

Meilenstein 5 – NPM, Sass & WebSockets

Ziel ist eine Verbesserung der Struktur, Erweiterbarkeit und Wartbarkeit in zwei Bereichen der Anwendung durch: 1) im Backend über den Einsatz von NPM für das Management von Abhängigkeiten in JavaScript und 2) Sass als Framework für die Vorverarbeitung für CSS.

Zudem verfolgen wir das Ziel der weiteren Verbesserung der Usability von Abalo durch die Anwendung von interaktiven Nachrichten. Die Nachrichten sollen automatisch beim Eintreten vom Webserver an den Webbrowser ohne Zeitverzögerung übertragen werden.

Aufgabe 1

Refactoring Artikeleingabe. Entwickeln Sie die Artikeleingabe unter Einsatz von Vue erneut und tragen Sie die Artikel über den Webservice unter POST /api/articles ein. Welche Features von Vue setzen Sie dabei ein?

Aufgabe 2

Installation NPM. Installieren Sie NPM auf Ihrem Entwicklungsrechner:

<https://nodejs.org/en/download/>

Aufgabe 3

NPM. Binden Sie Vue über NPM in Abalo ein.

Aufgabe 4

Styling. Stylen Sie (mindestens) die folgenden Komponenten in Abalo:

1. Menü,
 2. Artikelsuche und -liste und
 3. Artikeleingabe.
- a) Verwenden Sie mindestens für 5 Namen von Klassen die BEM-Methodologie, wobei Sie alle Bestandteile (Block, Element, Modifier) ausnutzen.
 - b) Verwenden Sie beim Styling mindestens 5 verschiedene Features von Sass. Welche Features nutzen Sie?

Aufgabe 5

Einbindung Komponente. Suchen Sie sich eine Komponente von Vue unter <https://vuecomponents.com> aus. Bereiten Sie eine Präsentation als Video für die anderen Kursteilnehmer:innen vor, in dem Sie die Komponente vorstellen.

Es gelten die folgenden Rahmenbedingungen:

- Mindestlänge: 90 Sekunden.
- Zeigen Sie die Komponente anhand eines Beispiels in Ihrem Abalo Projekt.
- Wie wird die Komponente in Vue eingebunden?
- Welche Konfigurationsparameter stehen zur Verfügung?
- Für welchen Anwendungsfall kann man die Komponente einsetzen? Wofür kann man diese nicht einsetzen?

Laden Sie das Video in ILIAS hoch unter > Praktikum > Komponenten, wobei als Dateiname verwenden:

<TeamNr>_<KomponentenName>.mp4

Die TeamNr finden Sie in der Teamzuordnung.

Hinweis: Sie sind völlig frei bei der Gestaltung des Videos

Woche 2

Aufgabe 6

WebSockets: Installation Broadcaster. Installieren Sie den Websocket-Server Ratchet cboden/ratchet sowie ratchet/pawl (als Client) über composer.

<https://packagist.org/packages/cboden/ratchet>

Aufgabe 7 (optional)

Ziel der Aufgabe ist ein erstes Kennenlernen und Einüben der Verwendung von WebSockets. Verwenden Sie den installierten Broadcaster für die folgenden Aufgaben.

Implementieren Sie eine Anwendung ...

- **5-ws1-connect.html**, die eine WebSocketverbindung aufbaut und nach erfolgreicher Verbindung eine Nachricht „Connected“ auf der Konsole ausgibt.
- **5-ws2-message.html**, die eine Nachricht auf der Konsole "Message received: <msg>" darstellt, die vom Broadcaster verteilt wird. Senden Sie dafür über den im Projekt bloatless/php-websocket enthaltenen Client (unter

cli/client.php) die Nachricht "<msg>" an den Broadcaster. Tauschen Sie <msg> durch beliebige Texte aus.

- **5-ws3-selected-message.html**, die eine Nachricht an den Webbrowser sendet. Die Nachricht soll neben dem Text eine beliebige Kennungsid übertragen. Der Webbrowser soll den Text nur dann darstellen, wenn die übertragene Kennungsid mit einer konstanten Kennungsid im clientseitigen JavaScript Code übereinstimmt.
- **5-ws4-vue.html**, die eine minimale Vue-Anwendung enthält und eine empfangene Nachricht über Vue ausgibt.

Aufgabe 8

Wartungsnachricht. Der Prototyp von Abalo erfährt aktuell noch sehr häufig Aktualisierungen. Es soll eine Meldung zu einem anstehenden Wartungsmodus verzögerungsfrei an alle Nutzer:innen der Webseite geschickt werden. Die Meldung lautet:

In Kürze verbessern wir Abalo für Sie!
Nach einer kurzen Pause sind wir wieder
für Sie da! Versprochen.

Implementieren Sie dazu einen WebSocket zwischen Webbrowser und Webserver. Schicken Sie eine Nachricht an den Broadcaster, der alle Nutzer:innen der Webseite sofort informiert.

Aufgabe 9

Verkaufsmeldung. Angemeldete Nutzer:innen sollen in Abalo sofort eine Benachrichtigung erhalten, sobald einer ihrer angebotenen Artikel verkauft ist. Zu dem Zweck soll eine Nachricht

„Großartig! Ihr Artikel <Artikelname> wurde erfolgreich verkauft!“

sofort ausschließlich beim dem/der betreffenden Nutzer:in (der/die den Artikel angeboten hat) mit dem korrekten Namen des Artikels dargestellt werden. Entwerfen Sie einen neuen Endpunkt als Webservice /api/articles/{id}/sold über ein POST, der die Nachricht über einen WebSocket verteilt.

Hinweis: Der Artikel muss nicht als „verkauft“ markiert werden. Ziel ist die Implementierung der Verteilung der Nachricht.

Aufgabe 10

Angebot. Angemeldete Nutzer:innen sollen ihre Artikel als Angebot bewerben können. Dafür gibt es eine Schaltfläche „Artikel jetzt als Angebot anbieten“, die jede/n Besitzer:in zu den eigenen Artikeln sieht. Bei Betätigung dieser Schaltfläche

erhalten alle anderen aktuellen Betrachter:innen des Artikels eine Meldung:

„Der Artikel <Artikelname> wird nun günstiger angeboten! Greifen Sie schnell zu.“

Ein/e Nutzer:in gilt als „Betrachter:in“, wenn diese/r den betreffenden Artikel in der Ergebnismenge der Artikelliste aktuell sieht. In der Artikelliste soll der betreffende Artikel hervorgehoben dargestellt werden (z.B. in fatter Schrift).

- a) Stellen Sie bei Betätigen der Schaltfläche die genannte Meldung dar.
- b) Verwenden Sie Axios für den Versand der Verarbeitungsaufforderung der Betätigungsaktion der Schaltfläche an den Webserver.
- c) (Optional) Heben Sie den betroffenen Artikel in der Artikelliste visuell hervor.

Hinweis: Der Preis des Artikels muss nicht angepasst werden.

Aufgabe 11

Erweiterungen. Nenne Sie drei weitere Erweiterungsmöglichkeiten in Abalo durch WebSockets. Erklären Sie für jede Möglichkeit den Kundennutzen.

Aufgabe 12 (optional)

Full-Featured WebSockets. Entwickeln Sie die Aufgaben „Wartungsnachricht“, „Verkaufsmeldung“ und „Angebot“ unter Verwendung der Bibliotheken Echo und Laravel-WebSockets. Verwenden Sie unterschiedliche Channels für das Broadcasting der Nachrichten.

Hinweis: Die Aufgabe ist komplex und umfasst eine Vielzahl unterschiedlicher weiterer technischer Komponenten. Entsprechend existieren viele potenzielle Fehlerquellen, die eine erfolgreiche Umsetzung dieser Aufgabe erschweren.

Aufgabe

Abgabe. Laden Sie Ihre Ergebnisse als ZIP in ILIAS hoch. Der Name des ZIPs soll sein:

<TeamNr>_<Meilensteinnummer>.zip

Die TeamNr finden Sie in der Teamzuordnung. TeamNr und Meilensteinnummer schreiben Sie bitte ohne führende Null. Also z.B. 4_2.zip für Team 4 Meilenstein 2.

Das ZIP soll alle Ergebnisse des Meilensteins (Antworten zu Freitextaufgaben, Grafiken, Quelltexte, ...) enthalten. Fügen Sie zusätzlich ein README ein, wo Sie Vor-, Nachname und Matrikelnummer der beteiligten Teammitglieder hinterlegen.