

# Wie können neue/zusätzliche Features eingebaut werden?

## 1. Code Erklärung Allgemein

Zu Beginn stellte sich bereits heraus, dass es gar nicht so einfach ist Daten von einem Endgerät auf eine Webapplikation zu laden. Den Browsern ist es aus Sicherheitsgründen nicht erlaubt direkt auf die Dateien der Festplatte eines Endgerätes zuzugreifen. Um das Hochladen von Artikeln auf unsere Plattform zu ermöglichen mussten wir einen lokalen Server zwischenschalten. Dieser agiert wie eine API, welche die Daten auf dem entsprechenden Port hostet. Alle Code Dateien innerhalb des public Ordners werden gehostet, alles darum ist Teil des Servers.

Die Daten werden anschließend auf der Seite “/xml” bereitgestellt. Zuvor wurde das XML zur leichten weiterverarbeitung zu JSON konvertiert. Die Seite “/xml” ist für den User nicht aufrufbar und enthält keinerlei grafische Elemente, ausschließlich die XML-Dateien im JSON-Format.

loadData() lädt die JS-Objekte Dateien, und speichert diese anschließend im Array “data“. Von dort werden mithilfe von filteredData() über die Radiobuttons und das Suchfeld die entsprechenden Artikel Dateien geladen und mit showData() dann angezeigt. Im Anschluss greift generateElement() und erzeugt aus den geladenen Daten eine Tabelle diese, hat zwei wechselnde Hintergrundfarben zur besseren Übersichtlichkeit.

Mit Hilfe von “.innerHTML,” in den load Funktionen (loadHome, loadProject & loadAnnotation), werden die Artikel dann auf der Seite integriert und angezeigt, sodass eine modulare Struktur entsteht. Diese laden die entsprechenden HTML Seiten mit einer kurzen Verzögerung (timedelay) um sicherzustellen, dass alle Daten geladen sind bevor die Seite gebaut wird. All dies findet im content Bereich, der Seite, die sich zwischen dem Header und dem Footer befindet, statt.

Auf der Annotationsseite kann durch die zugehörige Javascript Datei der ausgewählte Artikel per Klick farblich markiert werden. Dafür lädt und zerteilt splitText() die Artikel Objekte aus dem Localstorage in einzelne Sätze. Die Funktion checkCurrentColor() ist im Allgemeinen sehr einfach. Es wird lediglich bei einem Klick auf einen Satz eine Klasse für die ausgewählte Farbe hinzugefügt. Bei einem erneuten Klick wird diese

Klasse über die Methode toggleClassList wieder entfernt, wenn auf ein bereits markiertes Feld mit einer neuen Farbe geklickt wird, wird diese Farbe mit der neuen ausgewählten Farbe überschrieben und erhält auch diese Klasse.

## 2. Erstellung von neuen Features

