Bern University
of Applied Sciences

URL-Archiver

Project report

Course of study	Project 1
Author	Nicolin Dora, Abidin Vejseli & Kilian Wampfler
Advisor	Dr. Simon Kramer
Co-advisor	Frank Helbling

Version 0.1 of October 2, 2023

- Technik und Informatik
- Informatik

Abstract

One-paragraph summary of the entire study – typically no more than 250 words in length (and in many cases it is well shorter than that), the Abstract provides an overview of the study.

Contents

List of Tables

List of Figures

Listings

1. Introduction

1.1. Initial Situation

In the age of rapidly evolving digital content, there's an increasing need to archive and preserve web resources. URLs, which provide direct access to these resources, are often embedded within various file types, including but not limited to Unicode-text files like .BIB, .TEX, and .HTML, as well as .PDF files. However, with the internet's transient nature, these URLs can become obsolete or the content they point to can change, leading to the loss of valuable information. It is, therefore, crucial to have a mechanism in place to archive these URLs, ensuring their content remains preserved and accessible over time.

1.2. Project Goal

The primary aim of the "URL-Archiver" project is to develop a Free/Libre and Open Source Software (FLOSS) licensed, platform-independent Java program that can effectively archive URLs. The envisioned software should:

1. Accept as input a directory or any Unicode-text file (like .BIB, .TEX, .HTML), or a .PDF file.
2. Scan the provided input for any embedded URLs.
3. Extract all identified URLs.
4. Offer an option to spring-load these URLs in a Web-browser, providing an immediate view of the referenced content.
5. Submit these URLs to the online archiving service, https://archive.ph, ensuring their content is stored and safeguarded against potential future changes or deletions.
6. Retrieve the archived URLs from the service.
7. Output a CSV-file detailing the original URLs and their corresponding archived versions.
8. Optionally allow the integration of the archived URLs into a .BIB file, facilitating the ease of reference in academic or professional contexts.

Furthermore, in the spirit of best programming practices, the program's code should be minimalistic, modular, and self-explanatory. This not only ensures maintainability but also aids any future development or modifications.

1.3. Priorities

1. **Functionality**: The primary priority is the accurate extraction and archiving of URLs. The software should reliably identify URLs in varied file types and ensure their successful archiving on https://archive.ph.
2. **Usability**: Given the diverse potential user base, the program should be platform-independent and possess a user-friendly interface. While the underlying mechanisms may be complex, the user experience should be seamless and intuitive.
3. **Code Quality**: Emphasis should be placed on writing clean, minimal, and modular code. This not only aids in potential future enhancements but also in debugging and troubleshooting.
4. **Documentation**: As with any software project, proper documentation is paramount. The project report should be concise, adhering to the principle of being "maximally informative, minimally long," ensuring clarity of information without overwhelming the reader.
5. **Integration with Existing File Types**: The ability to seamlessly insert archived URLs into .BIB files is a priority, given the potential academic applications of the software.

2. Specification

2.1. System Delimitation

2.1.1. System Environment (statics)

2.1.2. Process Environment (dynamics)

2.2. Requirements

2.2.1. Functional requirements (added value)

2.2.2. Boundary and Pre-Conditions

2.3. Usability

2.3.1. Personas

2.3.2. Storyboard

2.3.3. UX-Prototyping

3. Implementation

3.1. Architecture (e.g., back-/frontend)

3.2. Processes

4. Deployment/Integration

4.1. Installation (Sysadmin) Manual & Script

4.2. User Manual

5. Conclusion

5.1. Discussion

5.1.1. Example from BFH Template - Delete

What is the significance of your results? – the final major section of text in the paper. The Discussion commonly features a summary of the results that were obtained in the study, describes how those results address the topic under investigation and/or the issues that the research was designed to address, and may expand upon the implications of those findings. Limitations and directions for future research are also commonly addressed.

5.2. Bottom Line

5.3. Future Work

Bibliography

A. Original Project Description

```
# URL-Archiver

## Description

The goal of this project is to deliver a FLOSS-licensed,
platform-independent Java-program (called "URL-Archiver") that
(1) takes as input (the path of) a directory or any Unicode-text-
    (e.g.: .BIB, .TEX; .HTML; etc.) or .PDF-file
     (https://www.baeldung.com/java-curl);
(2) scans it for any URLs
 (https://stackoverflow.com/questions/4026614/extract-text-from-pdf-files ,
    https://librepdf.github.io/OpenPDF , https://pdfbox.apache.org ; see
     also
    https://en.wikipedia.org/wiki/List_of_PDF_software);
(3) extracts all URLs (regular expression ;-) from the text;
(4) optionally spring-loads all URLs in a Web-browser;
(5) posts all URLs to https://archive.ph ;
(6) gets the resulting archived URLs;
(7) outputs a CSV-file of the resulting key-value (URL, archived URL)
 pairs; and
(8) optionally inserts the archived URLs into a .BIB-file.

The program code should be minimal, modular, and self-explaining.

The project report should be concise (maximally informative, minimally
 long).
It must contain this project description as a quotation.

## Technologies

Java, LaTeX

## Advisor

Dr. Simon Kramer
```

Declaration of Authorship

I hereby declare that I have written this thesis independently and have not used any sources or aids other than those acknowledged.

All statements taken from other writings, either literally or in essence, have been marked as such.

I hereby agree that the present work may be reviewed in electronic form using appropriate software.

October 2, 2023

N. Dora

A. Vejseli

K. Wampfler