SOLI Town Hall #000

Date: Friday, September 6, 2024
Time: 2:00 – 3:00pm (UTC-4)
Recording URL: (pending)

SOLI

Agenda

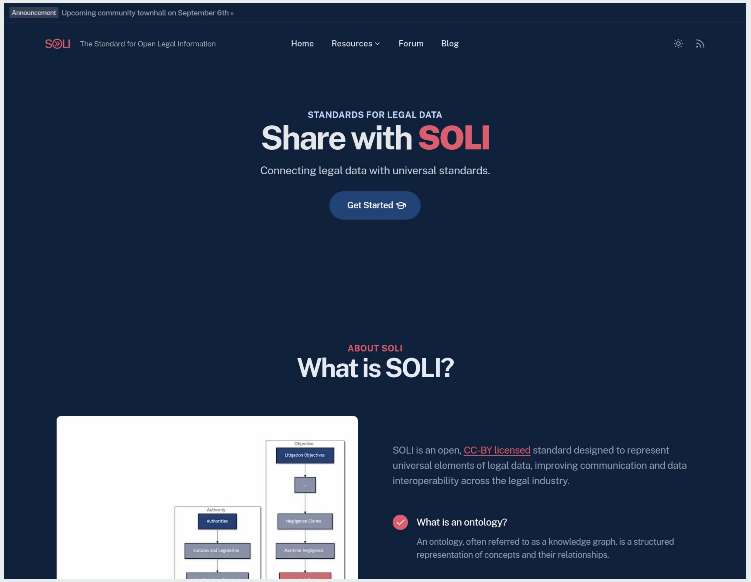
Topic	Timing (minutes)
Welcome and Intro	5
Project Context	5
SOLI Overview	5-10
Current Status	15
Future Roadmap	15
Open Q&A	-

SOLI

Welcome and Intro

Welcome and Introduction (5 minutes)

- Brief overview of SOLI and its mission
- Introduction of key team members



Announcement Upcoming community townhall on September 6th »

The Standard for Open Legal Information

STANDARDS FOR LEGAL DATA

Share with SOLI

Connecting legal data with universal standards.

ABOUT SOLI

What is SOLI?

SOLI is an open, CC-BY licensed standard designed to represent universal elements of legal data, improving communication and data interoperability across the legal industry.

✓ What is an ontology?

An ontology, often referred to as a knowledge graph, is a structured representation of concepts and their relationships.

SOLI

Project Context

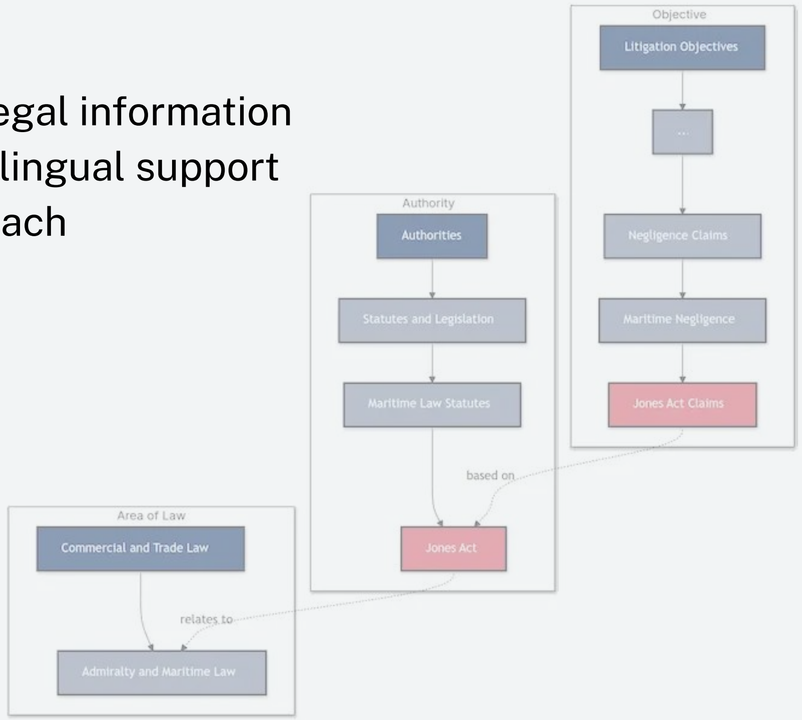
Project Background and Context (5 minutes)

- Discuss the motivation behind SOLI's creation
- Address the relationship with prior projects
- Highlight lessons learned and how SOLI aims to overcome previous challenges

SOLI Overview

SOLI Project Overview (5-10 minutes)

- SOLI as an open, integrative standard for legal information
- Comprehensive tags with unique IDs, multilingual support
- Discuss the open, community-driven approach



SOLI

Current Status

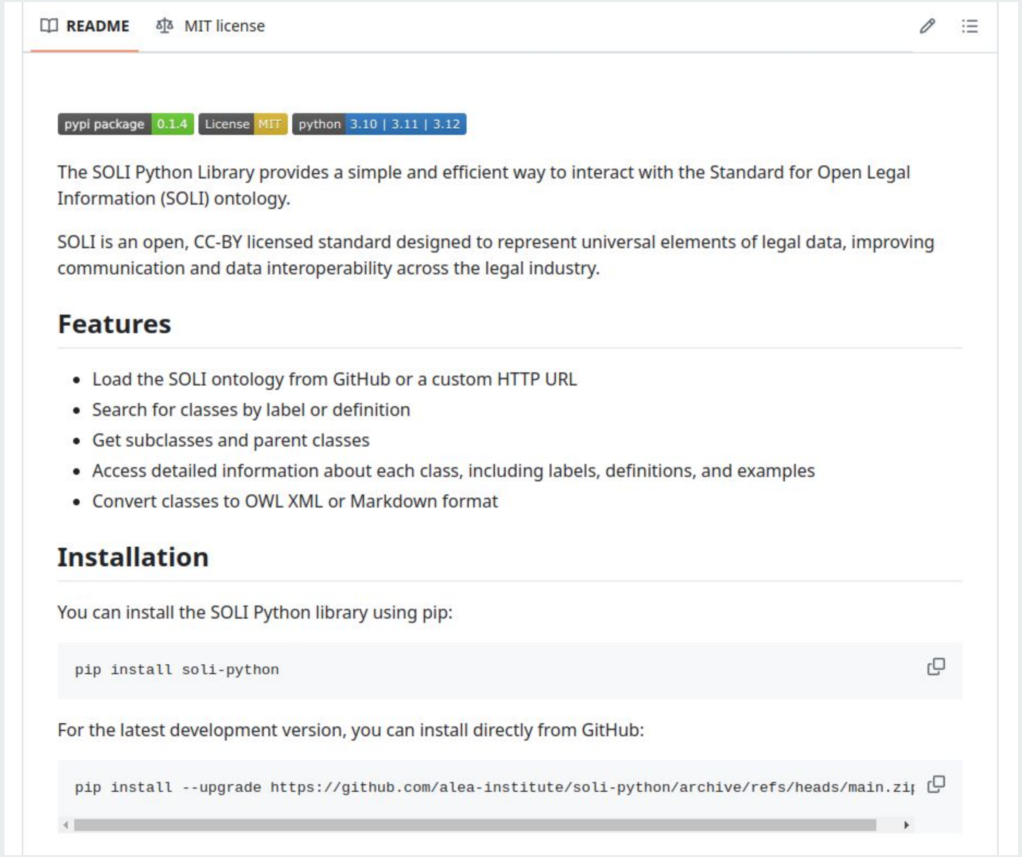
SOLI

Current Status and Released Projects (15 minutes)

- Introduce and briefly demonstrate recently launched resources:
  - SOLI Python Library
  - SOLI API
    - Source code and Docker image availability
    - Publicly-hosted instance at https://soli.openlegalstandard.org/
  - SOLI Data Generation Library
  - SOLI Annotation Library (soon)
- Discuss additional topics for Discourse forum
- Discuss initial adoption, usage stats by class/taxonomic area, and community feedback

SOLI Python Library

https://github.com/alea-institute/soli-python


README MIT license

pypi package 0.1.4 License MIT python 3.10 | 3.11 | 3.12

The SOLI Python Library provides a simple and efficient way to interact with the Standard for Open Legal Information (SOLI) ontology.

SOLI is an open, CC-BY licensed standard designed to represent universal elements of legal data, improving communication and data interoperability across the legal industry.

Features

- Load the SOLI ontology from GitHub or a custom HTTP URL
- Search for classes by label or definition
- Get subclasses and parent classes
- Access detailed information about each class, including labels, definitions, and examples
- Convert classes to OWL XML or Markdown format

Installation

You can install the SOLI Python library using pip:

```
pip install soli-python
```

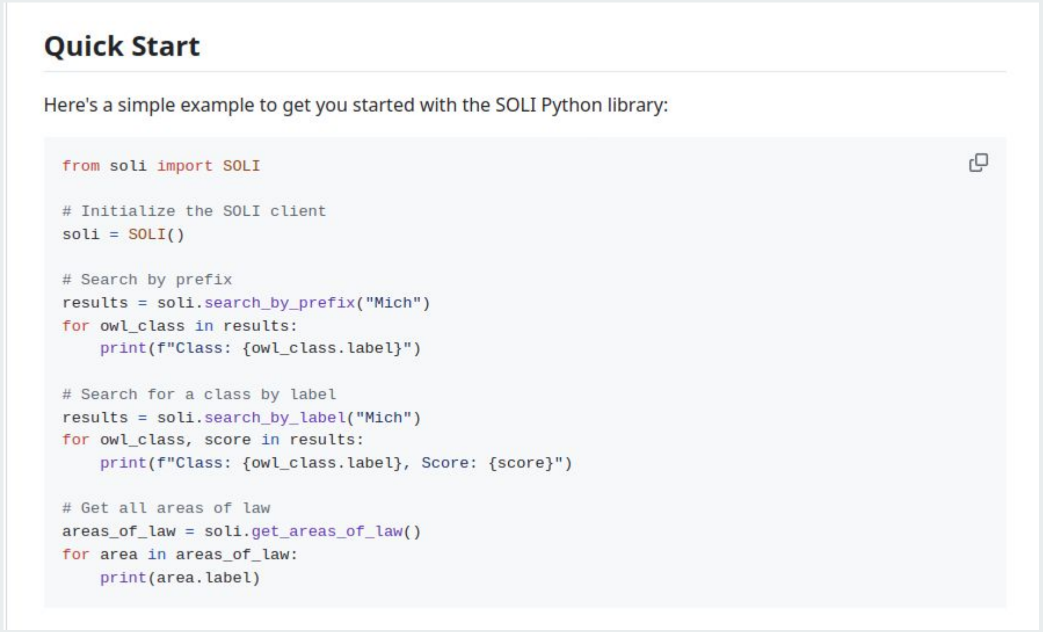
For the latest development version, you can install directly from GitHub:

```
pip install --upgrade https://github.com/alea-institute/soli-python/archive/refs/heads/main.zi
```

SOLI

SOLI Python Library

https://github.com/alea-institute/soli-python



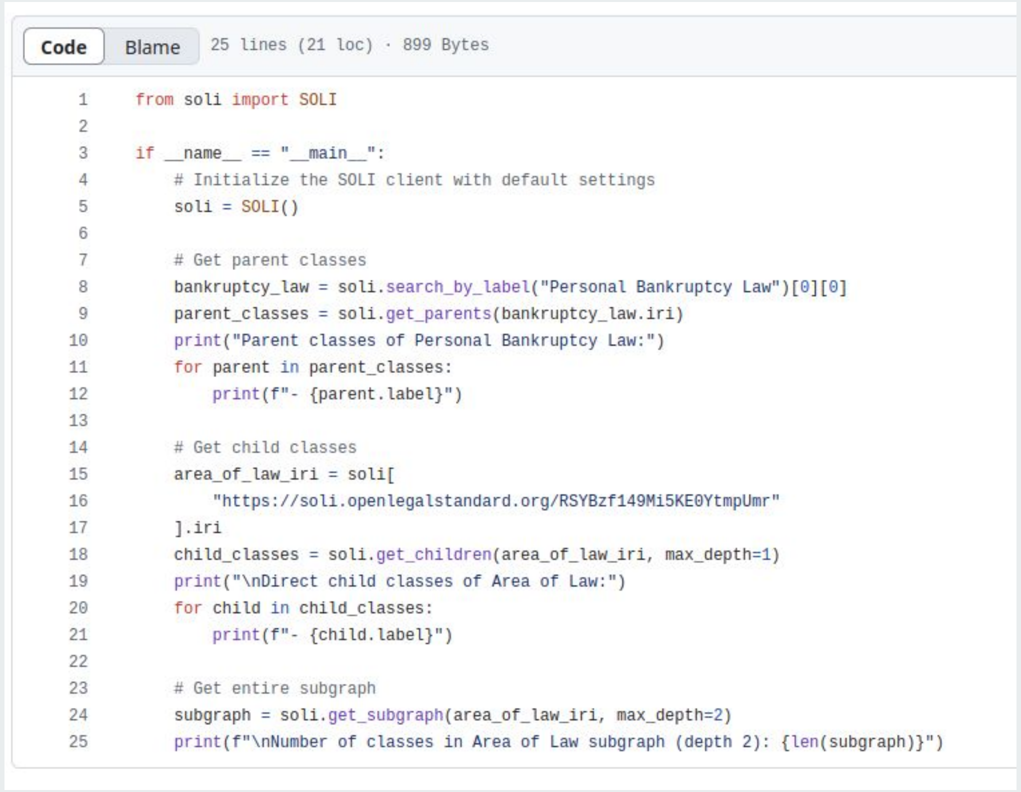
Quick Start

Here's a simple example to get you started with the SOLI Python library:

```python
from soli import SOLI

# Initialize the SOLI client
soli = SOLI()

# Search by prefix
results = soli.search_by_prefix("Mich")
for owl_class in results:
    print(f"Class: {owl_class.label}")

# Search for a class by label
results = soli.search_by_label("Mich")
for owl_class, score in results:
    print(f"Class: {owl_class.label}, Score: {score}")

# Get all areas of law
areas_of_law = soli.get_areas_of_law()
for area in areas_of_law:
    print(area.label)
```

SOLI Python Library

https://github.com/alea-institute/soli-python

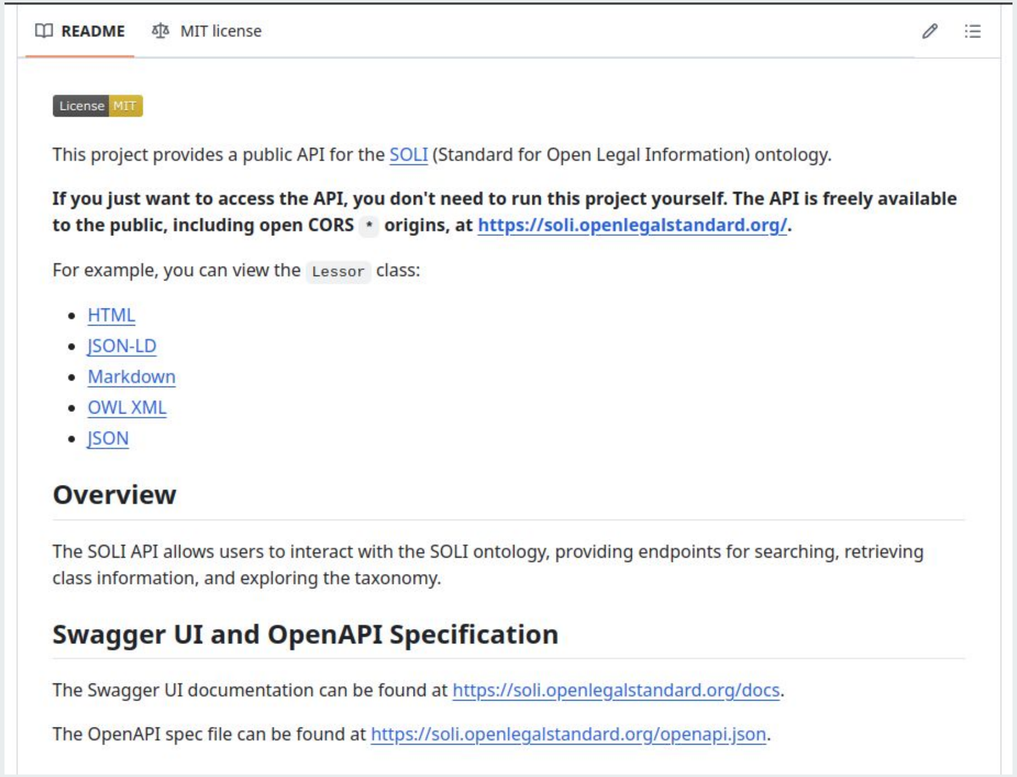


Code Blame 25 lines (21 loc) · 899 Bytes

```python
from soli import SOLI

if __name__ == "__main__":
    # Initialize the SOLI client with default settings
    soli = SOLI()

    # Get parent classes
    bankruptcy_law = soli.search_by_label("Personal Bankruptcy Law")[0][0]
    parent_classes = soli.get_parents(bankruptcy_law.iri)
    print("Parent classes of Personal Bankruptcy Law:")
    for parent in parent_classes:
        print(f"- {parent.label}")

    # Get child classes
    area_of_law_iri = soli[
        "https://soli.openlegalstandard.org/RSYBzf149Mi5KE0YtmpUmr"
    ].iri
    child_classes = soli.get_children(area_of_law_iri, max_depth=1)
    print("\nDirect child classes of Area of Law:")
    for child in child_classes:
        print(f"- {child.label}")

    # Get entire subgraph
    subgraph = soli.get_subgraph(area_of_law_iri, max_depth=2)
    print(f"\nNumber of classes in Area of Law subgraph (depth 2): {len(subgraph)}")
```

SOLI API

https://github.com/alea-institute/soli-api



README MIT license

License MIT

This project provides a public API for the SOLI (Standard for Open Legal Information) ontology.

If you just want to access the API, you don't need to run this project yourself. The API is freely available to the public, including open CORS `*` origins, at https://soli.openlegalstandard.org/.

For example, you can view the `Lessor` class:

- HTML
- JSON-LD
- Markdown
- OWL XML
- JSON

Overview

The SOLI API allows users to interact with the SOLI ontology, providing endpoints for searching, retrieving class information, and exploring the taxonomy.

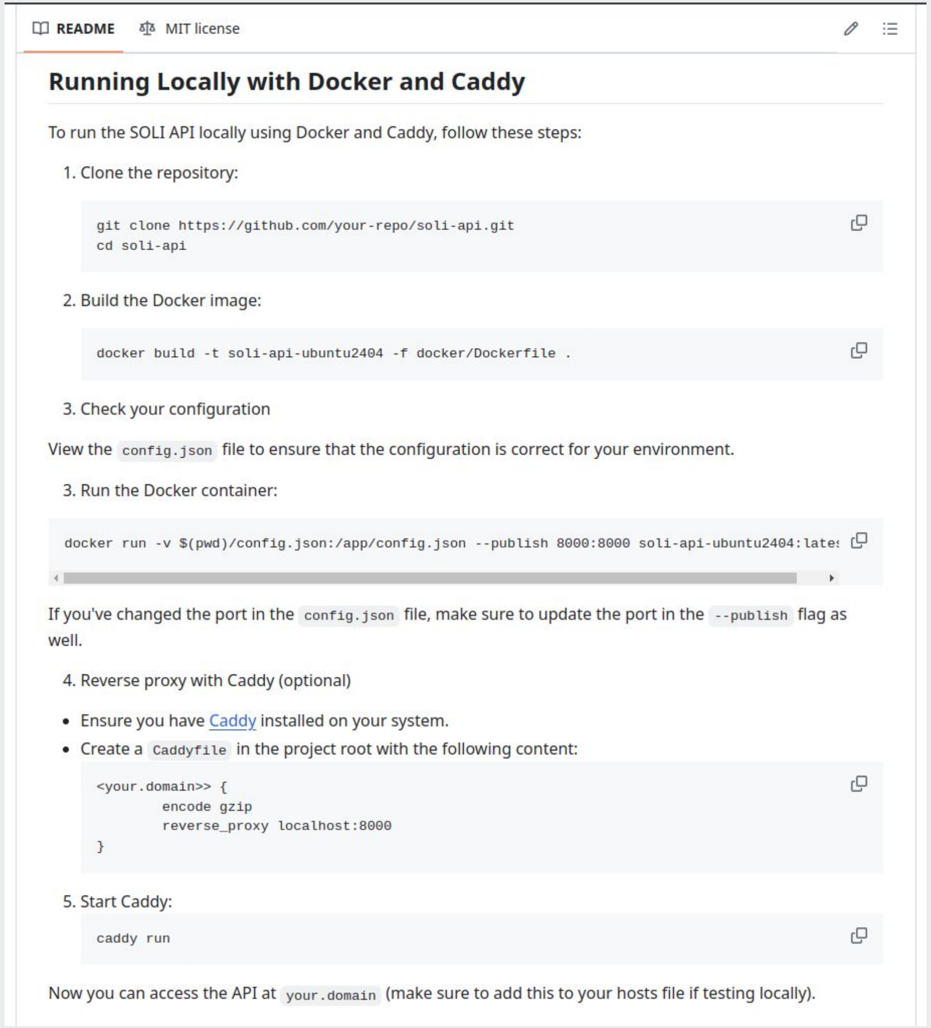
Swagger UI and OpenAPI Specification

The Swagger UI documentation can be found at https://soli.openlegalstandard.org/docs.

The OpenAPI spec file can be found at https://soli.openlegalstandard.org/openapi.json.

SOLI API

https://github.com/alea-institute/soli-api


README MIT license

Running Locally with Docker and Caddy

To run the SOLI API locally using Docker and Caddy, follow these steps:

1. Clone the repository:

```
git clone https://github.com/your-repo/soli-api.git
cd soli-api
```

2. Build the Docker image:

```
docker build -t soli-api-ubuntu2404 -f docker/Dockerfile .
```

3. Check your configuration

View the `config.json` file to ensure that the configuration is correct for your environment.

3. Run the Docker container:

```
docker run -v $(pwd)/config.json:/app/config.json --publish 8000:8000 soli-api-ubuntu2404:lates
```

If you've changed the port in the `config.json` file, make sure to update the port in the `--publish` flag as well.

4. Reverse proxy with Caddy (optional)

- Ensure you have Caddy installed on your system.
- Create a `Caddyfile` in the project root with the following content:

```
<your.domain>> {
        encode gzip
        reverse_proxy localhost:8000
}
```

5. Start Caddy:

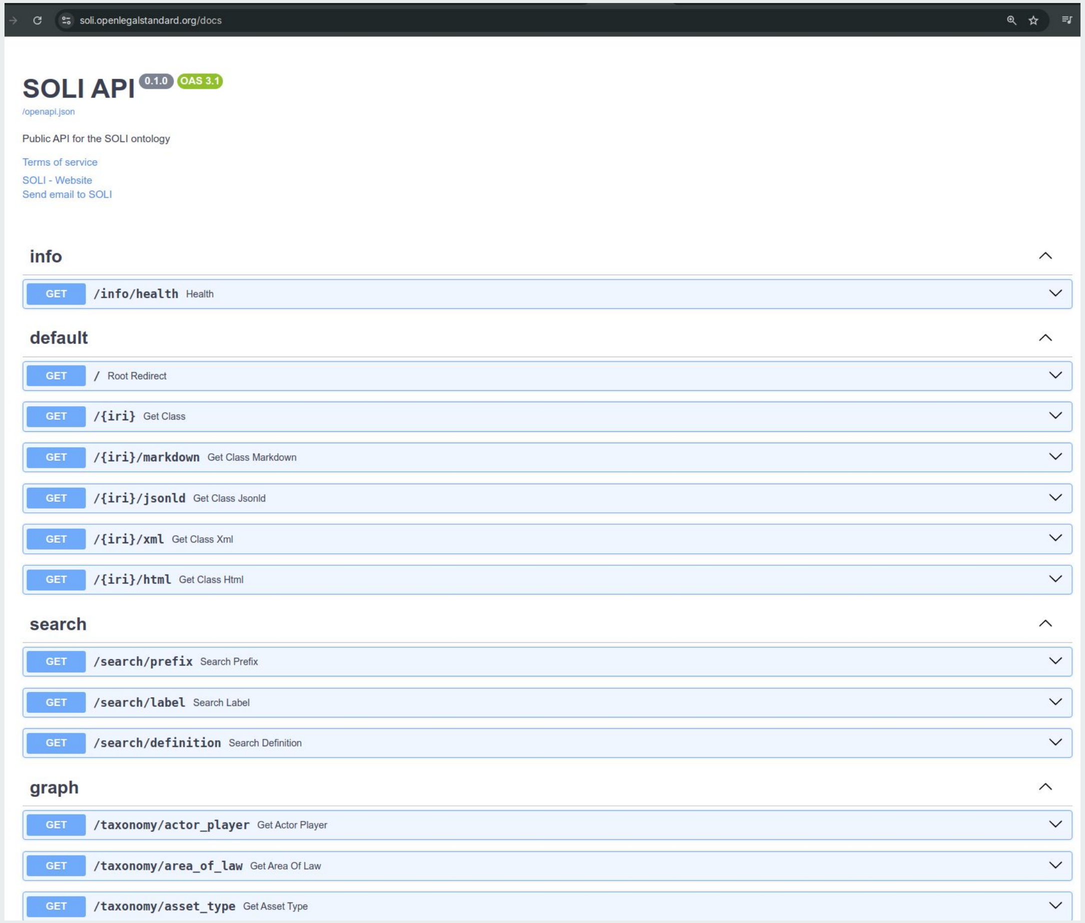
```
caddy run
```

Now you can access the API at `your.domain` (make sure to add this to your hosts file if testing locally).

SOLI

SOLI API

https://soli.openlegalstandard.org/docs



soli.openlegalstandard.org/docs

SOLI API 0.1.0 OAS 3.1

/openapi.json

Public API for the SOLI ontology

Terms of service

SOLI - Website

Send email to SOLI

info

GET /info/health Health

default

GET / Root Redirect

GET /{iri} Get Class

GET /{iri}/markdown Get Class Markdown

GET /{iri}/jsonld Get Class Jsonld

GET /{iri}/xml Get Class Xml

GET /{iri}/html Get Class Html

search

GET /search/prefix Search Prefix

GET /search/label Search Label

GET /search/definition Search Definition

graph

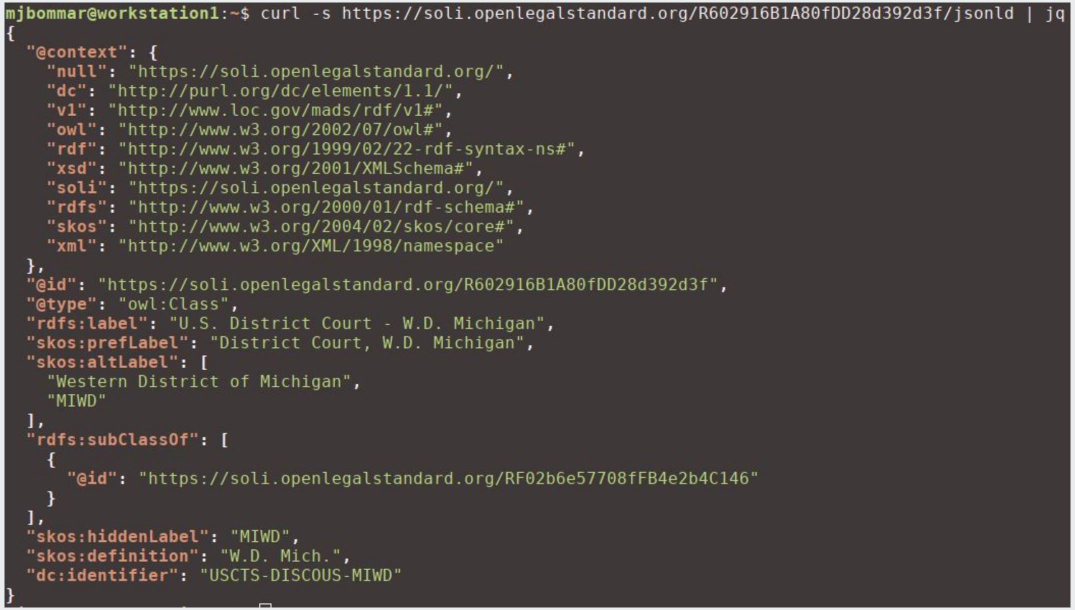
GET /taxonomy/actor_player Get Actor Player

GET /taxonomy/area_of_law Get Area Of Law

GET /taxonomy/asset_type Get Asset Type

SOLI API

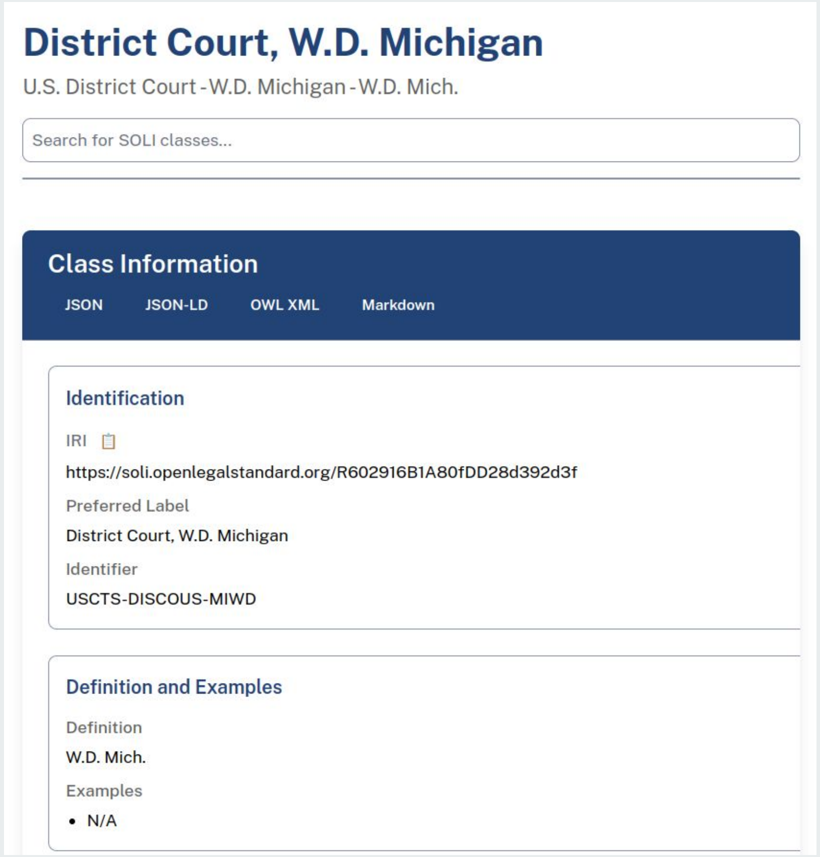
https://soli.openlegalstandard.org/docs



```
mjbommar@workstation1:~$ curl -s https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f/jsonld | jq
{
  "@context": {
    "null": "https://soli.openlegalstandard.org/",
    "dc": "http://purl.org/dc/elements/1.1/",
    "v1": "http://www.loc.gov/mads/rdf/v1#",
    "owl": "http://www.w3.org/2002/07/owl#",
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "soli": "https://soli.openlegalstandard.org/",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "xml": "http://www.w3.org/XML/1998/namespace"
  },
  "@id": "https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f",
  "@type": "owl:Class",
  "rdfs:label": "U.S. District Court - W.D. Michigan",
  "skos:prefLabel": "District Court, W.D. Michigan",
  "skos:altLabel": [
    "Western District of Michigan",
    "MIWD"
  ],
  "rdfs:subClassOf": [
    {
      "@id": "https://soli.openlegalstandard.org/RF02b6e57708fFB4e2b4C146"
    }
  ],
  "skos:hiddenLabel": "MIWD",
  "skos:definition": "W.D. Mich.",
  "dc:identifier": "USCTS-DISCOUS-MIWD"
}
```

SOLI API

https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f
https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f/html
https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f/xml
https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f/jsonld
https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f/markdown


District Court, W.D. Michigan

U.S. District Court - W.D. Michigan - W.D. Mich.

Identification

IRI
https://soli.openlegalstandard.org/R602916B1A80fDD28d392d3f

Preferred Label
District Court, W.D. Michigan

Identifier
USCTS-DISCOUS-MIWD

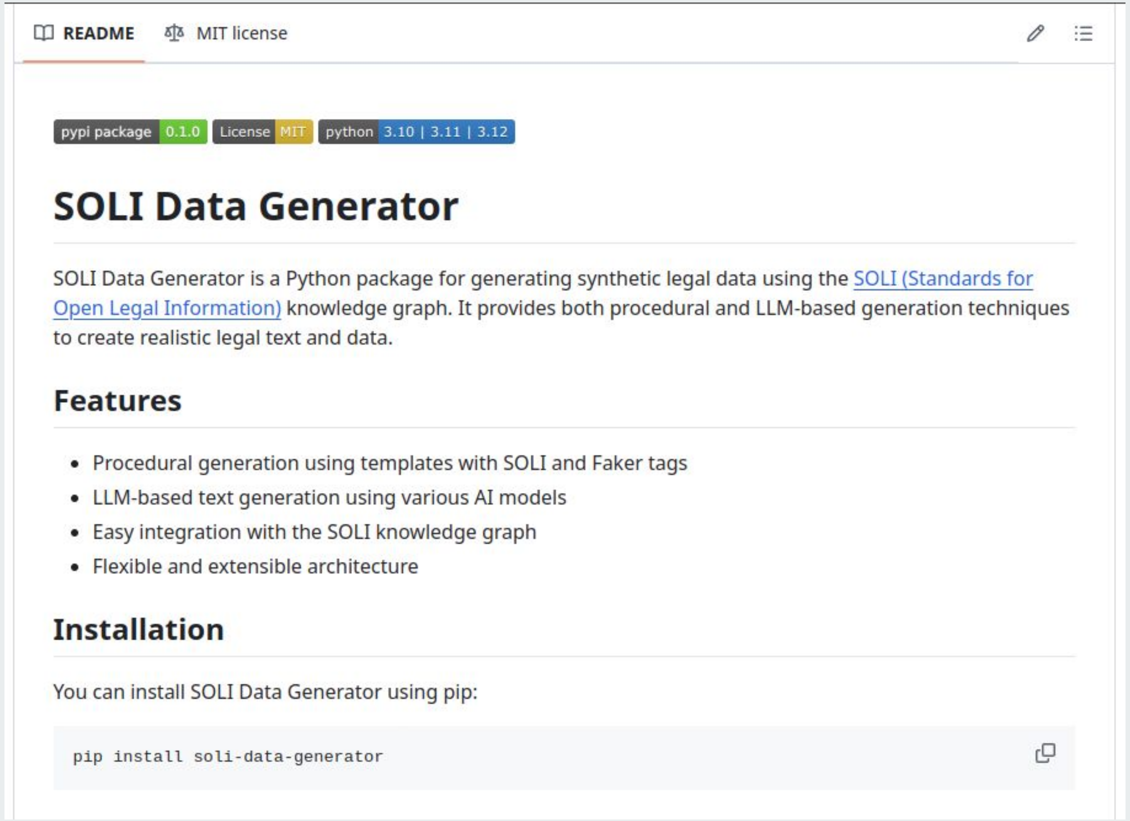
Definition and Examples

Definition
W.D. Mich.

Examples
- N/A

SOLI Data Gen

https://github.com/alea-institute/soli-data-generator



README MIT license

pypi package 0.1.0 License MIT python 3.10 | 3.11 | 3.12

SOLI Data Generator

SOLI Data Generator is a Python package for generating synthetic legal data using the SOLI (Standards for Open Legal Information) knowledge graph. It provides both procedural and LLM-based generation techniques to create realistic legal text and data.

Features

- Procedural generation using templates with SOLI and Faker tags
- LLM-based text generation using various AI models
- Easy integration with the SOLI knowledge graph
- Flexible and extensible architecture

Installation

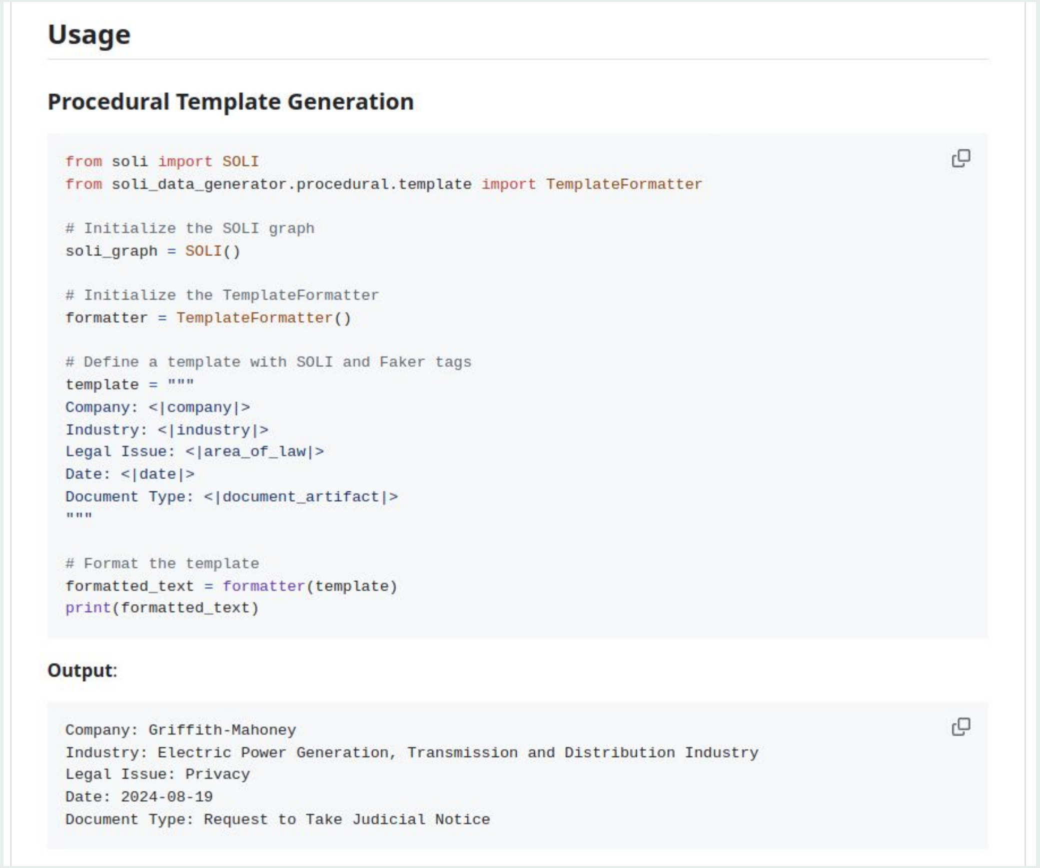
You can install SOLI Data Generator using pip:

```
pip install soli-data-generator
```

SOLI

SOLI Data Gen

https://github.com/alea-institute/soli-data-generator



Usage

Procedural Template Generation

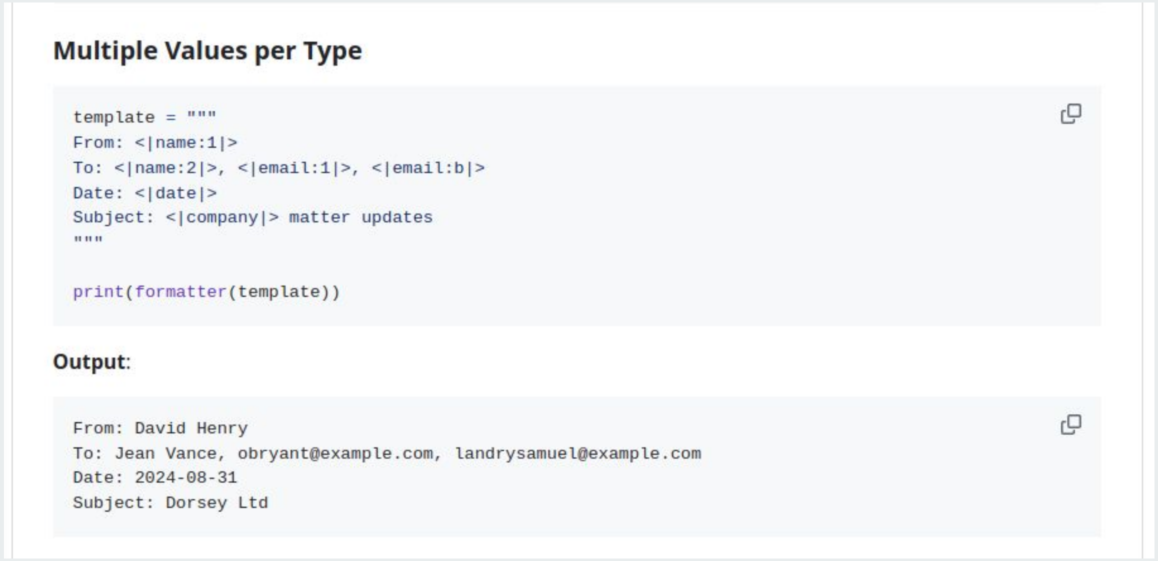
```python
from soli import SOLI
from soli_data_generator.procedural.template import TemplateFormatter

# Initialize the SOLI graph
soli_graph = SOLI()

# Initialize the TemplateFormatter
formatter = TemplateFormatter()

# Define a template with SOLI and Faker tags
template = """
Company: <|company|>
Industry: <|industry|>
Legal Issue: <|area_of_law|>
Date: <|date|>
Document Type: <|document_artifact|>
"""

# Format the template
formatted_text = formatter(template)
print(formatted_text)
```

Output:

```
Company: Griffith-Mahoney
Industry: Electric Power Generation, Transmission and Distribution Industry
Legal Issue: Privacy
Date: 2024-08-19
Document Type: Request to Take Judicial Notice
```

SOLI Data Gen

https://github.com/alea-institute/soli-data-generator



Multiple Values per Type

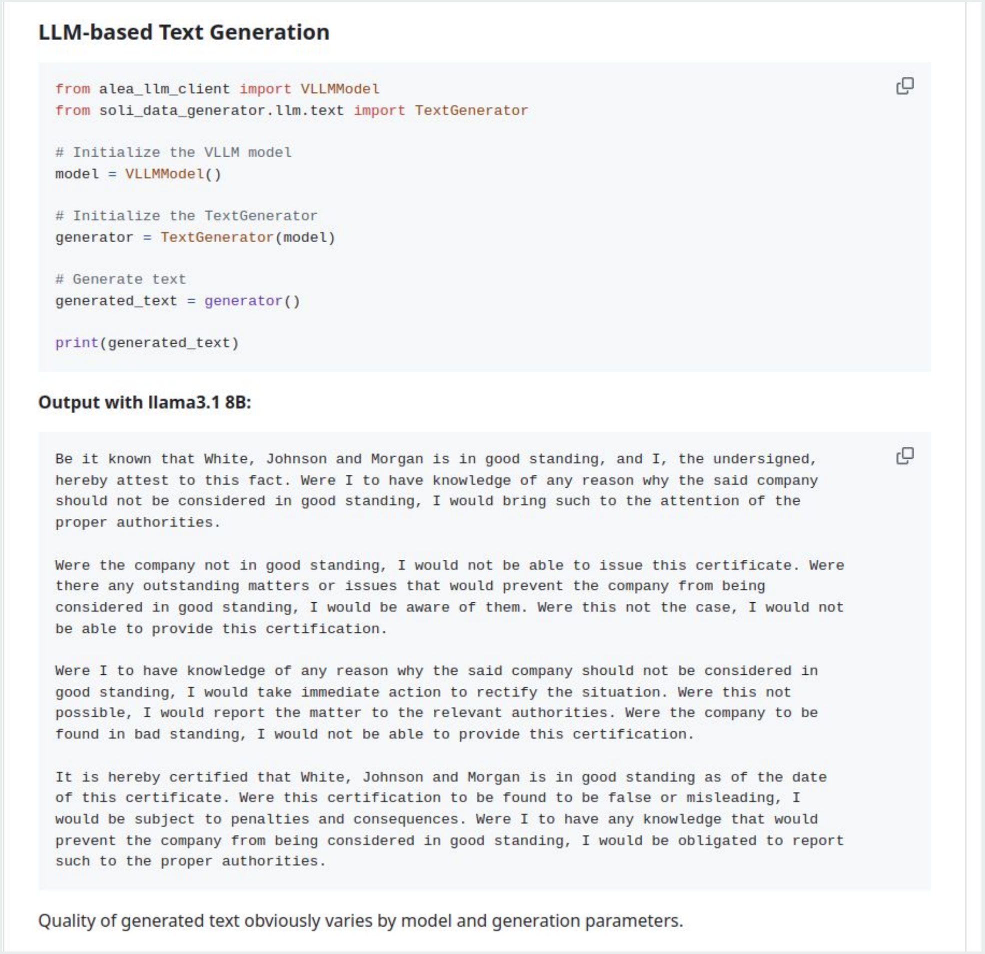
```
template = """
From: <|name:1|>
To: <|name:2|>, <|email:1|>, <|email:b|>
Date: <|date|>
Subject: <|company|> matter updates
"""

print(formatter(template))
```

Output:

```
From: David Henry
To: Jean Vance, obryant@example.com, landrysamuel@example.com
Date: 2024-08-31
Subject: Dorsey Ltd
```

SOLI Data Gen

https://github.com/alea-institute/soli-data-generator



LLM-based Text Generation

```python
from alea_llm_client import VLLMModel
from soli_data_generator.llm.text import TextGenerator

# Initialize the VLLM model
model = VLLMModel()

# Initialize the TextGenerator
generator = TextGenerator(model)

# Generate text
generated_text = generator()

print(generated_text)
```

Output with llama3.1 8B:

```
Be it known that White, Johnson and Morgan is in good standing, and I, the undersigned,
hereby attest to this fact. Were I to have knowledge of any reason why the said company
should not be considered in good standing, I would bring such to the attention of the
proper authorities.

Were the company not in good standing, I would not be able to issue this certificate. Were
there any outstanding matters or issues that would prevent the company from being
considered in good standing, I would be aware of them. Were this not the case, I would not
be able to provide this certification.

Were I to have knowledge of any reason why the said company should not be considered in
good standing, I would take immediate action to rectify the situation. Were this not
possible, I would report the matter to the relevant authorities. Were the company to be
found in bad standing, I would not be able to provide this certification.

It is hereby certified that White, Johnson and Morgan is in good standing as of the date
of this certificate. Were this certification to be found to be false or misleading, I
would be subject to penalties and consequences. Were I to have any knowledge that would
prevent the company from being considered in good standing, I would be obligated to report
such to the proper authorities.
```

Quality of generated text obviously varies by model and generation parameters.

SOLI

Future Roadmap

SOLI Ontology Structure and Future Considerations (15 minutes)

- Overview of the current SOLI ontology structure
- Discuss the balance between:
  - Simple lists
  - Taxonomic aspects
  - Ontological aspects
- Discuss meta-frameworks/representations
- Discuss relation to collection/automation
- Present challenges and considerations for future development
- Invite community input on balancing these aspects in the roadmap
- Collaborate to develop POCs for additional/alternative approaches

SOLI

Open Q&A