

Conflict Resolution in Collaborative Web Apps with Offline Use

Summary

With the advent of progressive web apps (PWA) the line between web apps and native apps is becoming blurred. In particular with the use of service worker technology already supported by all popular browsers, web apps can continue to function without a network connection. The service worker (SW) then acts as a proxy to the backend and responds in its stead when the network is not available, while keeping track of the changes made locally. Then, when the connection is re-established, the changes are then synced with the server. Within this project you will extend the offline capabilities of an example web application QDAcity.

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. Currently we already support a subset of user actions in offline mode within QDAcity.

Within this project you will extend this subset of features, and in particular improve the user experience for conflict resolution in the case where the same object has been edited offline, and concurrently online by a different user. There needs to exist a merge algorithm that can identify conflict-free merges and a user interface for displaying the diff for two versions that need to be resolved manually.

Work Results

- Preparation
 - App Engine development
 - React-, service worker-, workbox- documentation
 - Read previous thesis on offline experience in QDAcity
- Implementation
 - Built on the QDAcity technology stack
 - Java 8 + JDO + Google Datastore backend
 - JS + React.js + Workbox frontend.
 - Supported operations for offline use which need to be cached within the service worker for

later synchronisation on re-connect include the following (in order of priority).

- Editing text documents
- Working with the UML editor
- Differential synchronisation for offline edits
- UI for showing a user friendly and editable diff for conflicts during synchronisation
- 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/>