import {

swal,

removeSwal,

$,

$$,

CLASS\_NAMES,

} from './utils';

const {

CONTENT,

MODAL\_TEXT

} = CLASS\_NAMES;

afterEach(() => removeSwal());

describe("show content", () => {

test("shows no content by default", () => {

swal("Hello");

expect($$(CONTENT).length).toBe(0);

});

test("shows input when using content: 'input'", () => {

swal({

content: "input",

});

const inputSelector = `${CONTENT}\_\_input`;

expect($$(CONTENT).length).toBe(1);

expect($$(inputSelector).length).toBe(1);

expect($$(inputSelector).get(0).getAttribute('type')).toBeNull();

});

test("can customize input with more advanced content options", () => {

swal({

content: {

element: "input",

attributes: {

placeholder: "Type your password",

type: "password",

},

},

});

const inputSelector = `${CONTENT}\_\_input`;

expect($$(CONTENT).length).toBe(1);

expect($$(inputSelector).length).toBe(1);

expect($$(inputSelector).get(0).getAttribute('type')).toBe('password');

expect($$(inputSelector).get(0).getAttribute('placeholder')).toBe('Type your password');

});

test("can set content to custom DOM node", () => {

let btn = document.createElement('button');

btn.classList.add('custom-element');

swal({

content: btn,

});

expect($$(CONTENT).length).toBe(1);

expect($$('custom-element').length).toBe(1);

});

});

describe("show modal text", () => {

test("transforms newline to break", () => {

swal('Hello\nWorld\n');

expect($(`.${MODAL\_TEXT} br`).length).toBe(2);

});

test("escapes HTML elements", () => {

const text = '<script>bad stuff</script>';

swal(text);

expect($(`.${MODAL\_TEXT} script`).length).toBe(0);

expect($$(MODAL\_TEXT).text()).toEqual(expect.stringMatching(text));

});

});