Offline Experience for Web Application

Summary

QDAcity is a web application for conducting Qualitative Data Analysis (QDA) of text data. It is used by researchers to analyze qualitative data such as transcribed interviews with the purpose of theory building research. A major advantage of QDAcity is, that as a web application it makes collaboration on a research project more efficient. However the biggest drawback compared of traditional tools for this task is that work is only possible if a connection to our server exists.

With growing browser support for service workers it becomes feasible to create an offline experience with standard web technologies, with the service worker acting as a proxy to the network and backend. You can manage a cache of responses and when the network is available the app should be able to sync in the background. Within this project you will explore how this can be applied to QDAcity.

At the end of this thesis the main task of the application, the coding of documents, shall be possible with intermittent connection loss.

Work Results

- Preparation
    - App Engine development
    - React
    - Service worker documentation
- Implementation
    - Built on the QDAcity technology stack
        - Java 8 + JDO + Google Datastore backend
        - JS + React.js frontend.
    - At loss of connection the application should inform the user about the offline state
    - After a loss of connection, a default response should signal a notification to users that the operation is currently not supported offline, for all non-supported operations.
    - Supported operations for offline use which need to be cached within the service worker for later synchronitation on re-connect include the following (in order of priority).
        - Applying of codes to text and removing of coding instances from text
        - Changing of code attributes (name, color, memo, etc)

- - ■ Adding, removing and relocating codes within the code system

    - ■ Adding and removing documents

  - ○ Furhter, improvements of loading performance should be evaluated through caching of content in the service-worker.

- ● Description of solution in written thesis

Supervisor

Andreas Kaufmann, M.Sc., andreas.kaufmann@fau.de
Prof. Dr. Dirk Riehle, dirk.riehle@fau.de

Open Source Research Group
Computer Science Department
Friedrich-Alexander University

More information: http://osr.cs.fau.de/theses/resources/