Transformations en document RDF pour les 10 premières publications du sujet "COMP"

```
import pandas as pd
from rdflib import Graph, Literal, RDF, URIRef, Namespace
from rdflib.namespace import DC, FOAF, DCTERMS, XSD
import urllib.parse

BIBO = Namespace('http://purl.org/ontology/bibo/')
EX = Namespace('http://example.org/')

def clean_uri(uri):
    return urllib.parse.quote(uri)

def create_rdf_graph(publications_df, file_name):
    g = Graph()
    g.bind('dcterms', DCTERMS)
    g.bind('foaf', FOAF)
    g.bind('bibo', BIBO)
    g.bind('ex', EX)

    for idx, row in publications_df.iterrows():
        publication_uri =
URIRef(f"http://example.org/publication/{idx}")

        g.add((publication_uri, RDF.type, BIBO.Document))
        g.add((publication_uri, DCTERMS.subject,
Literal(row['topic'])))
        g.add((publication_uri, DCTERMS.title, Literal(row['title'])))
        g.add((publication_uri, DCTERMS.isPartOf,
Literal(row['publicationName'])))
        g.add((publication_uri, BIBO.doi, Literal(row['doi'])))

        if pd.notna(row['volume']):
            g.add((publication_uri, BIBO.volume,
Literal(row['volume'])))
        if pd.notna(row['issue']):
            g.add((publication_uri, BIBO.issue,
Literal(row['issue'])))
        if pd.notna(row['pageRange']):
            g.add((publication_uri, BIBO.pages,
Literal(row['pageRange'])))
        g.add((publication_uri, DCTERMS.date,
Literal(row['coverDate'])))
        g.add((publication_uri, BIBO.citedBy,
Literal(row['citedbyCount'], datatype=XSD.integer)))

        if pd.notna(row['authors']):
            authors = row['authors'].split(', ')
            for author in authors:
```

```
                author_uri =
URIRef(f"http://example.org/author/{clean_uri(author.replace(' ',
'_'))}")
                g.add((author_uri, RDF.type, FOAF.Person))
                g.add((author_uri, FOAF.name, Literal(author)))
                g.add((publication_uri, DCTERMS.creator, author_uri))

                if pd.notna(row['affiliation']):
                    affiliations = eval(row['affiliation'])
                    for affiliation in affiliations:
                        affilname = affiliation['affilname']
                        affiliation_uri =
URIRef(f"http://example.org/affiliation/{clean_uri(affilname.replace('
', '_'))}")
                        g.add((affiliation_uri, RDF.type,
FOAF.Organization))
                        g.add((affiliation_uri, FOAF.name,
Literal(affilname)))
                        g.add((author_uri, FOAF.member,
affiliation_uri))
    rdf_output_path = f'{file_name}.rdf'
    g.serialize(destination=rdf_output_path, format='xml')

    print(f"RDF data has been written to {rdf_output_path}")

file_path = 'publications_preprocessed.csv'
publications_df = pd.read_csv(file_path)

topic_to_filter = 'COMP'

filtered_df = publications_df[publications_df['topic'] ==
topic_to_filter].head(1)

create_rdf_graph(filtered_df, f'publications_{topic_to_filter}')

rdf_output_path = f'publications_COMP.rdf'
rdf_output_path
```

```
RDF data has been written to publications_COMP.rdf
```

```
'publications_COMP.rdf'
```

Transformations en document RDF

```
import pandas as pd
from rdflib import Graph, Literal, RDF, URIRef, Namespace
from rdflib.namespace import DC, FOAF, DCTERMS, XSD
import urllib.parse

BIBO = Namespace('http://purl.org/ontology/bibo/')
```

```
EX = Namespace('http://example.org/')

def clean_uri(uri):
    return urllib.parse.quote(uri)

g = Graph()
g.bind('dcterms', DCTERMS)
g.bind('foaf', FOAF)
g.bind('bibo', BIBO)
g.bind('ex', EX)

file_path = 'publications_preprocessed.csv'
publications_df = pd.read_csv(file_path)

for idx, row in publications_df.iterrows():
    publication_uri = URIRef(f"http://example.org/publication/{idx}")

    g.add((publication_uri, RDF.type, BIBO.Document))
    g.add((publication_uri,DCTEMS.subject, Literal(row['topic'])))
    g.add((publication_uri, DCTERMS.title, Literal(row['title'])))
    g.add((publication_uri, DCTERMS.isPartOf,
Literal(row['publicationName'])))
    g.add((publication_uri, BIBO.doi, Literal(row['doi'])))

    if pd.notna(row['volume']):
        g.add((publication_uri, BIBO.volume, Literal(row['volume'])))
    if pd.notna(row['issue']):
        g.add((publication_uri, BIBO.issue, Literal(row['issue'])))
    if pd.notna(row['pageRange']):
        g.add((publication_uri, BIBO.pages,
Literal(row['pageRange'])))
    g.add((publication_uri, DCTERMS.date, Literal(row['coverDate'])))
    g.add((publication_uri, BIBO.citedBy, Literal(row['citedbyCount'],
datatype=XSD.integer)))

    if pd.notna(row['authors']):
        authors = row['authors'].split(', ')
        for author in authors:
            author_uri =
URIRef(f"http://example.org/author/{clean_uri(author.replace(' ',
'_'))}")
            g.add((author_uri, RDF.type, FOAF.Person))
            g.add((author_uri, FOAF.name, Literal(author)))
            g.add((publication_uri, DCTERMS.creator, author_uri))

            if pd.notna(row['affiliation']):
                affiliations = eval(row['affiliation'])
                for affiliation in affiliations:
                    affilname = affiliation['affilname']
                    affiliation_uri =
```

```
URIRef(f"http://example.org/affiliation/{clean_uri(affilname.replace('
', '_'))}")
                    g.add((affiliation_uri, RDF.type,
FOAF.Organization))
                    g.add((affiliation_uri, FOAF.name,
Literal(affilname)))
                    g.add((author_uri, FOAF.member, affiliation_uri))

rdf_output_path = 'publicationsenRDF.rdf'
g.serialize(destination=rdf_output_path, format='xml')

print(f"RDF data has been written to {rdf_output_path}")
```

```
RDF data has been written to publicationsenRDF.rdf
```