Assessed Coursework 1

Developing a Simple Web Browser

F21SC: Industrial Programming

Louis Prud’homme

Table of contents

Introduction

ntm

Requirements’ checklist

- ✓ Sending HTTP request messages
- ✓ Receiving HTTP response messages
  - ✓ Display received HTML
  - ✓ Display received HTTP response status code (code and status)
- ✓ Reload current page
- ✓ Home page
  - ✓ Create home page URL
  - ✓ Edit home page URL

    - [x] Home page URL persisted
- [x] Favourites
    - [x] Add a favorite (composed of a name and an URL)
    - [x] Delete favourites
    - [x] Modify favourites
    - [x] Request a favorite by click
    - [x] Favourites persisted
- [x] History
    - [x] Next and previous pages navigation
    - [x] Jump to page by click
    - [x] History persisted
- [x] GUI
    - [x] Buttons
    - [x] Shortcuts

To the best of my knowledge, no requirement was overlooked.

Design considerations

Architecture design

I opted for a structure based on the MVP pattern, as it seemed quited practical in the context of a WinForm application. I also looked into the MVC and MVVM patterns, but the former seemed rather unadapted to WinForms while the latter seem too complex for a project with only four model sources.

Data structures

Indeed, I indentified the three model sources as the homepage URL, the favourites, the history and the navigation. I made the choice to encapsulate them in a `UserProfile` class, to simplify the backup.

Each model source is structured differently ; for exemple, the home page URL is stored as a mere `URI` object in the `UserProfile` instance.

- Favourites are individually represented by the `Fav` class and stored in the `HashSet<Fav>`-encapsulating `FavoritesRepository` class. The `HashSet>` was chosen to guarantee each favourite would be represented once.
- History entries are represented as `HttpQuery` objects and stored in the `SortedDictionary<long, HttpQuery>`-encapsulating `GlobalHistory` class, where the `long` key represents a millisecond-precise timestamp of when the query was initiated. The `SortedDictionnary` was chosen to preserve the order of entries.

- The user’s navigation is stored in the `LocalNavigation` class, which encapsulates a `LinkedList`. It is optimized for navigating through its nodes.

Advanced language constructs

I made use of the asynchronous possibilities offered by the `HttpClient` class I use to execute the user’s HTTP queries, which allows the interface to be still responsive during requests.

Furthermore, I also made heavy use of Linq, thanks to which I seldom accessed `IEnumerable<T>` structures through loops, especially in the model classes.

One of the primary features enabling me to organize this project as is are delegates. Each `IPresenter` and `IView` classes comprises at least two of them, as events are used for upwards communication (from view to presenter to application context).

User guide

Developer guide

Testing

Reflections on programming language and implementation

Conclusion