# Website Downloader

This project contains two modules, a `fetcher` to download the entire content of a specified website and a `server` to provide search results based on the keywords extracted from the downloaded website.

## Requirements

This project uses a `cmake` based build system. Hence to build it in the first place `cmake >=3.6` is required. Additionally it also depends on several external libraries which are necessary for the project to work. The required libraries are as follows:

- `openssl` - To fetch websites from `https` hosts
- `mysql` - To store fetched and processed links
- `threads` - For multi-threading during website download

### Debian specific packages

```
sudo apt-get update
sudo apt-get install cmake libssl-dev pkgconf libmysqlclient-dev mysql-client
mysql-server
```

### MAC OSX

```
brew update
brew install cmake openssl mysql-server mysql-client mysql-dev
```

## Building

To build this project execute:

```
cmake CMakelists.txt
make
```

## Run

### Setup MySQL database

```
mysql -u <username> -p <password>
mysql> create database np_select;
```

### Inspect database (optional)

```
mysql> use np_select;
mysql> SELECT * FROM Links WHERE id > 0 ORDER BY id ASC LIMIT 1000;
```

### Configuration

The configuration can be provided in `config.ini`. Different config files can be used at the same time as long as they are passed on to the program as arguments. If nothing specified, program will take `config.ini` as the default configuration.

#### storage

- `save_location`: location to download the website
- `invalid_save_location`: location to download invalid URLs

#### server

- `host`: hostname of website to be downloaded
- `protocol`: http or https
- `start_page`: entry point
- `begin_at`: beginning id of database index (can be used to resume download)
- `cert`: location to certificate chain
- `timeout`: timeout in seconds

#### database

- `host`:
- `username`:
- `password`:
- `name`: name of the database

#### search_engine

- `root_path`: location to the template files for the search engine

### Start executables

Both the executables can be called as follows:

```
./fetcher [config_file]
./server [config_file]
```

## About

This repository was developed by Kunal Pal and Thorsten Born as part of their Network Programming lab project.