Diese Doku soll es etwas einfacher machen, das alertBase.vue Component für Seiten einzusetzen. Das Component soll es ermöglichen, eigene Meldungs- und Bestätigungsfenster für Webseiten zu verwenden, statt in JavaScript die basic Fenster mit alert() und confirm() nutzen zu müssen.

**Hinweis**: In den Texten habe ich einige Variable & Methodennamen farblich markiert. Rot markierte Bezeichner sind Namen, die so wie geschrieben übernommen werden müssen. Grün markierte Bezeichner sind Variablen/Methodennamen die frei wählbar sind.

**Alert-Typen**: Aktuell kann man mit der Komponente zwei Arten von Alerts/Fenster erzeugen — ein Bestätigungsfenster mit einem Bestätigungs- und Abbrechenknopf, und ein einfaches Alertfenster, mit einem Knopf zum Schließen. Im weiteren Verlauf der Doku verwende ich die Bezeichnung „*Meldung*“ wenn ich beide Typen referenziere, und *Alertfenster* bzw. *Bestätigungsfenster* für Bezug auf eines der zwei konkreten Typen

**Meldungstextinhalte**: Die Textinhalte der verschiedenen HTML-Elemente der Meldung werden über Props festgelegt

* alert-heading: Textinhalt der Überschrift für die Meldung
* alert-paragraph: Textinhalt des Fließtexts für die Meldung
* alert-close-button: Textinhalt, der auf den Knopf zum Schließen der Meldung erscheinen soll
* alert-okay-button: Textinhalt, der auf den Bestätigungsknopf des Bestätigungsfensters erscheinen soll. Dieser Prop ist nur notwendig/relevant für Bestätigungsfenster.

**Einbindungsschritte**: Um eine Meldung einzubinden, muss folgende Sachen in der Datei vorhanden sein, in der sie eingebunden werden soll

* **Visibilitätsvariable (alertVisible)**: In der data() Option muss die Variable "alertVisible" definiert und auf *false* gesetzt werden. Diese Variable wird zwischen den Komponenten zugegriffen, um das Anzeigen und Verbergen zu realisieren.
* **Event Handler**: Für jeden Knopf der Meldung muss es jeweils eine entsprechende Methode in der methods() Option geben. Referenzen auf diese Methoden werden in einer Variable in data() (zB alertHandlers) gespeichert, damit diese als Wert fürs v-on Directive angegeben werden kann, um mehrere Handlers zu übergeben. In dieser Variable muss die Methodenreferenz für den Schließknopf unter den Namen "closeAlertClicked" gespeichert werden, da close-alert-clicked ein benutzerdefiniertes Event vom AlertBase Komponent ist, das emitted wird. Falls es sich bei der Implementierung um ein Bestätigungsfenster handelt, muss in alertHandlers auch die Methodenreferenz für den Bestätigungsknopf unter den Namen "confirmationClicked" gespeichert werden, aus ähnlichem Grund.
* **Inhalt der Event Methoden**: Die Methode, die unter "closeAlertClicked" in "alertHandlers" gespeichert wird invertiert lediglich den Wert der this.alertVisible variable. Im Falle eines Bestätigungsfensters, wäre die Methode für "confirmationClicked" in "alertHandlers" einfach die Methode, die aufgerufen werden soll nachdem der User seine Intention bestätigt hat.
* **HTML Tag Anforderungen**: Im *alert-base* tag wird v-if verwendet, um das Fenster anzuzeigen, wenn this.alertVisible true ist. Außerdem wird v-on auf die Data Variable mit den Handler-Methoden-Referenzen gesetzt. Für den "alert-type-class" prop muss der wert "alertTypeBasic" oder "alertTypeConfirmation" übergeben werden, je nachdem, ob das Component für ein einfaches Alert-Fenster oder Bestätigungsfenster eingesetzt wird. Ansonsten müssen noch die Props für die Textinhalte angegeben werden, die oben in **Meldungstextinhalte** aufgeführt wurden.
* Provide Option: Um die "alertVisible" Data-Variable der Seite/Komponent fürs AlertBase Komponent verfügbar zu machen, muss die provide() Option eingefügt werden, der diese Variable an AlertBase mit demselben Namen übergibt

**Auslösung des Fensters**: Um das Fenster anzuzeigen, muss einfach die alertVisible-Umkehrungsfunktion (zB this.updateAlertVisibility()) im Code aufgerufen werden. Dies würde den Wert von alertVisible von *False* auf *True* ändern.

**Beispiel anhand Implementierung eines Bestätigungsfensters in MenuItemsOpened.vue**

**In <template>**

* <alert-base v-if="this.alertVisible" v-on="alertHandlers" alert-heading="Wirklich Abmelden?" alert-paragraph="Wollen Sie sich abmelden?" alert-close-button="Abbrechen" alert-okay-button="Abmelden" alert-type-class="alertTypeConfirmation"/>

**In data()**

* alertVisible: false
* alertHandlers: {confirmationClicked: this.logout, closeAlertClicked: this.updateAlertVisibility}

**Die Provide Option**

* provide() {return {alertVisible: this.alertVisible}}

**In Methods**

* updateAlertVisibility() {this.alertVisible = !this.alertVisible;}
* redirectToView(menuitem) {

if (menuitem.name === 'Abmelden') {

this.updateAlertVisibility();

} else {

this.$router.push(this.getRoute(menuitem.name));

}

}

* logout() {

this.alertVisible = false;

axios.post('/logout', {}, {

headers: {

'X-CSRF-TOKEN': document.querySelector('meta[name="csrf-token"]').getAttribute('content')

}

}).then(() => {

localStorage.removeItem('auth\_token');

this.$router.push('/login');

}).catch(error => {

console.error('Logout fehlgeschlagen', error);

alert('Logout fehlgeschlagen. Bitte versuchen Sie es erneut.');

});

}

**Kompatabilität mit der Composition API**: Bei Implementierung auf einer Seite, die bereits die Composition API verwendet, ist der Prozess etwas aufwendiger, da das AlertBase Komponent mit der Options API entwickelt wurde. Welche Schritte in dem Fall noch dazu kommen würde ich hinzuschreiben, wenn ich wieder da bin.

**PS**: Man könnte wahrscheinlich die alertVisible Variable auf der Seite frei wählen, solange diese Variable dann in Provide unter den Namen alertVisible übergeben wird (return {alertVisible: this.differentNamedVisibilityVariable}). Das habe ich aber noch nicht konkret ausprobiert. Ich wollte es nur schon mal gesagt haben