

CivCharter AI Access Grant User Guide

Internal steward/founder operating guide. Public use by founder approval only. Updated for assist-only MCP staging bridge, June 28, 2026.

Purpose

CivCharter AI Access Grants are temporary, scoped, revocable delegations from a human account to an AI guest. The AI is a guest, not an owner. It must use canonical `/api/ai/*` routes, respect granted scopes, protect credentials, and verify server responses before claiming success.

The assist-only bridge adds a safer option: AI may stage a draft civic action, but the public feed is unchanged until the human account owner opens the review link and clicks Publish Live.

Standard safe sequence

Step	Name	Required behavior
1	Bootstrap	Open the human-created grant link and exchange it for a short-lived session bearer.
2	Discovery	Call <code>/api/ai/discovery</code> to learn current scopes, canonical routes, targets, and constraints.
3	Preflight	Run <code>/api/ai/preflight</code> before any write or stewardship action when supported.
4A	Direct execute	Use only canonical <code>/api/ai/*</code> routes with Authorization: Bearer <code><session token></code> .
4B	Assist-only stage	Use <code>/api/ai/stage</code> or MCP <code>stage_civic_action</code> when a human approval loop is required.
5	Verify	Claim success only after <code>ok:true</code> and a receipt, <code>human_url</code> , <code>approval_url</code> , or verified response object.

Credential-silent behavior

Do not echo raw grant tokens, bearer/session tokens, approval tokens, or Authorization headers in ordinary chat, reports, PDFs, matrices, or restart files. If an AI cannot execute authenticated HTTP safely, it should use placeholder-free command templates, assist-only staging, or sanitized pasteback output only.

Ordinary scopes

Common scopes include `charter.read`, `feed.read`, `post.create`, `proposal.create`, `proposal.vote`, `exchange.create`, `exchange.respond`, `group.update`, `circle.create`, and `steward.manage` where enabled. Scope names describe allowed API action families. Actions outside the grant scope must be denied or preflighted as unsafe.

Assist-only MCP bridge

The assist-only bridge is the recommended path when AI should help users co-author content without directly publishing to the live civic feed. It introduces a separate `ai_staged_actions` table and a one-time human review link.

Core rule

Staging is not publication. A successful stage response means CivCharter stored a pending draft and generated a human review URL. It does not mean the post is live.

Human approval sequence

- AI receives a valid short-lived grant session bearer with `post.create` scope.
- AI calls `POST /api/ai/stage` or invokes the local FastMCP `stage_civic_action` tool.
- CivCharter validates scope, payload, citations, visibility, and target access.
- CivCharter writes `ai_staged_actions` with status `pending` and stores only `approval_token_hash`.
- AI returns the `/ai/review?token=...` approval URL to the human and states that nothing is live yet.
- Human signs in, reviews the draft, and clicks `Publish Live` or `Reject Draft`.
- `Publish Live` creates the final `CivicPost` through the existing `create_civic_post` path; rejection publishes nothing.

Canonical routes and tools

Name	Path/tool	Result
Public docs	<code>/ai/assist-only</code>	Human-readable bridge documentation.
Public primer	<code>/ai/primer</code>	No-secret AI operating primer.
Public spec	<code>/ai/spec</code>	Machine-readable route and grant model.
Stage draft	<code>POST /api/ai/stage</code>	Returns <code>approval_url</code> and <code>published_live:false</code> .
Review draft	<code>/ai/review?token=...</code>	Human approval page; token is consumed.
MCP tool	<code>mcp_bridge.py -> stage_civic_action</code>	Stages a post draft through FastMCP.

Example stage payload

```
{"action_type":"post.create","payload":{"post_type":"opinion","posture":"reflection","title":"Civic reflection","body":"Draft text for human review.","visibility":"public","citation_ids":["CHARTER_ANCHOR_UUID"]}}
```

Steward AI scope

This section applies only when `ENABLE_STEWARD_SCOPE=true` and the human grant owner is a configured founder, global steward, or database-scoped steward with active assignments.

What `steward.manage` means

The `steward.manage` scope is an elevated backend-only AI Access Grant scope. It allows an AI guest to assist a founder or steward with report review, spam scanning, and existing stewardship actions. It does not make the AI a platform owner, does not bypass human authority, and does not change the chronological feed.

Allowed steward capabilities

Capability	Canonical route
Read reports	<code>/api/ai/steward/reports</code>
Read steward feed preview	<code>/api/ai/steward/feed</code>
Lookup steward assignments	<code>/api/ai/steward/assignments</code>
Lookup steward circles	<code>/api/ai/steward/circles</code>
Safe user lookup	<code>/api/ai/steward/users</code>
Flag content	<code>/api/ai/steward/flag</code>
Hide, restore, or remove circulation	<code>/api/ai/steward/content/moderate</code>
Mute/read-only user	<code>/api/ai/steward/user/mute</code>
Unmute user	<code>/api/ai/steward/user/unmute</code>
Review report	<code>/api/ai/steward/report/review</code>
Add steward note	<code>/api/ai/steward/note</code>

Mandatory steward workflow

The AI must call `/api/ai/discovery` first, run `/api/ai/preflight` before steward writes, use only canonical `/api/ai/steward/*` routes, and verify `ok:true` before claiming an action succeeded. Browser routes such as `/steward` are for humans and must not be used with bearer-token AI requests.

Hard limits

Steward AI may not change feed ranking or ordering. CivCharter remains chronological, not algorithmic. Steward AI may not expose secrets, raw user emails, raw IPs, private exports, bearer tokens, grant tokens, approval tokens, or Authorization headers. Steward actions are auditable and should use the least restrictive effective action.

Operator checklist

- Before schema/database patches, run the standard external backup and a direct SQLite backup.
- Confirm the patch does not replace .env, instance/, live SQLite files, run.py, UO server files, or hosted game asset directories.
- Smoke-test /, /ai/primer, /ai/spec, /ai/assist-only, and the PDF guide path.
- Confirm /api/ai/discovery advertises action.stage when post.create is granted.
- Confirm /api/ai/stage returns ok:true, approval_url, docs_url, assist_only_mode:true, and published_live:false.
- Confirm no CivicPost appears before approval.
- Confirm Publish Live creates one post and consumes the token.
- Confirm Reject Draft consumes the token and creates no post.
- Confirm FastMCP uses CIVCHARTER_AI_BEARER locally and never stores that bearer in source files.

Language for AI responses

Correct: "I staged this draft for human review. Nothing is live yet. Please open the approval link and choose Publish Live or Reject Draft."

Incorrect: "I posted this to CivCharter" when the only completed step was /api/ai/stage.

Scaling rule

Future action enums such as proposal.create, exchange.create, group.update, or steward decision drafts should reuse the same tokenized pending-record approval loop before any direct public write is considered. Each new staged action type should define required scope, payload validation, preview rendering, approval action, rejection action, audit log entry, and token consumption behavior.