import { getNode } from './utils';

import { CANCEL\_KEY } from './options/buttons';

import CLASS\_NAMES from './class-list';

const {

OVERLAY,

SHOW\_MODAL,

BUTTON,

BUTTON\_LOADING,

} = CLASS\_NAMES;

import state, { SwalState } from './state';

export const openModal = (): void => {

let overlay = getNode(OVERLAY);

overlay.classList.add(SHOW\_MODAL);

state.isOpen = true;

};

const hideModal = (): void => {

let overlay = getNode(OVERLAY);

overlay.classList.remove(SHOW\_MODAL);

state.isOpen = false;

};

/\*

\* Triggers when the user presses any button, or

\* hits Enter inside the input:

\*/

export const onAction = (namespace: string = CANCEL\_KEY): void => {

const { value, closeModal } = state.actions[namespace];

if (closeModal === false) {

const buttonClass = `${BUTTON}--${namespace}`;

const button = getNode(buttonClass);

button.classList.add(BUTTON\_LOADING);

} else {

hideModal();

}

state.promise.resolve(value);

};

/\*

\* Filter the state object. Remove the stuff

\* that's only for internal use

\*/

export const getState = (): SwalState => {

const publicState = Object.assign({}, state);

delete publicState.promise;

delete publicState.timer;

return publicState;

};

/\*

\* Stop showing loading animation on button

\* (to display error message in input for example)

\*/

export const stopLoading = (): void => {

const buttons: NodeListOf<Element> = document.querySelectorAll(`.${BUTTON}`);

for (let i = 0; i < buttons.length; i++) {

const button: Element = buttons[i];

button.classList.remove(BUTTON\_LOADING);

}

};