UWCSECourseHUB

Xinya Tang

IMT 542 24SP

Information School
UNIVERSITY of WASHINGTON

Info Story

- An **API** that aggregates lecture resources from open-source CSE courses at the UW.
- The data is sourced from the CSE courses website calendar page, through Python web scraping tools.
- Extract lecture topics, resource links, and store them by quarter and course locally and to host them on an API

So that— —

User can query with specific keywords to get the related lecture resources in an one-stop way, without hopping around different websites.

Info Story

The query (API endpoint) has **three dimensions**:

- **Course Name**: Users can query for a specific course page by inputting the course name. Each course page displays the course name and course general info.
- **Course Quarter**: Users can query for a specific course quarter page. Each course quarter page presents all of the lecture resources of a specific course in a specific quarter.
- **Lecture Topic**: In each course quarter page, users can query specific topic keywords to get the related lecture resources.

Existing structure and FAIR assess

https://www.cs.washington.edu/education/courses/

Existing structure: course calendar HTML file

- Findability: relatively difficult to find a lecture and associated information
- Accessibility: easily get lost, time consuming
- Interoperability: decent use of metadata
- Reusability: resources data subject to change

How I decided to improve the structure

New information structure: each API endpoint returns a JSON file, which contains an array of lecture objects. Each lecture object has two properties:

- "lecture description": this property contains a string describing the lecture.

- "links": this property contains an array of URLs representing the links to various resources related to the lecture.

Here's the information schema of the JSON file:

```
{
"lecture description": "String",
"links": ["String", "String", ...]
}
```

New structure (demo)

http://127.0.0.1:5000

- Home Page
- Course Page
- Course Quarter Complete Resource
- Topic Specific Query

Further quality improvement

- Data Integrity:

  The accuracy and integrity of scraped data may be compromised by inconsistencies or errors on source websites structure. Implementing more robust and dynamic web scraping techniques and more advanced data validation and verification mechanisms can help ensure the integrity of information presented to users.

- Data Timeliness:

  The timeliness of data presented may be compromised by content changes on source websites. Update the backend JSON file on a monthly basis (ideally bi-weekly basis) can help improve the timeliness of the API endpoint query result by users.

Thank you :)

Any questions?