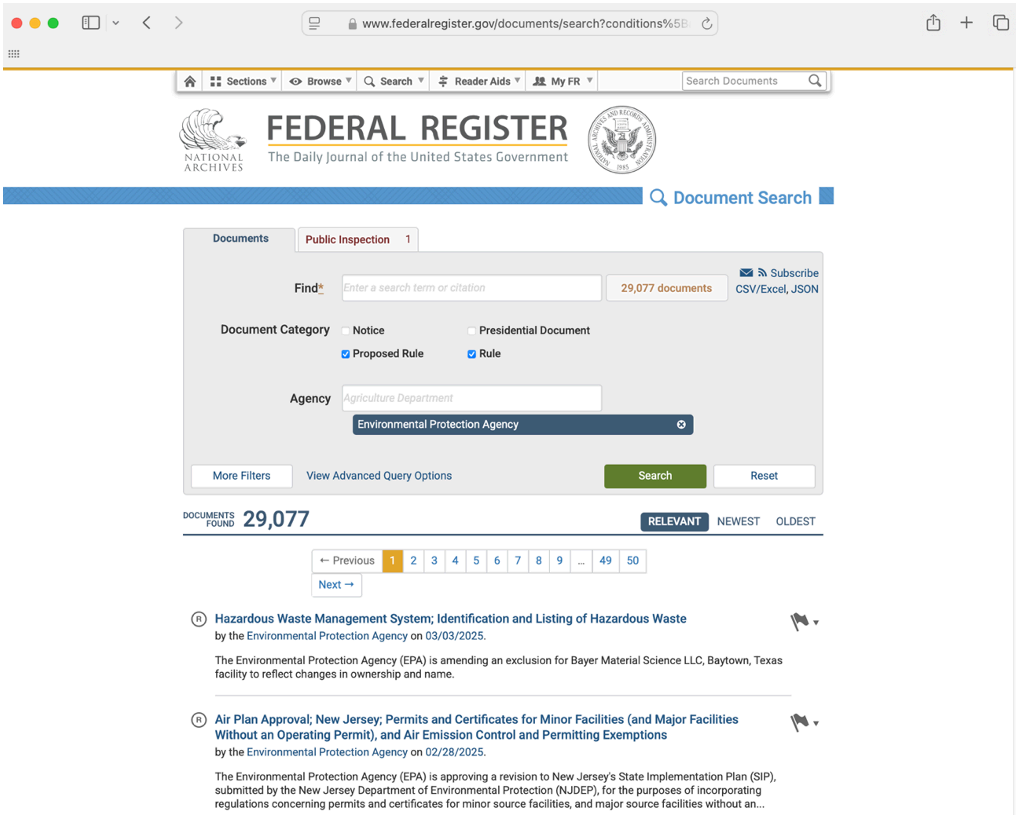
Drupal Feeds Using LLMs

If you regularly bring data into Drupal from external sources, you may be able to enrich the data using LLMs and a bit of custom programming. This repo demos a workflow utilizing Python scripts running on a Linux server, web scraping, LLMs and the Drupal Feeds module. Your tech stack may be different but the techniques discussed in this talk are easily translated to other environments.

The repo demonstrates two scrapers, one that interacts with the Federal Register API and the other which scrapes an EPA web page. The scrapers output JSON suitable for Drupal feeds import.

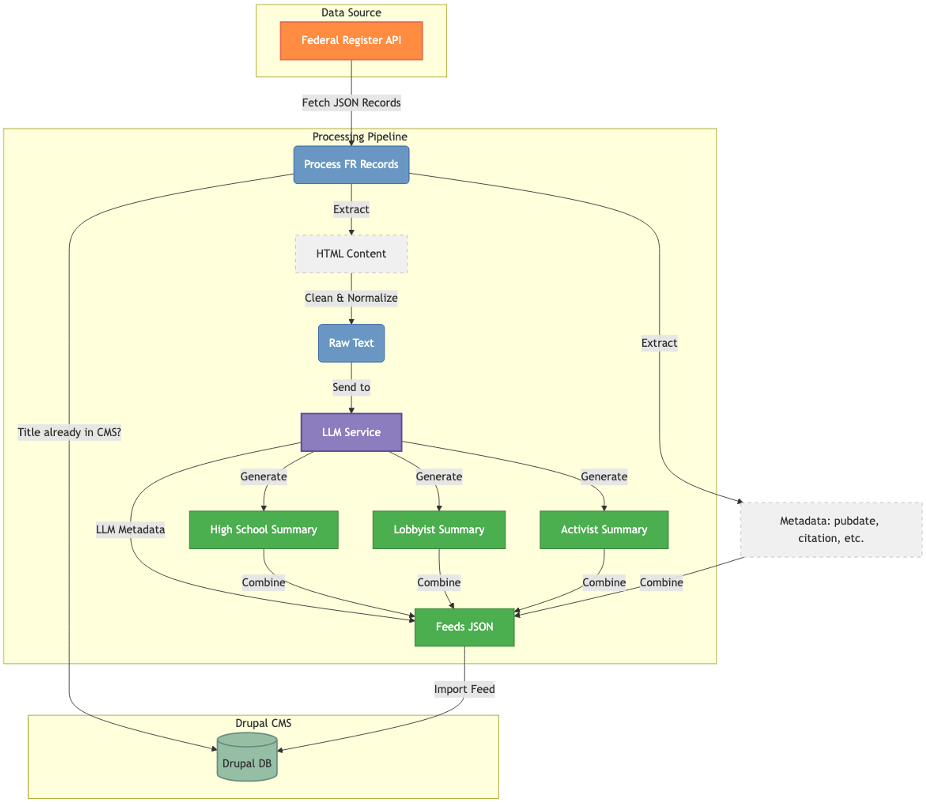
Federal Register



- Site: https://federalregister.gov
- Background: This is where the U.S. government publishes all proposed federal regulations or rules about to go into effect. The site has an excellent UI and is also known for its developer-friendly API and JSON output.
- Goals: connect to Federal Register API and get JSON metadata for recently-published rules. Then:
    - Make minor transformations to their field data for our matching fields in Drupal, e.g. prepending a citation to their title to match our Drupal node title best practices
    - Add 3 LLM-generated summaries individually focused on the needs of: 1) high school students, 2) corporate lobbyists, 3) environmental activists.

Federal Register processing pipeline

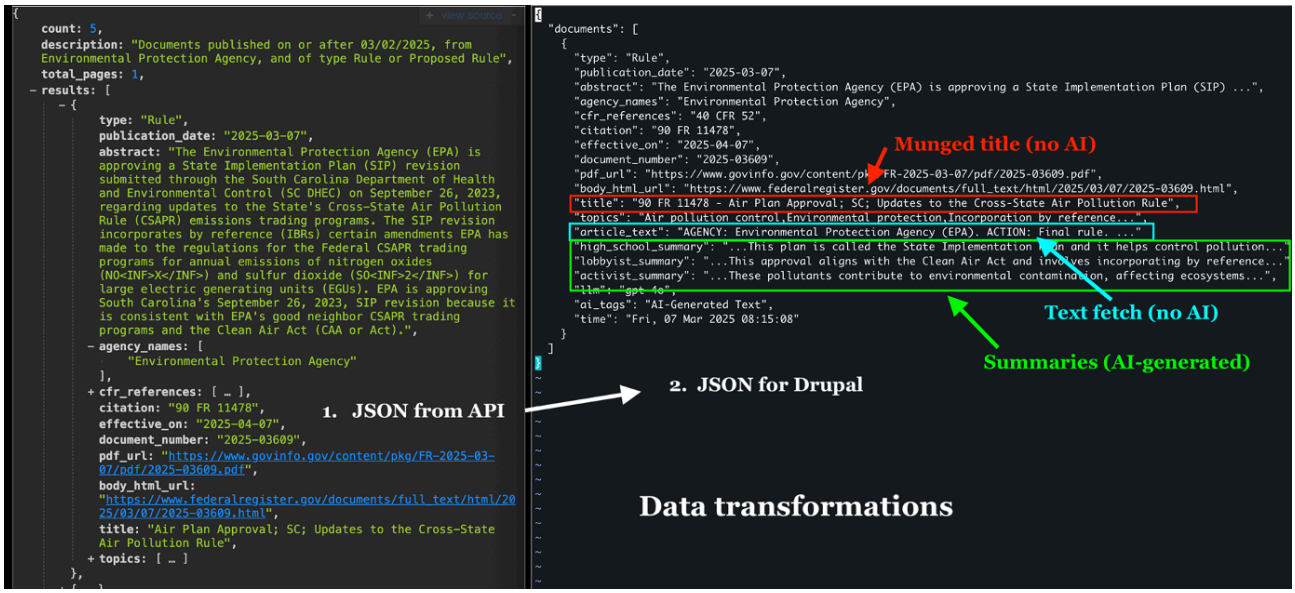
High level overview of the data processing game plan:



Inputs and Outputs

Federal Register developer interactive API: https://www.federalregister.gov/developers/documentation/api/v1#/Federal%20Register%20Documents/get_documents__format_

Federal Register JSON request for the EPA: https://www.federalregister.gov/api/v1/documents.json?fields[]=abstract&fields[]=type&fields[]=pdf_url&fields[]=document_number&fields[]=publication_date&fields[]=agencies&fields[]=title&per_page=3&conditions[publication_date][gte]=2025-03-01&conditions[agencies][]=environmental-protection-agency&conditions[type][]=RULE&conditions[type][]=PRORULE

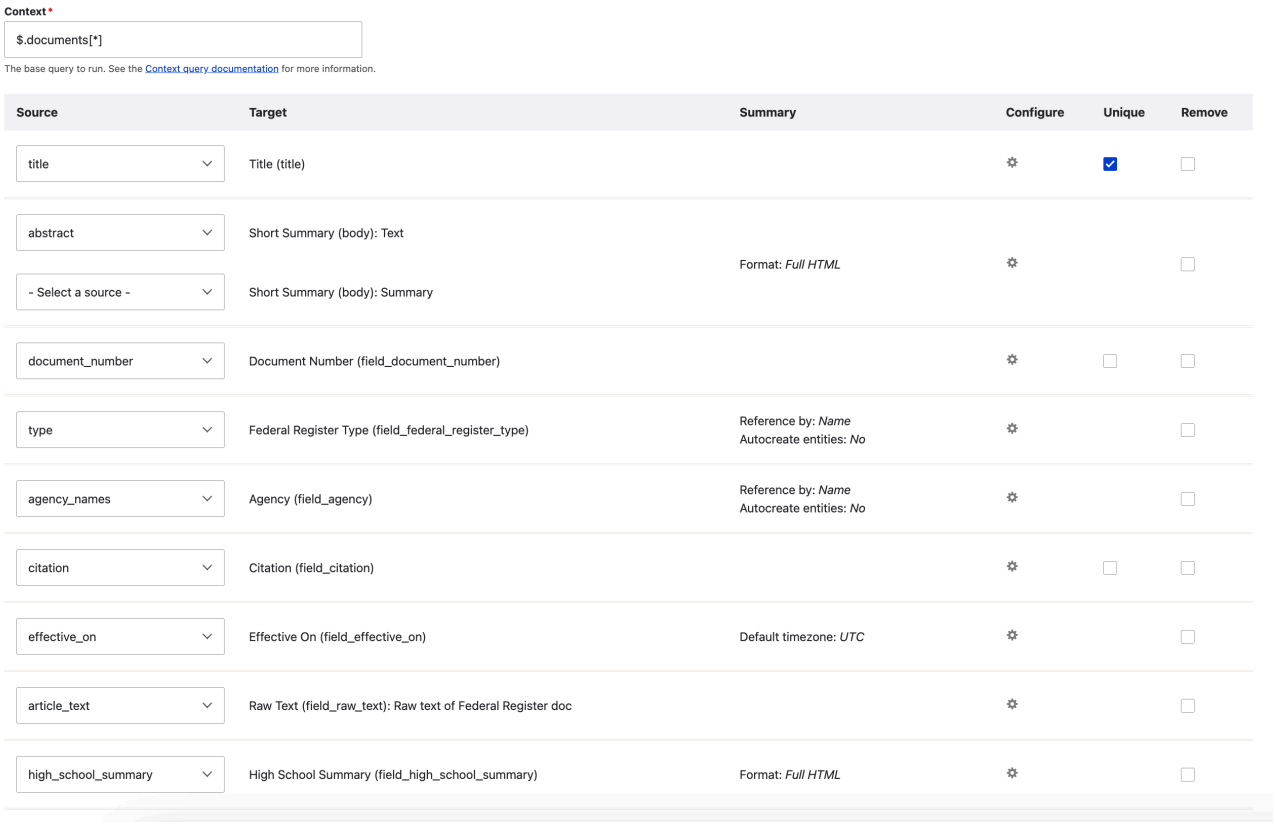


Drupal

On the Drupal side of things we have a content type Federal Register that has matching fields:

Field	Widget
Title	Textfield
Short Summary	Text area with a summary
High School Summary	Text area (multiple rows)
Activist Summary	Text area (multiple rows)
Lobbyist Summary	Text area (multiple rows)
Document Number	Textfield
Agency	Check boxes/radio buttons
Federal Register Type	Check boxes/radio buttons
Effective On	Date and time

And our Federal Register Feed that will consume our JSON output looks like this:



News Meta-summary

We can also have our script create a News summary that iterates over all the Federal Registers we are importing and ask the LLM to summarize all the other summaries, a kind of meta-summary. This should not be published as is – that's "cheating" – but it could certainly form the skeleton on which a human editor could build. Imagining we are using a simple Drupal "Article" content type we might see:

Environmental Protection Agency Regulatory Review from February 28, 2025 to March 07, 2025

LLM-generated meta-summary - a human should edit!

14 March 2025

The EPA is actively engaged in various air quality and hazardous waste management actions, as evidenced by recent Federal Register notices. Several actions involve State Implementation Plan (SIP) revisions, including approvals for South Carolina (90 FR 11478, 2025-03-07) concerning the Cross-State Air Pollution Rule (CSAPR) and New Jersey (90 FR 10872, 2025-02-28) regarding permits for minor and major facilities. These revisions aim to reduce emissions of nitrogen oxides (NOX) and sulfur dioxide (SO2) and regulate stationary sources to meet National Ambient Air Quality Standards (NAAQS). Activists should **scrutinize** whether these regulations go far enough to protect **public health** and **ecosystems**, especially for **vulnerable populations**. It's crucial to **advocate** for stronger emission reduction targets and increased **monitoring** to ensure **accountability**.

Other actions include proposed approvals for Alabama's SIP revision (90 FR 11500, 2025-03-07), which modifies the definition of "volatile organic compounds" (VOC), and a reopened comment period for Ohio's regional haze plan (90 FR 10876, 2025-02-28) focusing on permit conditions for power plants. The Alabama revision warrants careful examination of the **long-term effects** of added non-VOCs on **public health** and the **environment**. The Ohio plan provides an opportunity to **advocate** for stronger **environmental protections** and **challenge** permit conditions that may not adequately protect **public health** and the **environment** from **emissions** contributing to **air pollution** and **acid rain**.

The EPA is also addressing hazardous waste management, including the deletion of sites from the Superfund National Priorities List (NPL) (90 FR 11218, 2025-03-05) and an amendment reflecting a name change for a facility in Baytown, TX (90 FR 11025, 2025-03-03). While these actions may seem like progress or administrative updates, it's essential to **scrutinize** the details and ensure continued **monitoring** to prevent future risks to **public health** or **ecosystems**. Activists should **demand** that facilities adhere to the strictest standards to prevent **environmental contamination** and protect **soil health**, paying close attention to potential impacts on **vulnerable populations** and **environmental justice** concerns.

Finally, the EPA has reopened the comment period for the proposed regulation addressing the risks associated with C.I. Pigment Violet 29 (PV29) (90 FR 11142, 2025-03-04). This presents an opportunity for **environmental** activists to **scrutinize** the proposed measures and **voice concerns** regarding potential **environmental** and **public health** impacts. It is important to **advocate** for stronger regulations to protect **vulnerable populations** and the **environment**, focusing on potential **environmental contamination**, impacts on **ecosystems**, and risks to **public health**. **Demand** greater **transparency** and **accountability** in the regulation of PV29 and **push for** stricter **emissions** controls and **monitoring** to minimize potential harm.

Sample Federal Register

After importing here's what a sample FR looks like:

90 FR 10876 - Air Plan Approval; Ohio; Regional Haze Plan for the Second Implementation Period

28 February 2025

Direct from Federal Register JSON

Short Summary

The Environmental Protection Agency (EPA) is reopening the public comment period for 15 days for a proposed rule published August 30, 2024. Reopening the public comment period is focused exclusively on the references to various permit conditions within three Director's Final Findings and Orders (DFFOs) included in the docket that were issued by the Ohio Environmental Protection Agency (Ohio EPA) as those conditions apply to the limits in the DFFOs for Cardinal Power Plant, Ohio Valley Electric Corp.--Kyger Creek, and General James M. Gavin Power Plant.

Agency

AI classification

Environmental Protection Agency

High School Summary

The Environmental Protection Agency (EPA) is giving the public another chance to share their thoughts on a proposed rule about air quality in Ohio. This rule is part of a larger plan to reduce haze and improve visibility in national parks and wilderness areas. The EPA wants feedback specifically on some permit conditions for three power plants: Cardinal Power Plant, Ohio Valley Electric Corp.—Kyger Creek, and General James M. Gavin Power Plant. These permits set limits on how much pollution these plants can release. The EPA initially asked for comments on this rule, but some important information about the permits wasn't available at the time. Now that this information is available, the EPA is reopening the comment period so people can review it and share their opinions. This is important because it affects the air we breathe and the health of our environment.

Lobbyist Summary

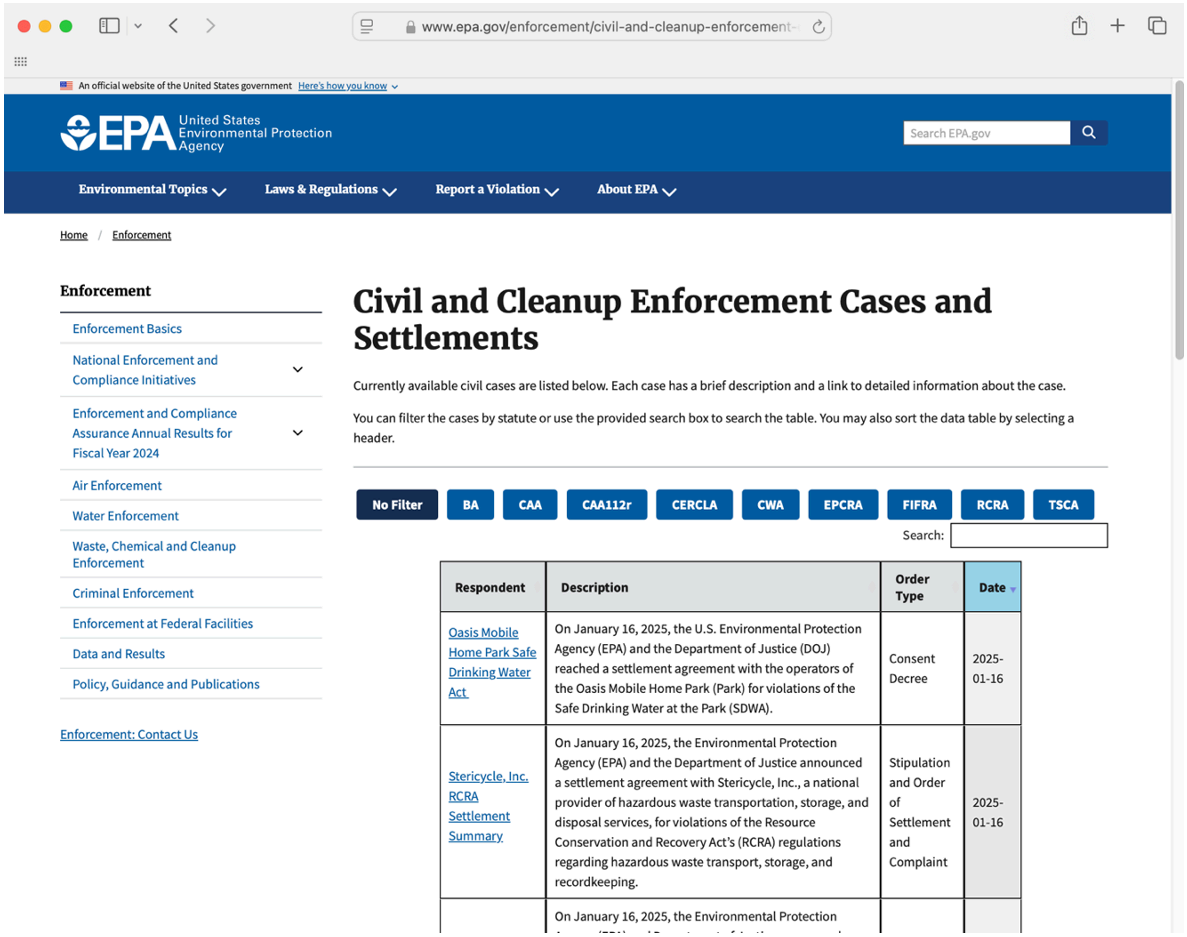
AI summarization

The EPA has reopened the comment period for a proposed rule concerning Ohio's Regional Haze State Implementation Plan (SIP). This reopening focuses exclusively on permit conditions within Director's Final Findings and Orders (DFFOs) for Cardinal Power Plant, Ohio Valley Electric Corp.—Kyger Creek, and General James M. Gavin Power Plant. The permit conditions, initially unavailable, pertain to emission limits outlined in the DFFOs. **Of interest to lobbyists:** The EPA seeks comments on these specific permit conditions as they relate to the limits in the DFFOs. **Actionable items:** Review the newly available permit conditions for the Cardinal, Kyger Creek, and Gavin power plants. **Prepare and submit comments** addressing the permit conditions' impact on the proposed rule. Understand the potential **compliance** implications for the affected power plants. Assess potential **costs** associated with meeting the permit conditions. Evaluate the **economic implications** of the proposed rule and permit conditions on the power sector in Ohio. **Opportunities to engage** with the EPA during the reopened comment period to **influence the final rule**.

Activist Summary

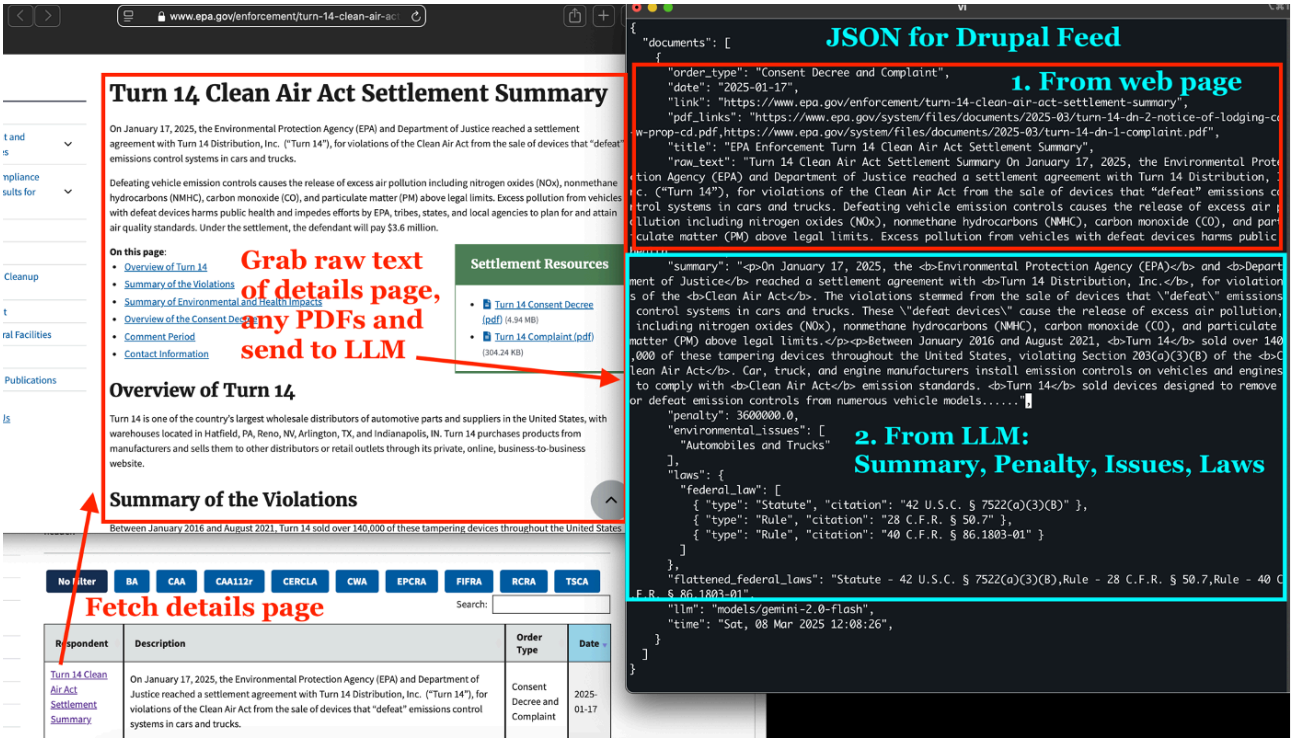
The EPA is providing a second opportunity to comment on a proposed rule regarding Ohio's plan to reduce regional haze. This renewed comment period is centered on specific permit conditions for three power plants: Cardinal Power Plant, Ohio Valley Electric Corp.—Kyger Creek, and General James M. Gavin Power Plant. These permits dictate the allowable pollution levels for these facilities. The EPA initially sought public input, but crucial permit details were missing. Now that these details are accessible, the EPA is reopening the comment period.

Civil Enforcement Actions by the EPA

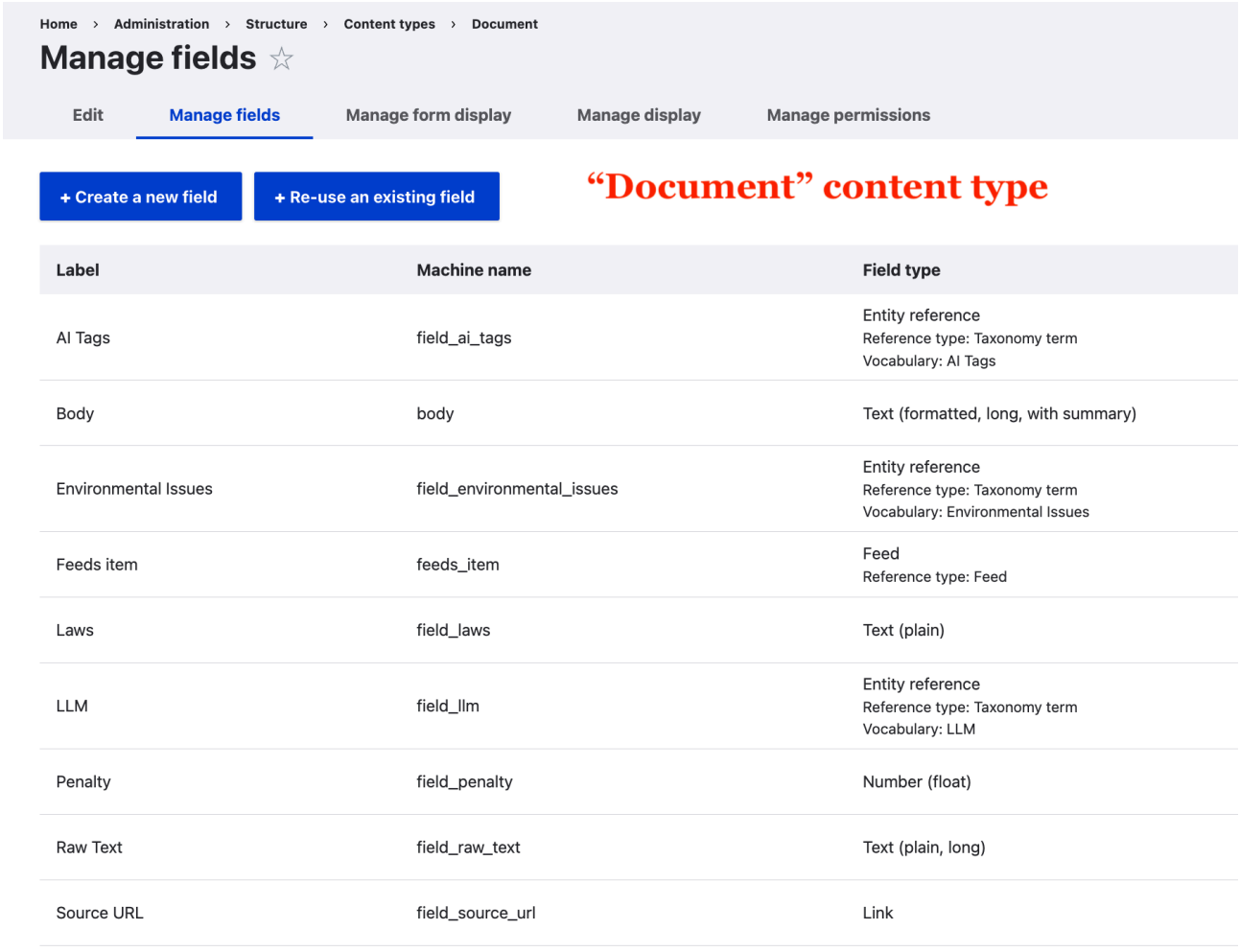


- Site: https://www.epa.gov/enforcement/civil-and-cleanup-enforcement-cases-and-settlements
- Background: Centralized source for EPA civil actions for violations. The EPA [historically] enforces a wide range of environmental laws, including the Clean Air Act (CAA) and Clean Water Act (CWA).
- Goals:
  - Better summaries
  - Extract penalty info
  - What specific laws were violated?
  - Categorize environmental issues

Inputs and Outputs



Document content type we'll be using for the EPA Civil Actions and its associated Feed



Home › Administration › Structure › Content types › Document

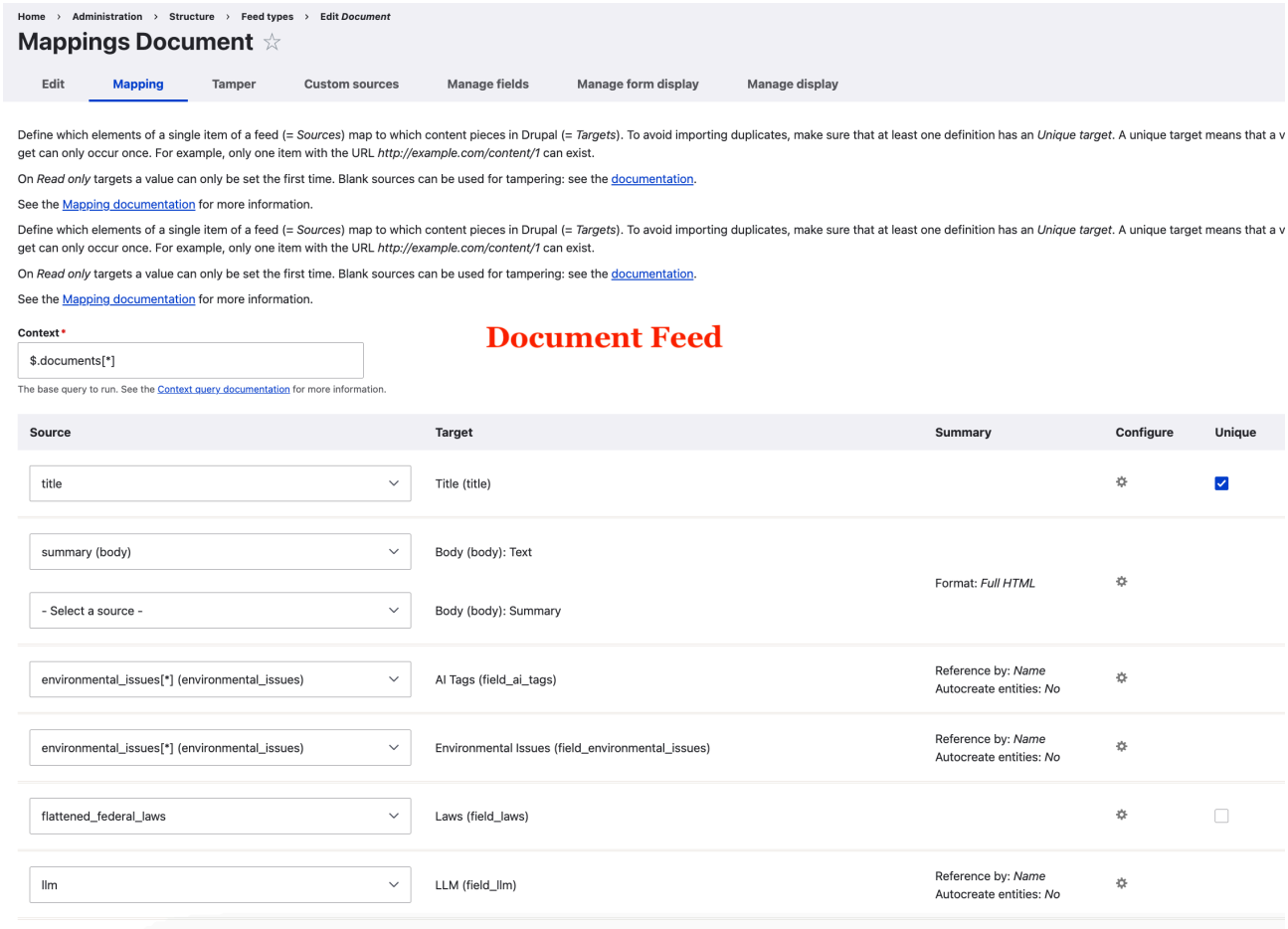
Manage fields ☆

Edit | Manage fields | Manage form display | Manage display | Manage permissions

+ Create a new field | + Re-use an existing field

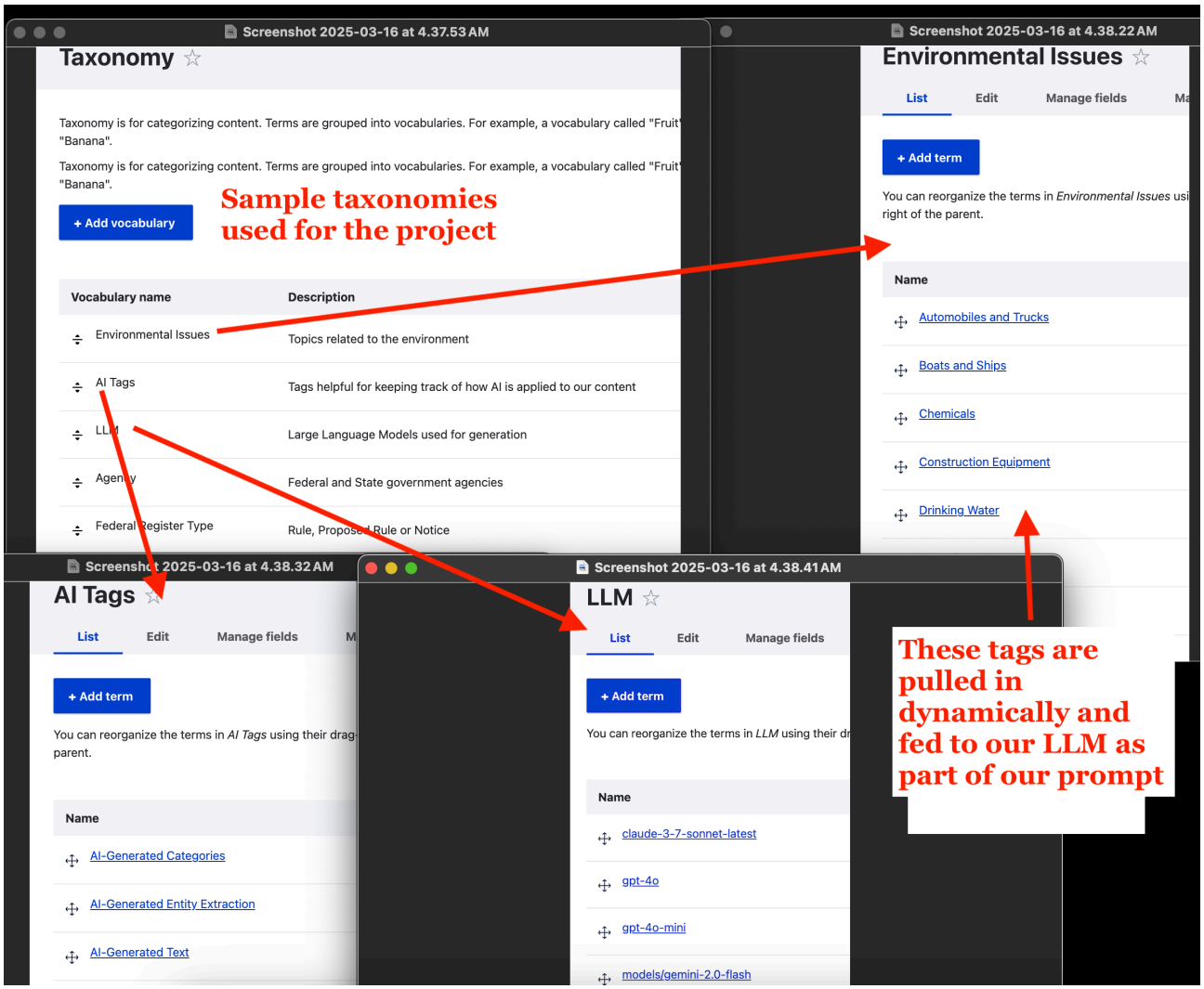
"Document" content type

Label	Machine name	Field type
AI Tags	field_ai_tags	Entity reference Reference type: Taxonomy term Vocabulary: AI Tags
Body	body	Text (formatted, long, with summary)
Environmental Issues	field_environmental_issues	Entity reference Reference type: Taxonomy term Vocabulary: Environmental Issues
Feeds item	feeds_item	Feed Reference type: Feed
Laws	field_laws	Text (plain)
LLM	field_llm	Entity reference Reference type: Taxonomy term Vocabulary: LLM
Penalty	field_penalty	Number (float)
Raw Text	field_raw_text	Text (plain, long)
Source URL	field_source_url	Link

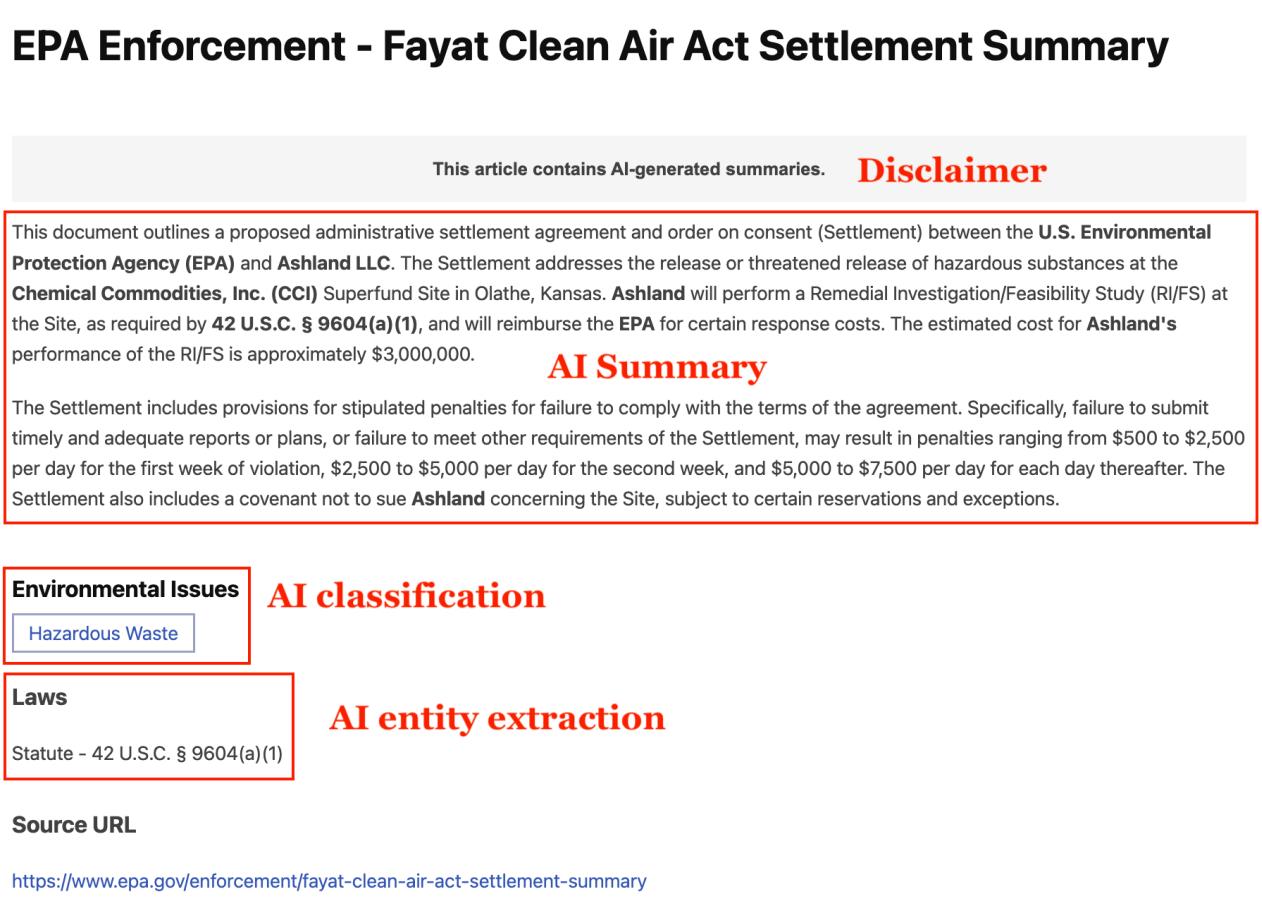


Drupal Taxonomies

Here are the sample vocabularies and terms used in the project. Note that we dynamically import the Environmental Issues tags to our Python code at runtime, add them to our Pydantic model and use them as part of our prompt to the LLM to shape its behavior (“categorize documents according to these tags, and only these tags”).


Screenshot 2025-03-16 at 4.37.53 AM
Taxonomy
Taxonomy is for categorizing content. Terms are grouped into vocabularies. For example, a vocabulary called "Fruit" "Banana".
Taxonomy is for categorizing content. Terms are grouped into vocabularies. For example, a vocabulary called "Fruit" "Banana".
+ Add vocabulary
Sample taxonomies used for the project
Vocabulary name
Description
Environmental Issues
Topics related to the environment
AI Tags
Tags helpful for keeping track of how AI is applied to our content
LLM
Large Language Models used for generation
Agency
Federal and State government agencies
Federal Register Type
Rule, Proposed Rule or Notice
Screenshot 2025-03-16 at 4.38.22 AM
Environmental Issues
List
Edit
Manage fields
+ Add term
You can reorganize the terms in Environmental Issues
right of the parent.
Name
Automobiles and Trucks
Boats and Ships
Chemicals
Construction Equipment
Drinking Water
Screenshot 2025-03-16 at 4.38.32 AM
AI Tags
List
Edit
Manage fields
+ Add term
You can reorganize the terms in AI Tags using their drag
parent.
Name
AI-Generated Categories
AI-Generated Entity Extraction
AI-Generated Text
Screenshot 2025-03-16 at 4.38.41 AM
LLM
List
Edit
Manage fields
+ Add term
You can reorganize the terms in LLM using their dr
Name
claude-3-7-sonnet-latest
gpt-4o
gpt-4o-mini
models/gemini-2.0-flash
These tags are pulled in dynamically and fed to our LLM as part of our prompt

Example civil action Document import:



EPA Enforcement - Fayat Clean Air Act Settlement Summary

This article contains AI-generated summaries. Disclaimer

This document outlines a proposed administrative settlement agreement and order on consent (Settlement) between the **U.S. Environmental Protection Agency (EPA)** and **Ashland LLC**. The Settlement addresses the release or threatened release of hazardous substances at the **Chemical Commodities, Inc. (CCI)** Superfund Site in Olathe, Kansas. **Ashland** will perform a Remedial Investigation/Feasibility Study (RI/FS) at the Site, as required by **42 U.S.C. § 9604(a)(1)**, and will reimburse the **EPA** for certain response costs. The estimated cost for **Ashland's** performance of the RI/FS is approximately $3,000,000.

AI Summary

The Settlement includes provisions for stipulated penalties for failure to comply with the terms of the agreement. Specifically, failure to submit timely and adequate reports or plans, or failure to meet other requirements of the Settlement, may result in penalties ranging from $500 to $2,500 per day for the first week of violation, $2,500 to $5,000 per day for the second week, and $5,000 to $7,500 per day for each day thereafter. The Settlement also includes a covenant not to sue **Ashland** concerning the Site, subject to certain reservations and exceptions.

Environmental Issues

Hazardous Waste

AI classification

Laws

Statute - 42 U.S.C. § 9604(a)(1)

AI entity extraction

Source URL

https://www.epa.gov/enforcement/fayat-clean-air-act-settlement-summary

LLM Considerations

Closed models, open source – they're constantly leapfrogging one another. At present, closed source foundation models produce more **reliable structured output**. This may change in the future.

The code here has options for OpenAI's "gpt-4o", Anthropic's "claude-3-7-sonnet-latest" and Google's "gemini-2.0-flash". Currently we default to **Gemini Flash** because it is fast, reliable, inexpensive (as of March 2025: input $0.10 / 1,000,000 input tokens, $0.40 / 1,000,000 output tokens) and has a long context window (1,000,000 tokens) that can readily accommodate multi-hundred page PDFs.

Optimization – a custom title_exists() function checks to see if we already have a node in Drupal; if so, we avoid making unnecessary LLM calls. This is essential when performing web scraping at scale.

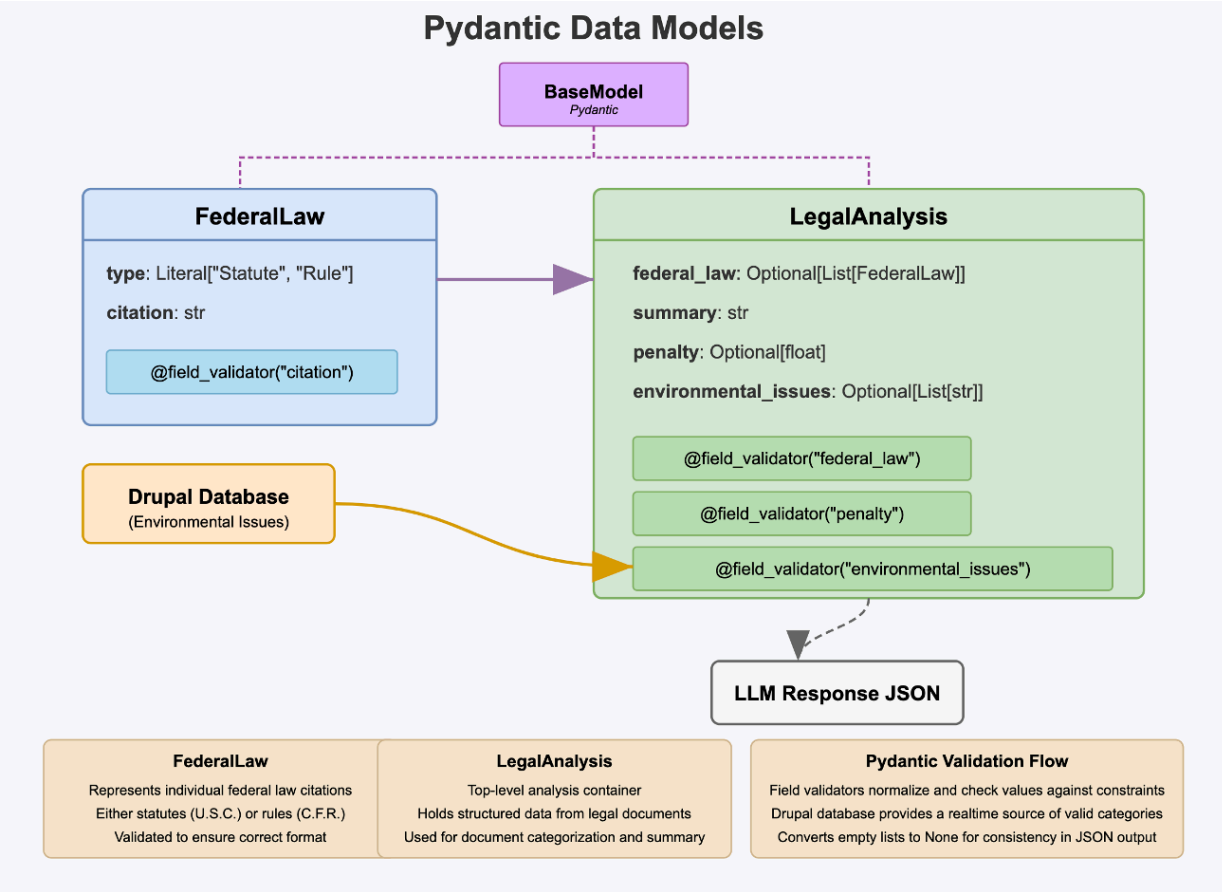
Pydantic and Instructor

Two Python libraries aid greatly in steering and validating the output of LLMs:

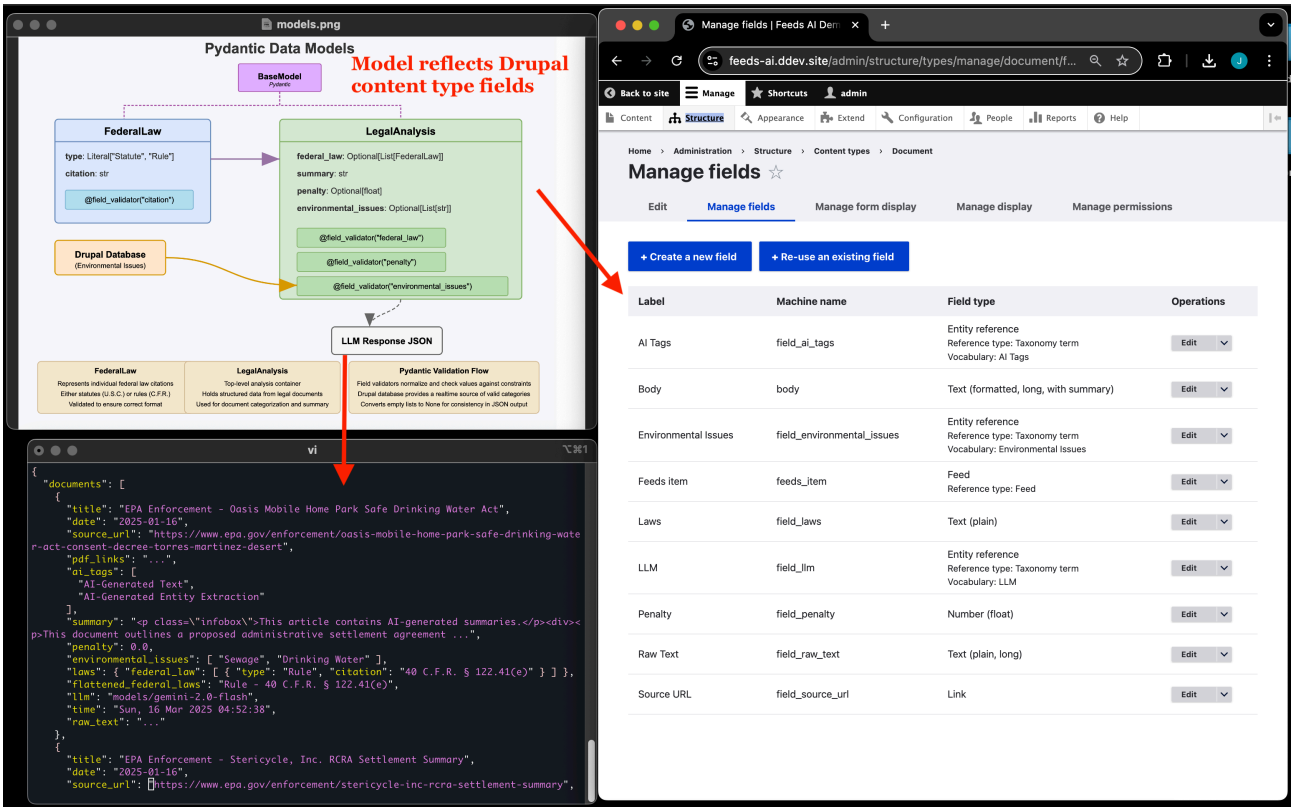
- Pydantic ensures that data meets certain criteria before being sent downstream. It's used in many contexts, not just with LLMs. For example, in validating inputs to an API or outputs to a database. A key feature for LLM usage is that Pydantic allows you to define a "data model" with rich annotations suitable for use with the Instructor library.
- Instructor helps steer LLMs toward a desired behavior. In particular, you can take a deeply annotated Pydantic model and use it to provide detailed prompt instructions to an LLM, then verify that the return data from the LLM conforms to your expectations.

These libraries help us to tell LLMs precisely how we want them to structure their reply, e.g. JSON with particular fields and types of data; and validate the data before passing it on for downstream use. In particular, we want anything GenAI-related to pass muster before importing it to Drupal. Drupal, which is all about structured content, does well by partnering with a robust framework for getting structured output from LLMs.

Here are the models used for the EPA scraper:



And an overview of how the Pydantic models, Drupal content type and LLM output JSON work together:



Pydantic Field validators

These are optional functions you can write that automatically run against fields before Pydantic signs off on the data. You can use them to simply confirm that data matches certain criteria, or you can modify the data to match your needs. The point is that this is a structured way of enforcing standards for your data.

What we're choosing to validate:

- **citation** – make sure legal citations are in a standardized format, e.g. 40 C.F.R. §§ Part 1039" should be transformed to "40 C.F.R. § 1039
- **penalty** – make sure it's a numeric float with at most 2 decimal places
- **environmental_issues** – suppose we have a Drupal taxonomy we're already using. We can fetch the terms dynamically from the Drupal DB and use them both to tell the LLM what terms we want it to look for and then to make sure that's what it did.

Miscellaneous "Best Practices"

There are few "standards" yet for how websites should manage AI-generated content. You may want to consider:

- Creating a vocabulary "AI Tags" to help you keep track of nodes to which you've applied GenAI. For example: AI-Generated Text, AI-Generated Categories, or AI-Generated Entity Extraction
- It is quite likely that you'll use different LLMs over time. You may want to have a vocabulary for these as well with terms such as: claude-3-7-sonnet-latest, gpt-4o, gpt-4o-mini, gemini-2.0-flash
- If you're going to the trouble of extracting lots of raw text for use in GenAI, even though in its raw state it may not be suitable for end users it might be good for a) fulltext search; b) future GenAI passes over the same nodes
- People are justifiably wary of what's being pushed on them – consider including preamble / disclaimer text such as "This article contains an AI-generated summary" that fully informs people what they're getting.

An important additional consideration is tracking Drupal LLM usage. Drupal is noteworthy precisely because of its structured approach to data, including fields, and we may want to respect (and take advantage of) that affordance of granularity when considering an LLM usage audit strategy. It's possible that Drupal isn't the right place to perform auditing and bookkeeping and that it'll be easier to use an external database. Companies in the future are likely to have external LLMOps systems that are used for more assets than just Drupal so that might be the way to go.

- In the examples presented here note that we're using LLMs at the Drupal field level - whereas the "AI Tags" as applied in these examples are assigned at the node level. Having field-level tags for tracking AI may be a bit much to do within Drupal itself and an external auditing database may be a better bet.
- MLOps is an emerging discipline. Keeping track of what content in Drupal has been "touched" by LLMs raises many questions. For any given Drupal field there are multiple things you may want to track. Given that the behavior of LLMs is complicated, varies over time and model version, the prompt, etc. here is a minimum you might consider tracking:
  - which LLM model and version did you use, e.g. OpenAI gpt-4o, Anthropic Claude 3.7, etc.
  - the actual text of the prompt you used - rather than repeat this text over and over you might use a taxonomy
  - token usage
  - date / time you ran the model
- You might also consider storing the raw text blobs from PDFs externally as well so the Drupal DB doesn't bloat unnecessarily).

Drupal demo

You can create the simple demo site that corresponds to the scraping demo per https://github.com/criolho/Drupal-Feeds-Using-LLMs/blob/master/feeds-ai/README.md.

To Run the Scrapers

Note that these are barebones scrapers that only pull content from the first page of results. A fully-fledged scraping process would involve paging. That isn't the focus of this repo but the underlying code can readily be extended to accommodate paging.

1. You need Python 3.12 or higher
2. Create a new virtual env
3. Install requirements
4. Set env variables, e.g. use a .env file. You'll also need an API key for whatever LLMs you choose to you. Then set an env variable such as OPENAI_KEY, GEMINI_KEY, ANTHROPIC_KEY, etc.
5. Run scrapers

```
# Example .env file:
DB_HOST=127.0.0.1
```

```
DB_PORT=49677
DB_USER=db
DB_PASSWORD=db
DB_NAME=db


# Set up virtual environment, activate it and install prerequisites:
python -m venv feeds_scraper
source feeds_scraper/bin/activate
pip install -r requirements.txt

# Run fr scraper for agency = EPA (see config/agencies.json file)
retrieve Federal Registers dating 3/1/25 or later and generate a news
summary.

python fr.py -a epa -d 2025-03-01 -n

# Grab first 5 rows of EPA enforcement actions
python epa.py -numrecs 5
```