/\*

\* A list of all the deprecated options from SweetAlert 1.X

\* These should log a warning telling users how to upgrade.

\*/

export const logDeprecation = (name: string): void => {

const details: OptionReplacement = DEPRECATED\_OPTS[name];

const { onlyRename, replacement, subOption, link } = details;

const destiny = (onlyRename) ? 'renamed' : 'deprecated';

let message = `SweetAlert warning: "${name}" option has been ${destiny}.`;

if (replacement) {

const subOptionText = (subOption) ? ` "${subOption}" in ` : ' ';

message += ` Please use${subOptionText}"${replacement}" instead.`;

}

const DOMAIN = 'https://sweetalert.js.org';

if (link) {

message += ` More details: ${DOMAIN}${link}`;

} else {

message += ` More details: ${DOMAIN}/guides/#upgrading-from-1x`;

}

console.warn(message);

};

export interface OptionReplacement {

replacement?: string,

onlyRename?: boolean,

subOption?: string,

link?: string,

};

export interface OptionReplacementsList {

[name: string]: OptionReplacement,

};

export const DEPRECATED\_OPTS: OptionReplacementsList = {

'type': {

replacement: 'icon',

link: '/docs/#icon',

},

'imageUrl': {

replacement: 'icon',

link: '/docs/#icon',

},

'customClass': {

replacement: 'className',

onlyRename: true,

link: '/docs/#classname',

},

'imageSize': {},

'showCancelButton': {

replacement: 'buttons',

link: '/docs/#buttons',

},

'showConfirmButton': {

replacement: 'button',

link: '/docs/#button',

},

'confirmButtonText': {

replacement: 'button',

link: '/docs/#button',

},

'confirmButtonColor': {},

'cancelButtonText': {

replacement: 'buttons',

link: '/docs/#buttons',

},

'closeOnConfirm': {

replacement: 'button',

subOption: 'closeModal',

link: '/docs/#button',

},

'closeOnCancel': {

replacement: 'buttons',

subOption: 'closeModal',

link: '/docs/#buttons',

},

'showLoaderOnConfirm': {

replacement: 'buttons',

},

'animation': {},

'inputType': {

replacement: 'content',

link: '/docs/#content',

},

'inputValue': {

replacement: 'content',

link: '/docs/#content',

},

'inputPlaceholder': {

replacement: 'content',

link: '/docs/#content',

},

'html': {

replacement: 'content',

link: '/docs/#content',

},

'allowEscapeKey': {

replacement: 'closeOnEsc',

onlyRename: true,

link: '/docs/#closeonesc',

},

'allowClickOutside': {

replacement: 'closeOnClickOutside',

onlyRename: true,

link: '/docs/#closeonclickoutside',

},

};