Knowledge Representation Workshop 9

CITS3005

October 5, 2023

Ontologies on the web

The following code allows you to launch an ontology as a Flask web application. Follow the given instructions and make sure you understand each step. Then try apply to your own ontology.

1. set up a virtual environment: `python -m venv venv`.
2. activate the environment: `source venv/bin/activiate`
3. install flask and owlready2: `pip install owlready2 flask`
4. save the code on the following page in `bacteria-flask.py`
5. Run with `python bacteria-flask.py`
6. Open `http://localhost:5000/` in your browser.

Taken from Ontologies with Python, Lamy Jean-Baptiste

```
# File dynamic_website.py
from owlready2 import *
onto = get_ontology("owl/bacteria.owl").load()
#onto = get_ontology("http://lesfleursdunormal.fr/static/_downloads/bacteria.owl").load()
from flask import Flask, url_for
app = Flask(__name__)

@app.route('/')
def ontology_page():
    html  = """<html><body>"""
    html += """<h2>'%s' ontology</h2>""" % onto.base_iri
    html += """<h3>Root classes</h3>"""
    for Class in Thing.subclasses():
      html += """<p><a href="%s">%s</a></p>""" % (url_for("class_page", iri = Class.iri), Class.name)
    html += """</body></html>"""
    return html

@app.route('/class/<path:iri>')
def class_page(iri):
    Class = IRIS[iri]
    html = """<html><body><h2>'%s' class</h2>""" % Class.name
    html += """<h3>superclasses</h3>"""
    for SuperClass in Class.is_a:
      if isinstance(SuperClass, ThingClass):
        html += """<p><a href="%s">%s</a></p>""" %(url_for("class_page", iri = SuperClass.iri), SuperClass.name)
      else:
        html += """<p>%s</p>""" % SuperClass
    html += """<h3>equivalent classes</h3>"""
    for EquivClass in Class.equivalent_to:
        html += """<p>%s</p>""" % EquivClass
    html += """<h3>Subclasses</h3>"""
    for SubClass in Class.subclasses():
      html += """<p><a href="%s">%s</a></p>""" % (url_for("class_page", iri = SubClass.iri), SubClass.name)
    html += """<h3>Individuals</h3>"""
    for individual in Class.instances():
      html += """<p><a href="%s">%s</a></p>""" % (url_for("individual_page", iri = individual.iri), individual.name)
    html += """</body></html>"""
    return html

@app.route('/individual/<path:iri>')
def individual_page(iri):
    individual = IRIS[iri]
    html = """<html><body><h2>'%s' individual</h2>""" % individual.name
    html += """<h3>Classes</h3>"""
    for Class in individual.is_a:
      html += """<p><a href="%s">%s</a></p>""" % (url_for("class_page", iri = Class.iri), Class.name)
    html += """<h3>Relations</h3>"""
    if isinstance(individual, onto.Bacterium):
      html += """<p>shape = %s</p>""" % individual.has_shape
      html += """<p>grouping = %s</p>""" % individual.has_grouping
      if individual.gram_positive:
        html += """<p>Gram +</p>"""
      else:
        html += """<p>Gram -</p>"""
    html += """</body></html>"""
    return html

import werkzeug.serving
werkzeug.serving.run_simple("localhost", 5000, app)
```