount after voting
paymentReceiptHash	string	Hash of payment receipt
termsHashAtClaimTime	string	T&C hash when claim was filed
termsVersionAtClaimTime	string	T&C version number
providerAtClaimTime	string	Provider address at claim time
councilId	string	Assigned council ID
councilName	string	Council display name
claimantDeposit	string	Deposit paid by claimant (wei)
lockedCollateral	string	Collateral locked for this claim
status	string	Status name (Filed, Approved, etc.)

statusCode	number	Status enum value (0-7)
filedAt	string	ISO timestamp when filed
evidenceDeadline	string	ISO timestamp for evidence deadline
votingDeadline	string	ISO timestamp for voting deadline
hadVotes	boolean	Whether any votes were cast
isInEvidencePeriod	boolean	Currently in evidence period
isInVotingPeriod	boolean	Currently in voting period
canVote	boolean	Voting is currently allowed
canFinalize	boolean	Claim can be finalized
votingProgress	object	Voting statistics

Voting Progress Object

Field	Type	Description
approveVotes	string	Number of approve votes
rejectVotes	string	Number of reject votes
abstainVotes	string	Number of abstain votes
totalVotes	string	Total votes cast
requiredQuorum	string	Minimum votes for quorum
deadline	string	Voting deadline timestamp
quorumReached	boolean	Whether quorum was reached

Claim Status Codes

Code	Description
0	Filed - Claim submitted, in evidence period
1	EvidenceClosed - Evidence period ended
2	VotingClosed - Voting period ended
3	Approved - Claim approved by council
4	Rejected - Claim rejected by council
5	Executed - Funds distributed
6	Cancelled - Cancelled by claimant
7	Expired - No quorum reached

3.3 Get Votes

GET /claims/:claimId/votes

Returns all votes cast on a claim.

Response

Field	Type	Description
votes	Vote[]	Array of vote objects
count	number	Total number of votes

Vote Object

Field	Type	Description
voter	string	Voter wallet address
vote	string	Vote choice (Approve, Reject, Abstain)
voteCode	number	Vote enum (1=Approve, 2=Reject, 3=Abstain)
approvedAmount	string	Amount approved by this voter
reasoning	string	Voter's reasoning/justification
votedAt	string	ISO timestamp of initial vote
lastChangedAt	string null	ISO timestamp if vote was changed

3.4 Get My Vote

GET /claims/:claimId/my-vote

Query Parameters

Parameter	Type	Required	Description
address	address	Yes	Wallet address to check

Response

Field	Type	Description
hasVoted	boolean	Whether address has voted
vote	Vote null	Vote object if voted, null otherwise

3.5 Prepare Vote Transaction

POST /claims/:claimId/vote

Returns encoded transaction data for casting or changing a vote. The transaction must be executed client-side.

Request Body

Parameter	Type	Required	Description
vote	number	Yes	Vote choice: 1=Approve, 2=Reject, 3=Abstain
approvedAmount	string	No	Amount to approve (for Approve votes)
reasoning	string	No	Justification for the vote
voterAddress	address	Yes	Address of the voter

Response

Field	Type	Description
transaction	object	Transaction object with to, data, value
isChangeVote	boolean	True if modifying existing vote
message	string	Human-readable description

3.6 Prepare Finalize Transaction

POST /claims/:claimId/finalize

Returns transaction data for finalizing or executing a claim. This endpoint is smart: it returns finalizeClaim() if the claim needs finalization, or executeClaim() if the claim is already finalized and needs execution.

Response

Field	Type	Description
transaction	object	Transaction object with to, data, value
step	string	'finalize' or 'execute'
message	string	Description of what the transaction does

Errors

Code	Description
400	Claim already executed or in invalid state

3.7 Calculate Required Deposit

GET /claims/deposit-calculator

Calculates the required deposit for filing a claim against an agent.

Query Parameters

Parameter	Type	Required	Description
agentId	number	Yes	The agent's token ID
claimedAmount	string	Yes	Amount to claim (wei)

Response

Field	Type	Description
agentId	string	The agent ID
claimedAmount	string	The claim amount
requiredDeposit	string	Required deposit in USDC (wei)

3.8 Get Member's Councils

GET /claims/members/:address/councils

Returns all councils where the address is an active member.

Response

Field	Type	Description
councils	Council[]	Array of council objects
count	number	Number of councils

3.9 Get Member's Pending Claims

GET /claims/members/:address/pending-claims

Returns all pending claims in councils where the address is a member, sorted by voting deadline (most urgent first).

Response

Field	Type	Description
claims	Claim[]	Array of pending claim objects with hasVoted field
count	number	Number of pending claims

3.10 Get Claim Metadata

GET /claims/:claimId/metadata

Returns off-chain metadata (title, description) for a claim.

Response

Field	Type	Description
claim_id	string	The claim ID
title	string null	Claim title
description	string	Detailed claim description
created_at	string	Creation timestamp
updated_at	string	Last update timestamp

3.11 Save Claim Metadata

POST /claims/:claimId/metadata

Saves or updates off-chain metadata for a claim. Note: This endpoint currently lacks authentication - see security considerations.

Request Body

Parameter	Type	Required	Description
title	string	No	Claim title
description	string	Yes	Detailed claim description

3.12 Get Claim Messages

GET /claims/:claimId/messages

Returns the threaded message conversation for a claim, including evidence submissions.

Response

Field	Type	Description
messages	Message[]	Array of root messages with nested replies
totalCount	number	Total message count

Message Object

Field	Type	Description
id	number	Message ID
claim_id	string	Parent claim ID
parent_id	number null	Parent message ID for replies
author_address	string	Author wallet address
author_role	string	Role: claimer, provider, or councilor
content	string	Message text content
evidence_hash	string null	Hash of attached evidence
evidence_data	string null	Base64-encoded evidence file
evidence_filename	string null	Original filename
evidence_mimetype	string null	File MIME type
evidence_size	number null	File size in bytes
created_at	string	Creation timestamp
replies	Message[]	Nested reply messages

3.13 Post Claim Message

POST /claims/:claimId/messages

Posts a message or evidence to a claim conversation. Only allowed during the evidence period. Role is validated against on-chain data.

Request Body

Parameter	Type	Required	Description
authorAddress	address	Yes	Message author wallet address
authorRole	string	Yes	Role: claimer, provider, or councilor
content	string	Yes	Message text content
parentId	number	No	Parent message ID for replies
evidenceHash	string	No	Hash of evidence file
evidenceData	string	No	Base64-encoded file (max 10KB)
evidenceFilename	string	No	Original filename
evidenceMimetype	string	No	File MIME type
evidenceSize	number	No	File size in bytes

Errors

Code	Description
400	Invalid authorRole or missing required fields
403	Evidence period ended, role mismatch, or councilor trying to submit evidence
404	Claim not found

4. Councils Routes

Council routes provide access to council data and governance operations. Read operations are public; write operations require Safe owner authentication.

4.1 List Councils

GET /councils

Returns all active councils with on-chain data and database metadata.

Response

Field	Type	Description
councils	Council[]	Array of council objects
count	number	Number of councils

Council Object

Field	Type	Description
councilId	bytes32	Unique council identifier
name	string	Council name
description	string	Council description
vertical	string	Industry vertical (Finance, Healthcare, etc.)
memberCount	number	Current number of members
quorumPercentage	number	Required quorum (basis points, 5000 = 50%)
claimDepositPercentage	number	Deposit percentage (basis points)
votingPeriod	number	Voting period in seconds
evidencePeriod	number	Evidence period in seconds
active	boolean	Whether council is active
createdAt	string	ISO timestamp of creation
closedAt	string null	ISO timestamp if closed

4.2 Get Council Details

GET /councils/:councilId

Path Parameters

Parameter	Type	Required	Description
councilId	bytes32	Yes	Council ID (0x-prefixed 64-char hex)

4.3 Get Council Members

GET /councils/:councilId/members

Returns all members of a council with on-chain and database metadata.

Response

Field	Type	Description
members	Member[]	Array of member objects

count	number	Number of members
-------	--------	-------------------

Member Object

Field	Type	Description
address	string	Member wallet address
councilId	string	Council ID
joinedAt	string null	ISO timestamp when joined
claimsVoted	number	Number of claims voted on
active	boolean	Whether member is active
name	string null	Display name from database
email	string null	Email from database
description	string null	Bio from database

4.4 Get Council Agents

GET /councils/:councilId/agents

Returns all agents assigned to a council (via T&C registration or governance override).

Response

Field	Type	Description
councilId	string	Council ID
agents	Agent[]	Array of agent objects
count	number	Number of agents

4.5 Check If Council Can Close

GET /councils/:councilId/can-close

Checks whether a council can be closed (no assigned agents and no pending claims).

Response

Field	Type	Description
canClose	boolean	Whether council can be closed
reason	string undefined	Reason if cannot close

4.6 Propose Create Council

POST /councils/propose-create

Generates transaction data for creating a new council via Safe multisig.

Request Body

Parameter	Type	Required	Description
name	string	Yes	Council name (3-100 chars)
description	string	No	Description (max 500 chars)
vertical	string	Yes	Industry vertical (2-50 chars)
quorumPercentage	number	No	Quorum in basis points (1000-10000, default 5000)

claimDepositPercentage	number	No	Deposit % (100-5000, default 1000)
votingPeriodDays	number	No	Voting period (1-30 days, default 7)
evidencePeriodDays	number	No	Evidence period (1-14 days, default 3)

Response

Field	Type	Description
transaction	object	Transaction data (to, data, value)
safeUrl	string	URL to Safe transaction page
message	string	Instructions

4.7 Propose Close Council

POST /councils/:councilId/propose-close

Generates transaction data for closing a council. Council must have no agents and no pending claims.

Errors

Code	Description
400	Council cannot be closed (has agents or pending claims)

4.8 Propose Add Member

POST /councils/:councilId/propose-add-member

Generates transaction data for adding a council member. Also stores member metadata in database.

Request Body

Parameter	Type	Required	Description
memberAddress	address	Yes	New member wallet address
name	string	No	Display name (max 100 chars)
email	string	No	Email address
description	string	No	Bio (max 500 chars)

Response

Field	Type	Description
transaction	object	Transaction data (to, data, value)
safeUrl	string	URL to Safe transaction page
memberMetadata	object	Stored metadata
message	string	Instructions

4.9 Propose Remove Member

POST /councils/:councilId/propose-remove-member

Request Body

Parameter	Type	Required	Description
memberAddress	address	Yes	Member address to remove

4.10 Update Member Metadata

PUT /councils/:councilId/members/:memberAddress/metadata

Updates off-chain metadata for a council member.

Request Body

Parameter	Type	Required	Description
name	string	No	Display name
email	string	No	Email address
description	string	No	Bio/description

5. Safe Routes

Safe routes interact with the Safe Transaction Service for multisig operations. Most endpoints require Safe owner authentication.

5.1 Get Safe Info

GET /safe/info

Returns information about the governance Safe multisig.

Response

Field	Type	Description
address	string	Safe contract address
threshold	number	Required signatures (e.g., 2 of 3)
owners	string[]	Array of owner addresses
nonce	number	Current nonce for transactions

5.2 Get Pending Transactions

GET /safe/pending

Returns pending transactions requiring signatures from the Safe Transaction Service.

5.3 Get Transaction

GET /safe/transactions/:safeTxHash

Path Parameters

Parameter	Type	Required	Description
safeTxHash	bytes32	Yes	Safe transaction hash

Response

Field	Type	Description
safeTxHash	string	Transaction hash
to	string	Target contract address
data	string	Encoded call data
value	string	ETH value (usually 0)
operation	number	0=Call, 1=DelegateCall
nonce	number	Transaction nonce
confirmations	number	Current signature count
confirmationsRequired	number	Required signatures
isExecuted	boolean	Whether TX was executed
proposer	string	Address that proposed TX
description	string	Human-readable description

5.4 Propose Transaction

POST /safe/propose

Logs a transaction proposal for audit purposes. Actual proposal happens client-side using Safe SDK.

Request Body

Parameter	Type	Required	Description
to	address	Yes	Target contract address
data	string	Yes	Encoded call data (0x-prefixed)
value	string	No	ETH value (default '0')
description	string	Yes	Human-readable description (max 500)

5.5 Record Signature

POST /safe/sign/:safeTxHash

Records that a signature was added. Actual signing happens client-side.

5.6 Record Execution

POST /safe/execute/:safeTxHash

Records that a transaction was executed. Actual execution happens client-side.

6. Agents Routes

Agent routes handle governance operations on agents, specifically council reassignment.

6.1 Get Agent Council Assignment

GET /agents/:id

Returns an agent's current council assignment and reassignment eligibility.

Path Parameters

Parameter	Type	Required	Description
id	number	Yes	Agent token ID

Response

Field	Type	Description
agentId	string	Agent token ID
councilId	bytes32	Current council ID (or zero bytes if none)
councilName	string null	Council name if assigned
pendingClaims	number	Number of pending claims
canReassign	boolean	Whether agent can be reassigned (no pending claims)

6.2 Propose Agent Reassignment (Body)

POST /agents/propose-reassign

Generates transaction data for reassigning an agent to a different council.

Request Body

Parameter	Type	Required	Description
agentId	number	Yes	Agent token ID
newCouncilId	bytes32	Yes	Target council ID

Response

Field	Type	Description
message	string	Instructions
transaction	object	Transaction data (to, data, value)
details	object	Reassignment details (agentId, fromCouncilId, toCouncilId, toCouncilName)

Errors

Code	Description
400	Agent has pending claims, target council doesn't exist, or target council is inactive

6.3 Propose Agent Reassignment (Path)

POST /agents/:id/reassign

Alternative endpoint with agent ID in path instead of body.

Path Parameters

Parameter	Type	Required	Description
id	number	Yes	Agent token ID

Request Body

Parameter	Type	Required	Description
newCouncilId	bytes32	Yes	Target council ID

7. Pending Transactions Routes

Routes for tracking pending Safe transactions with metadata in the database.

7.1 List Pending Transactions

GET /pending

Returns all pending transactions with database metadata and Safe API confirmation counts.

Response

Field	Type	Description
transactions	Transaction[]	Array of pending transactions
safeThreshold	number	Required signatures
safeOwners	number	Total number of owners

Transaction Object

Field	Type	Description
safeTxHash	string	Safe transaction hash
actionType	string	Type: create_council, add_member, etc.
title	string	Human-readable title
description	string null	Detailed description
metadata	object	Additional metadata
proposedBy	string	Proposer address
proposedAt	string	ISO timestamp
status	string	pending, executed, or rejected
confirmations	number	Current signature count
confirmationsRequired	number	Required signatures

7.2 Get Pending Transaction

GET /pending/:safeTxHash

7.3 Store Pending Transaction

POST /pending

Stores metadata for a pending transaction after proposing to Safe.

Request Body

Parameter	Type	Required	Description
safeTxHash	string	Yes	Safe transaction hash
actionType	string	Yes	Action type
title	string	Yes	Human-readable title
description	string	No	Detailed description
metadata	object	No	Additional data

7.4 Reject Pending Transaction

DELETE /pending/:safeTxHash

7.5 Sync Pending Transactions

POST /pending/sync

Synchronizes local database status with Safe Transaction Service. Updates executed and rejected transactions.

Response

Field	Type	Description
success	boolean	Operation succeeded
checked	number	Number of transactions checked
updated	number	Number of statuses updated

8. Provider Agents Routes

Routes for managing agent metadata in the provider dashboard. These store off-chain information about agents.

8.1 Get Agent Metadata

GET /provider/agents/:agentId

Returns stored metadata for an agent.

Response

Field	Type	Description
agent_id	string	Agent token ID
owner_address	string	Owner wallet address
name	string	Agent display name
description	string null	Agent description
capabilities	string[] null	List of capabilities
website_url	string null	Website URL
created_at	string	Creation timestamp
updated_at	string	Last update timestamp

8.2 Get Agents by Owner

GET /provider/agents/owner/:address

Returns all agents owned by an address.

Response

Field	Type	Description
agents	AgentMetadata[]	Array of agent metadata objects

8.3 Create Agent Metadata

POST /provider/agents

Creates metadata for a newly minted agent. Verifies on-chain ownership before saving.

Request Body

Parameter	Type	Required	Description
agentId	string	Yes	Agent token ID
ownerAddress	address	Yes	Owner wallet address
name	string	Yes	Display name (1-255 chars)
description	string	No	Description (max 2000 chars)
capabilities	string[]	No	Capabilities (max 20, each max 100 chars)
websiteUrl	string	No	Website URL (max 500 chars)

Errors

Code	Description
403	Not the owner of this agent (on-chain check failed)
409	Agent metadata already exists

8.4 Update Agent Metadata

PUT /provider/agents/:agentId

Updates agent metadata. Requires x-owner-address header for ownership verification.

Headers

Parameter	Type	Required	Description
x-owner-address	address	Yes	Owner address for verification

Request Body

Parameter	Type	Required	Description
name	string	No	Display name
description	string	No	Description
capabilities	string[]	No	Capabilities list
websiteUrl	string null	No	Website URL (null to remove)

Errors

Code	Description
400	Missing or invalid x-owner-address header
403	Not the owner of this agent

9. Validation Routes (A2A Protocol)

Public endpoints for A2A Protocol integration. These provide agent cards, validation status, and trust information.

9.1 Get Agent Card

GET /v1/agents/:agentId/agent-card.json

Returns the A2A Protocol Agent Card for an agent. This is the URI that tokenURI() should point to.

Response

Field	Type	Description
name	string	Agent display name
description	string	Agent description
image	string	URL to agent image SVG
external_url	string	Link to agent on dashboard
attributes	array	NFT-standard trait attributes
extensions.trustful	object	Trustful-specific extension data

Trustful Extension Object

Field	Type	Description
version	string	Extension version ('1.0')
validatorAddress	string	TrustfulValidator contract
chainId	number	Chain ID (84532 for Base Sepolia)
collateral	object null	Collateral info if deposited
terms	object null	T&C info if registered
validation	object	Validation status and timestamp
verificationUrl	string	URL to verify validation

Example Response

```
{  
  "name": "Finance Assistant Agent",  
  "description": "AI agent for financial analysis",  
  "image": "https://api.trustful-agents.ai/v1/agents/1/image.svg",  
  "external_url": "https://provider.trustful-agents.ai/agents/1",  
  "attributes": [  
    { "trait_type": "Validation Status", "value": "valid" },  
    { "trait_type": "Collateral (USDC)", "value": "1000.00" }  
,  
  "extensions": {  
    "trustful": {  
      "version": "1.0",  
      "validatorAddress": "0x9628C1...",  
      "chainId": 84532,  
      "validation": { "status": "valid", "issuedAt": "2026-01-15T..." }  
    }  
  }  
}
```

9.2 Get Agent Image

GET /v1/agents/:agentId/image.svg

Returns a dynamically generated SVG badge showing the agent's validation status and collateral amount.

Response

Content-Type: image/svg+xml

9.3 Get Validation Status

GET /v1/agents/:agentId/validation.json

Returns the current validation status for an agent.

Response

Field	Type	Description
agentId	string	Agent token ID
isvalidated	boolean	Current validation status
validatorAddress	string	TrustfulValidator contract
chainId	number	Chain ID
timestamp	string	Response timestamp

9.4 Get Trust Info

GET /v1/agents/:agentId/trust-info.json

Returns comprehensive trust information including collateral, terms, and validation details.

Response

Field	Type	Description
version	string	API version ('1.0')
agentId	string	Agent token ID
validatorAddress	string	Validator contract address
chainId	number	Chain ID
collateral.amount	string	Collateral amount in USDC
collateral.asset	string	'USDC'
collateral.vaultAddress	string	CollateralVault contract
collateral.withdrawalPending	boolean	Withdrawal initiated
terms.hash	string	T&C content hash
terms.uri	string	IPFS URI for T&C
terms.councilId	string	Assigned council
validation.status	string	'valid', 'invalid', or 'revoked'
validation.issuedAt	string null	When validation was issued
timestamp	string	Response timestamp

9.5 API Health (v1)

GET /v1/health

Simple health check endpoint.

Response

Field	Type	Description
status	string	'ok'
timestamp	string	Current timestamp

10. Health Routes

10.1 System Health

GET /health

Returns the health status of all system dependencies.

Response

Field	Type	Description
status	string	'healthy' or 'degraded'
timestamp	string	Current timestamp
services.database	string	'ok' or 'error'
services.safe	string	'ok' or 'error'

HTTP Status Codes

Code	Description
200	All services healthy
503	One or more services unavailable (status: degraded)

Appendix A: HTTP Error Codes

Code	Description
400	Bad Request - Invalid parameters or validation error
401	Unauthorized - Missing or invalid authentication token
403	Forbidden - Not authorized for this action (e.g., not Safe owner)
404	Not Found - Resource does not exist
409	Conflict - Resource already exists
429	Too Many Requests - Rate limit exceeded
500	Internal Server Error - Unexpected error
503	Service Unavailable - Dependency unavailable

Error Response Format

```
{  
  "error": "Error message description",  
  "details": [  
    { "field": "fieldName", "message": "Validation error" }  
  ]  
}
```

Appendix B: Type Definitions

Address Types

```
// Ethereum address - 40 hex characters with 0x prefix
type Address = `0x${string}` // e.g., "0x1234...5678"

// Bytes32 - 64 hex characters with 0x prefix
type Bytes32 = `0x${string}` // e.g., "0xabcd...ef01"
```

Claim Status Enum

```
enum ClaimStatus {
  Filed = 0,          // Claim submitted, evidence period active
  EvidenceClosed = 1, // Evidence period ended
  VotingClosed = 2,   // Voting period ended
  Approved = 3,       // Claim approved by council
  Rejected = 4,       // Claim rejected by council
  Executed = 5,       // Funds distributed
  Cancelled = 6,      // Cancelled by claimant
  Expired = 7,        // No quorum reached
}
```

Vote Enum

```
enum Vote {
  None = 0,    // Not voted
  Approve = 1, // Vote to approve claim
  Reject = 2,   // Vote to reject claim
  Abstain = 3  // Abstain from voting
}
```

Action Types

```
type ActionType =
  | 'create_council'
  | 'close_council'
  | 'add_member'
  | 'remove_member'
  | 'reassign_agent'
```

Author Roles

```
type AuthorRole =
  | 'claimer' // The person who filed the claim
  | 'provider' // The agent's owner
  | 'councilor' // A council member
```

Appendix C: Complete Endpoint Matrix

Legend: ✓ = Required, ✗ = Not Required

Auth: Requires Bearer token from /auth/login. Safe: Requires authenticated user to be a Safe multisig owner.

Route	Method	Auth	Safe	Reads	Writes
/auth/nonce	GET	✗	✗	-	-
/auth/login	POST	✗	✗	Chain	DB
/auth/logout	POST	✓	✗	-	DB
/auth/me	GET	✓	✗	DB, Chain	-
/auth/me	PUT	✓	✗	-	DB
/claims	GET	✗	✗	Contract	-
/claims/:id	GET	✗	✗	Contract	-
/claims/:id/votes	GET	✗	✗	Contract	-
/claims/:id/my-vote	GET	✗	✗	Contract	-
/claims/:id/vote	POST	✗	✗	Contract	-
/claims/:id/finalize	POST	✗	✗	-	-
/claims/deposit-calculator	GET	✗	✗	Contract	-
/claims/:id/metadata	GET	✗	✗	DB	-
/claims/:id/metadata	POST	✗	✗	-	DB
/claims/:id/messages	GET	✗	✗	DB	-
/claims/:id/messages	POST	✗	✗	Contract	DB
/councils	GET	✗	✗	Contract, DB	-
/councils/:id	GET	✗	✗	Contract, DB	-
/councils/:id/members	GET	✗	✗	Contract, DB	-
/councils/:id/agents	GET	✗	✗	Contract, DB	-
/councils/:id/can-close	GET	✗	✗	Contract	-
/councils/propose-create	POST	✓	✓	-	-
/councils/:id/propose-close	POST	✓	✓	Contract	-
/councils/:id/propose-add-member	POST	✓	✓	Contract	DB
/councils/:id/propose-remove-member	POST	✓	✓	Contract	-
/councils/:id/members/:addr/metadata	PUT	✓	✗	-	DB
/safe/info	GET	✗	✗	Safe API	-
/safe/pending	GET	✓	✓	Safe API	-
/safe/transactions/:hash	GET	✓	✓	Safe API	-
/safe/propose	POST	✓	✓	-	DB
/safe/sign/:hash	POST	✓	✓	Safe API	DB
/safe/execute/:hash	POST	✓	✓	Safe API	DB
/agents/:id	GET	✗	✗	Contract	-
/agents/propose-reassign	POST	✓	✓	Contract	DB
/agents/:id/reassign	POST	✓	✓	Contract	DB
/pending	GET	✓	✗	DB, Safe API	DB
/pending/:hash	GET	✓	✗	DB	-
/pending	POST	✓	✗	-	DB
/pending/:hash	DELETE	✓	✗	-	DB
/pending/sync	POST	✓	✗	DB, Safe API	DB
/provider/agents/:id	GET	✗	✗	DB	-
/provider/agents/owner/:addr	GET	✗	✗	DB	-
/provider/agents	POST	✗	✗	Contract	DB
/provider/agents/:id	PUT	✗	✗	Contract	DB
/v1/agents/:id/agent-card.json	GET	✗	✗	DB, Contract	-
/v1/agents/:id/image.svg	GET	✗	✗	DB, Contract	-
/v1/agents/:id/validation.json	GET	✗	✗	Contract	-
/v1/agents/:id/trust-info.json	GET	✗	✗	Contract	-
/v1/health	GET	✗	✗	-	-
/health	GET	✗	✗	DB, Safe API	-

