

NexDirectory Scanner (Chrome Extension)

Comprehensive documentation for the event-plugin-wp-scanner extension.

1. Overview

NexDirectory Scanner is a Chrome side-panel extension that helps you extract event/listing data from web pages and import it directly into NexDirectory.

Primary use cases:

- Import a single event from a source page in minutes.
- Run bulk imports from a URL list.
- Keep imports aligned with your plugin defaults.
- Optionally use AI extraction and translation.

2. Core Features

Scan tab:

- Scans current page content and metadata.
- Extracts title, dates, venue, description, media URLs, contact fields.
- Supports required import fields (title and start).
- Supports dynamic custom/form fields.

Bulk tab:

- Paste one URL per line.
- Processes sequentially.
- Tracks progress, successes, failures.
- Optional translation support.

Settings tab:

- Connect to WordPress via site URL + Extension App Pass.
- Configure AI model and API key.
- Set import defaults and behavior toggles.

Logs tab:

- Local log viewer with copy/clear.

Help tab:

- Setup reminders and documentation link.

3. Architecture

Extension components:

- sidepanel.js – UI and interactions

- background.js – API calls, signing, bulk engine

Endpoints:

- /wp-json/eep/v1/extension/ping
- /wp-json/eep/v1/extension/import
- /wp-json/eep/v1/extension/preferences

Flow:

1. Ping validates connection and fetches schema.
2. Page scan (heuristic + optional AI).
3. User review/edit.
4. Signed import request.
5. Plugin validates and creates listing.

4. Requirements

- Chrome with Manifest V3 support
- NexDirectory plugin installed
- Connector enabled
- Extension App Pass generated
- Optional OpenAI key

5. Installation (Chrome)

1. Open chrome://extensions/
2. Enable Developer mode
3. Click Load unpacked
4. Select chrome-extension/event-plugin-wp-scanner
5. Pin extension
6. Open side panel

6. WordPress Setup

Enable Chrome Extension Connector in NexDirectory settings.

Copy Extension App Pass into extension.

Configure defaults: status, media policy, content mode, ownership, claimable, notifications, disclaimer.

If App Pass is regenerated, update extension immediately.

7. Extension Setup

Set Plugin Site URL.

Paste Extension App Pass.

Optional: OpenAI key and model (default gpt-5.2).

Test connection.
Save settings.

8. Quick Start

Single import:

1. Open source page
2. Scan
3. Verify fields
4. Import
5. Open listing

Bulk import:

1. Paste URLs
2. Optional translation
3. Start
4. Monitor progress

9. Security Model

Headers:

- X-EEP-App-Pass
- X-EEP-Timestamp
- X-EEP-Signature

Signature: HMAC-SHA256 over "." using App Pass key.

Server checks:

- Connector enabled
- App Pass match
- Valid timestamp window
- Valid signature

Recommendations:

- Treat App Pass as password
- Regenerate if leaked
- Use HTTPS only

10. REST API Reference

Ping: handshake, returns defaults and schema.

Preferences: sync extension defaults to plugin.

Import: creates/imports listing, validates fields, enforces policies, duplicate checks, optional notifications.

11. Troubleshooting

Connection fails: verify URL and App Pass.

Invalid signature: check clock and credentials.

Missing title/start: edit before import.

OpenAI fails: verify key and model.

Bulk stuck: check logs and restart with smaller batch.

12. Best Practices

- Use draft status for QA.
- Enable duplicate detection.
- Test small before bulk.
- Use verbatim for fidelity.
- Use rewrite only when AI configured.

13. FAQ

OpenAI required? No.

Multiple sites? Yes, update URL and App Pass.

Regenerate App Pass? Reconfigure extension.

Where do imports go? NexDirectory post type.

Selective notifications? Yes, per import.

14. File Map

- manifest.json
- sidepanel.html
- sidepanel.css
- sidepanel.js
- background.js
- images/