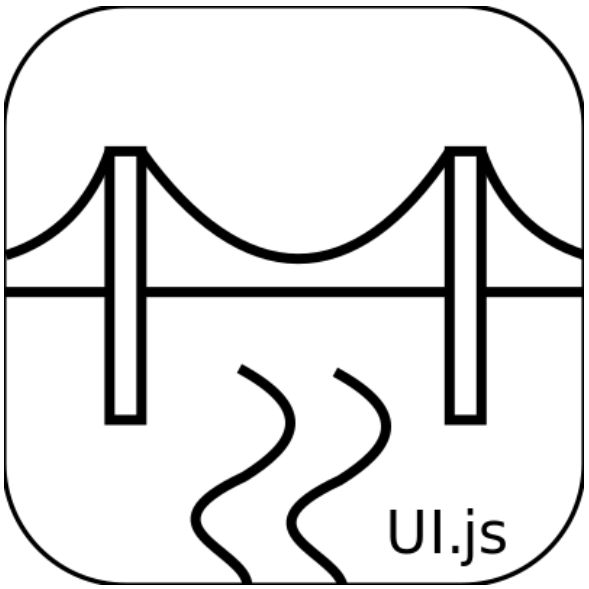

Quick guide for SBUI.js to create User Interfaces

1. Introduction

This guide will show how to create a UI from scratch using the SBUI.js library, now to start you have to download the SBUI.js library, you can download just the SBUI.js file or you can download the total project.

1.1 File: index.html

This the html of your web page, you use like your own html coded pages.

1.2 File: drawning.svg

The logo of SBUI.js library.

1.3 File: index.js

The JS file, which will container your source code.

1.4 File: SBUI.js

The JS file, which container the SBUI.js library source code.

2. Functions

We have some functions, which will be explained.

To create a component

SBUI.Element(elem, props);

elem -> you can pass an Element or a Element string name like “div” or “span”
props -> will be the events of a component attribute, and anothers, using JSON.
childs -> to append the childs in a component

will be the same as:

```
SBUI.Element(“div”, {});
```

2.1 Custom props

style -> in this case you have to declare style at props using a json format like React.

```
SBUI.Element(“div”, {
        style: { width: ‘100%’, height: ‘100%’ }
});
```

ref -> to a useRef hook, which will be explained later.

```
const divRef = SBUI.useRef();

SBUI.Element(“div”, {
    ref: divRef
});

console.log(divRef);
```

childs -> A array of SBUI.js Elements

```
const div = SBUI.Element(“div”);

SBUI.Element(“div”, {
      childs: [div]
});
```

if sompe props value seted null, the value will be removed, for example;

```
const div = SBUI.useState(“testing”);
SBUI.Element(“div”, {
      id: div
});

//will remove the id value from element;
div.setState(null);
```

have to store this in a variable

SBUI.render(selector, element)

Will delete everything on the page, and append the SBUI element.

selector: have to be a string with the selector. using the querySelector patern.
element: have to be the SBUI.Element();

```
const div = SBUI.Element(“div”);

SBUI.render(“body”, div);
```

don’t have to store this in a variable

SBUI.addToElement(selector, element)

it’s equals the SBUI.render, but this will not delete the already the rendered elements.

```
const div = SBUI.Element(“div”);

SBUI.addToElement(“body”, div);
```

don’t have to store this in a variable

SBUI.useState(value)

useState it’s the hooks to store a value, if you change the value, the useState hook will be identify a change and execute the func.

```
const text = SBUI.useState(“hello”);

SBUI.Element(“div”, {
      innerText: text
});
```

have to store this in a variable

SBUI.useRef()

```
const div = SBUI.useRef();

SBUI.Element(“div”, {
       ref: div
});
```

have to store this in a variable

SBUI.useEffect(func, array)

useEffect is a hook, will recognize the array, each array have something to do.

func -> is the function that has to be executed.
array -> it's the array to represented what have to do with the func.
if the array is undefined. the code is updating all time, like a update function.

```
SBUI.useEffect(() => {
//this function don't will be stop of executed
})
```

if the array is declared, but don't have nothing inside, this will be executed at once.

```
SBUI.useEffect(() => {
//will executed at once
},[])
```

if the array is declared, and have some hooks inside, he will se a change on a hook, and when a hook change will be executed the func.

```
const div = SBUI.useState("");

SBUI.useEffect(() => {
//this will be executed when div value change
},[div])
```

SBUI.useMemo()

Store a value to use later

this will be return an array with, first element the actual memo value, and the second a function to change the memo value.

```
const value = SBUI.useMemo();
value[1](“testing”);


//declared again;
const [ first, second ] = SBUI.useMemo();
console.log(first);

//setting the variable value, will not edit the value of first variable

second(“updating”);
```

have to store this in a variable

SBUI.alreadyInPage

a json if all SBUI elements already rendered in page, to store this value you just have to add the sbuielement to your HTML element. You can use ate SBUI.Element() function.

```
<!--HTML-->
<div sbuielement=”id”></div>


// JS

SBUI.alreadyInPage.id // will get the reference of id you can use at SBUI.Element()
```