import { stringToNode } from '../utils';

import { injectElIntoModal } from './modal';

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

const { BUTTON, DANGER\_BUTTON } = CLASS\_NAMES;

import { ButtonList, ButtonOptions, CONFIRM\_KEY } from '../options/buttons';

import { footerMarkup, buttonMarkup } from '../markup';

import { onAction } from '../actions';

import {

setActionValue,

setActionOptionsFor,

ActionOptions,

} from '../state';

/\*

\* Generate a button, with a container element,

\* the right class names, the text, and an event listener.

\* IMPORTANT: This will also add the button's action, which can be triggered even if the button element itself isn't added to the modal.

\*/

const getButton = (namespace: string, {

text,

value,

className,

closeModal,

}: ButtonOptions, dangerMode: boolean): Node => {

const buttonContainer: any = stringToNode(buttonMarkup);

const buttonEl: HTMLElement = buttonContainer.querySelector(`.${BUTTON}`);

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

buttonEl.classList.add(btnNamespaceClass);

if (className) {

const classNameArray = Array.isArray(className)

? className

: className.split(' ');

classNameArray

.filter(name => name.length > 0)

.forEach(name => {

buttonEl.classList.add(name);

});

}

if (dangerMode && namespace === CONFIRM\_KEY) {

buttonEl.classList.add(DANGER\_BUTTON);

}

buttonEl.textContent = text;

let actionValues: ActionOptions = {};

actionValues[namespace] = value;

setActionValue(actionValues);

setActionOptionsFor(namespace, {

closeModal,

});

buttonEl.addEventListener('click', () => {

return onAction(namespace);

});

return buttonContainer;

};

/\*

\* Create the buttons-container,

\* then loop through the ButtonList object

\* and append every button to it.

\*/

const initButtons = (buttons: ButtonList, dangerMode: boolean): void => {

const footerEl: Element = injectElIntoModal(footerMarkup);

for (let key in buttons) {

const buttonOpts: ButtonOptions = buttons[key] as ButtonOptions;

const buttonEl: Node = getButton(key, buttonOpts, dangerMode);

if (buttonOpts.visible) {

footerEl.appendChild(buttonEl);

}

}

/\*

\* If the footer has no buttons, there's no

\* point in keeping it:

\*/

if (footerEl.children.length === 0) {

footerEl.remove();

}

};

export default initButtons;