import { ButtonList } from './../options/buttons';

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

import { modalMarkup } from '../markup';

import { SwalOptions } from '../options';

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

const { MODAL, OVERLAY } = CLASS\_NAMES;

import initIcon from './icon';

import { initTitle, initText } from './text';

import initButtons from './buttons';

import initContent from './content';

export const injectElIntoModal = (markup: string): HTMLElement => {

const modal: Element = getNode(MODAL);

const el: HTMLElement = stringToNode(markup);

modal.appendChild(el);

return el;

};

/\*

\* Remove eventual added classes +

\* reset all content inside:

\*/

const resetModalElement = (modal: Element): void => {

modal.className = MODAL;

modal.textContent = '';

};

/\*

\* Add custom class to modal element

\*/

const customizeModalElement = (modal: Element, opts: SwalOptions): void => {

resetModalElement(modal);

const { className } = opts;

if (className) {

modal.classList.add(className);

}

};

/\*

\* It's important to run the following functions in this particular order,

\* so that the elements get appended one after the other.

\*/

export const initModalContent = (opts: SwalOptions): void => {

// Start from scratch:

const modal: Element = getNode(MODAL);

customizeModalElement(modal, opts);

initIcon(opts.icon);

initTitle(opts.title);

initText(opts.text);

initContent(opts.content);

initButtons(opts.buttons as ButtonList, opts.dangerMode);

};

const initModalOnce = (): void => {

const overlay: Element = getNode(OVERLAY);

const modal = stringToNode(modalMarkup);

overlay.appendChild(modal);

};

export default initModalOnce;