

vaadin}>



Vaadin in the real world - building the JU Vienna Raffle!

Too Much JS? Bring Your UI back to Java with Vaadin



Sebastian Kühnau
Vaadin Enthusiast

Why Web Development is hard with Java?

Full-Stack with Vaadin Flow

```
@Route("hello-word")
public class HelloWorldView extends VerticalLayout {

    public HelloWorldView(GreetingService service) {

        TextField name = new TextField("Name");
        Button sayHello = new Button("Greet");
        sayHello.addClickListener(e -> {
            service.saveGreeting(name.getValue());
        });

        Grid<> grid = new Grid<Greeting>();
        grid.addColumn(Greeting::getMessage);
        grid.setItems(service.getMessages());

        add(name, sayHello, grid);
    }
}
```

Get Started in a few steps

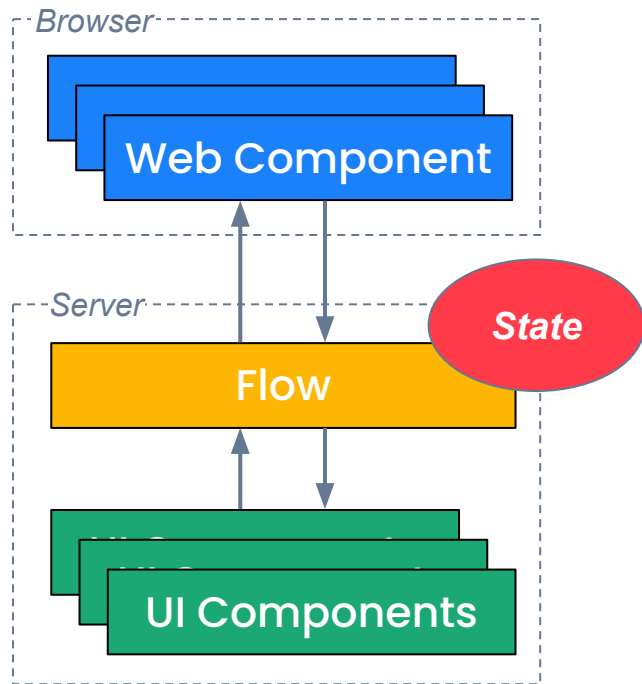
```
<dependency>
  <groupId>com.vaadin</groupId>
  <artifactId>vaadin</artifactId>
</dependency>

...

<plugin>
  <groupId>com.vaadin</groupId>
  <artifactId>vaadin-maven-plugin</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>prepare-frontend</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Vaadin from the bird's eye view

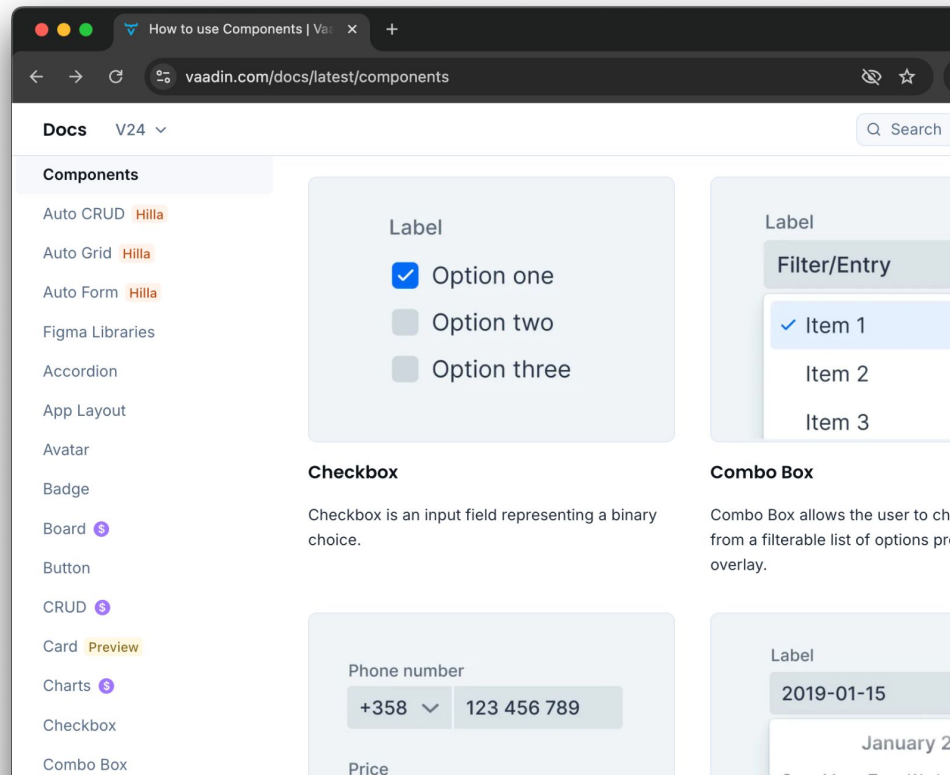
Go with the Flow



```
sebastian@a8bb571f-24f4-4c93-bcd6-a6c8e5e80c2f:~  
➔ ~ mvn package -Pproduction -Pnative native:compile
```

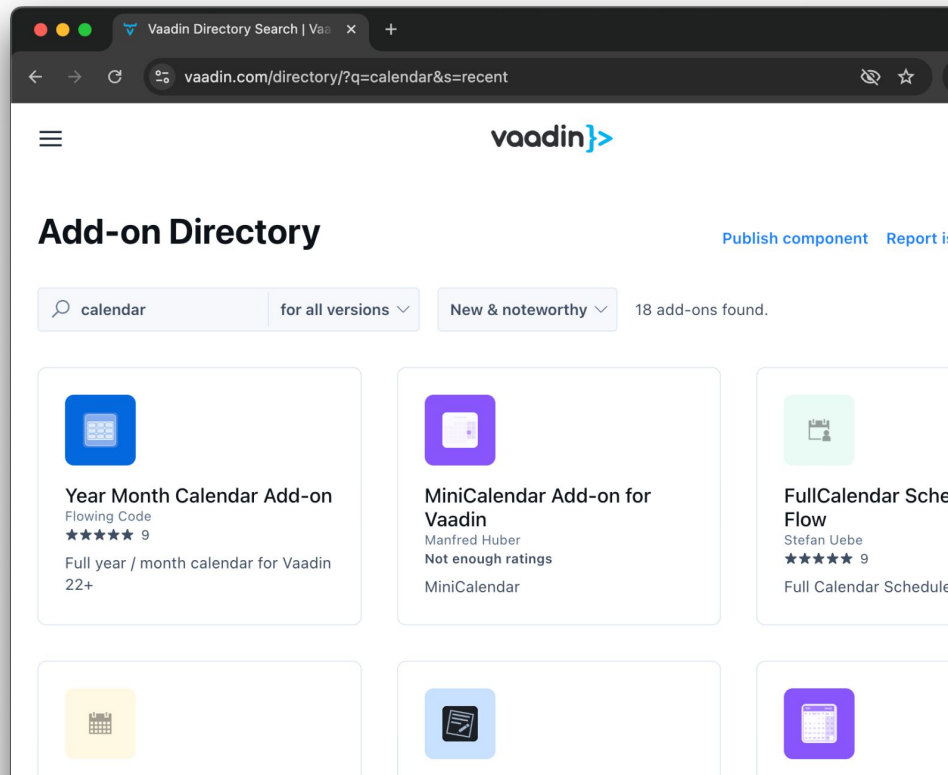
+55 Out of the box UI Components

A set of modern UI components for business web apps

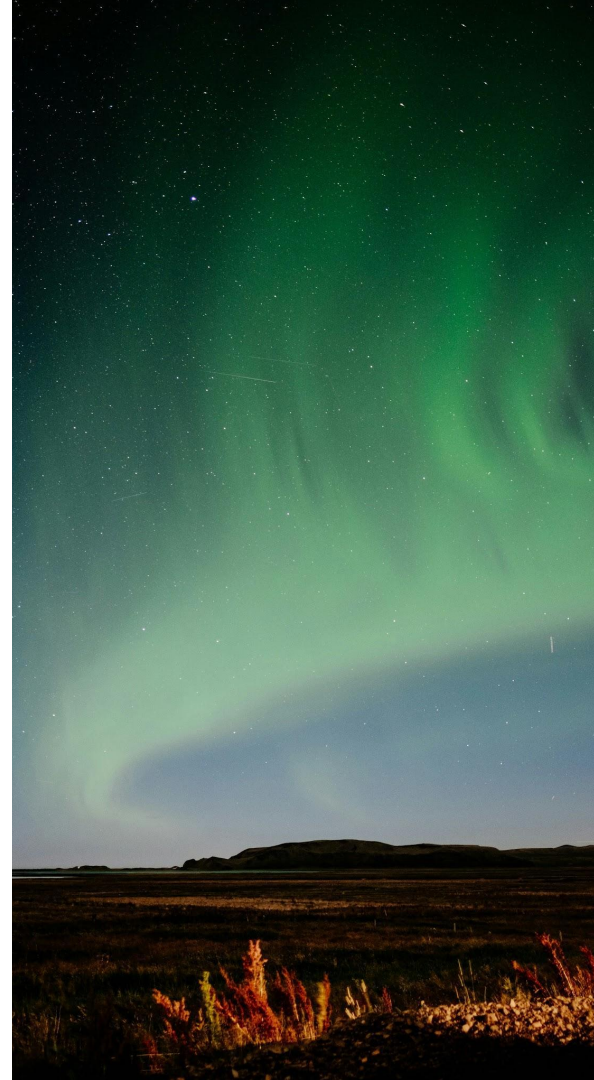


VAADIN ADD-ON DIRECTORY

+2000
open source
components
for Vaadin



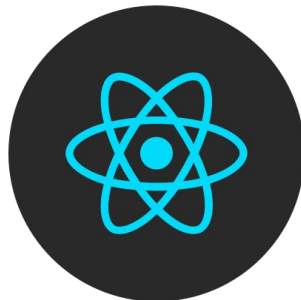
And then ... ?



You are not limited!

Other option for components

```
<!DOCTYPE html
<html lang="de">
<head>
  <meta charset=UTF-8>
  <title Canvas Beispiel>
</head>
</script>
  ccanvas = document.getElementById(canavs
  ctx.cacrontext1pc("2d")
  <script>
    ccanvass id==meinCanvas;
    arc(150,75, 50, 0, 2 *Mätf.1 Math.PI)
    ctx.fillStyle =
    ctx.fill
  </schnt>
</body>
</html>
```



Let's take a look at other options!

Native HTML "components"

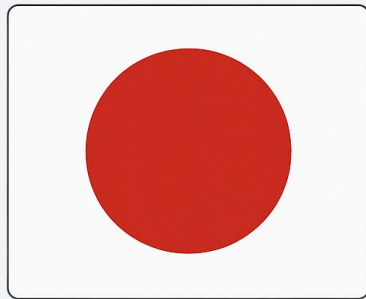
HTML

```
<canvas id="myCanvas"
  width="300" height="150"
  style="border:1px solid #000000;"></canvas>

<script>
  const canvas = document.getElementById("myCanvas");
  const ctx = canvas.getContext("2d");

  //draw circle
  ctx.beginPath();
  ctx.arc(150, 75, 50, 0, 2 * Math.PI);
  ctx.fillStyle = "red";
  ctx.fill();
</script>
```

Browser



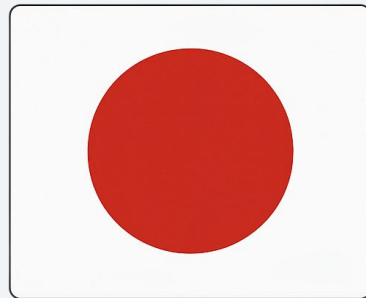
With usage of Element-API

→ Vaadin Flow

Vaadin

```
public MyCanvasView() {  
    Element canvas = new Element("canvas");  
    canvas.setAttribute("id", "myCanvas");  
    canvas.getStyle().setBorder("1px solid #000");  
    canvas.setAttribute("width", "300px");  
    canvas.setAttribute("height", "150px");  
    canvas.executeJs("""  
        ...  
        """);  
  
    getElement().appendChild(canvas);  
}
```

Browser



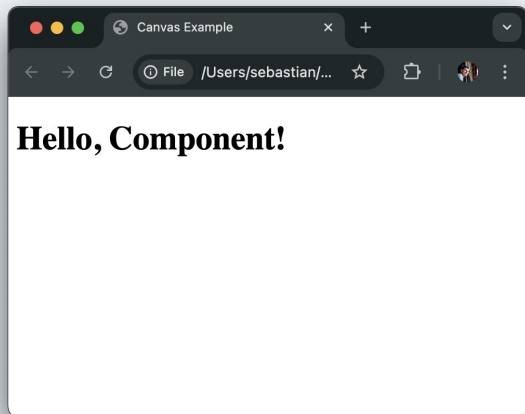
Let's take a look at other options!

JS based components

JS + HTML

```
/** JavaScript **/  
class HelloWorld extends HTMLElement {  
  connectedCallback() {  
    this.innerHTML = `<p>Hello, Component!</p>`;  
  }  
}  
customElements.define('hello-world', HelloWorld);  
  
/** HTML **/  
<hello-world></hello-world>
```

Browser



With usage of Element-API

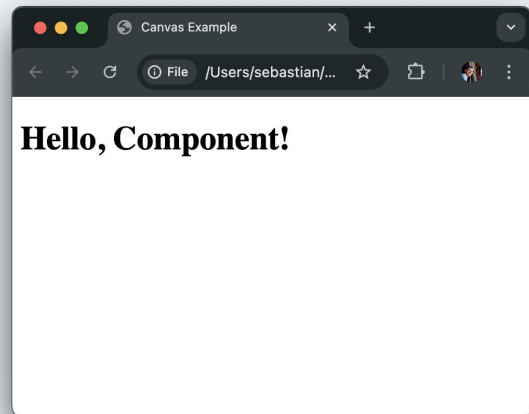
→ Vaadin Flow

Vaadin

```
@JsModule("../components/demo/js-component.js")
@Tag("hello-world")
public class HelloWorldComponent extends Component
    implements HasSize {

    public void setText(String text) {
        this.getElement().setProperty("innerHTML", text);
    }
}
```

Browser



Let's take a look at other options!

React Components

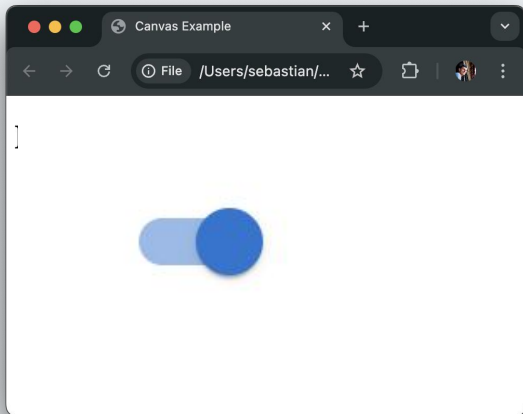
NPM + TSX

```
npm install @mui/material @emotion/react @emotion/styled

const label = { inputProps: {
  'aria-label': 'Switch demo'
} };

export default function BasicSwitches() {
  return (
    <div>
      <Switch {...label} defaultChecked />
    </div>
  );
}
```

Browser



Integration with React Adapter

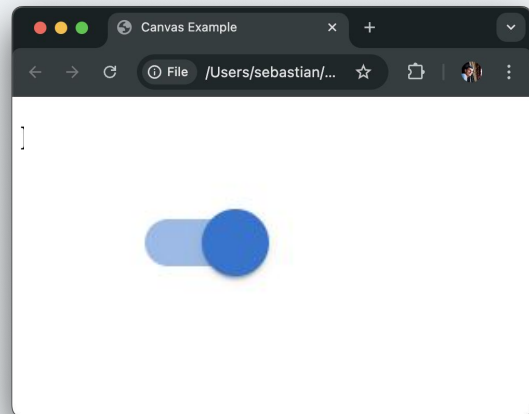
→ Vaadin Flow

Since V24.4

Vaadin

```
class ReactSwitchElement extends ReactAdapterElement {  
  protected render(hooks: RenderHooks): ReactElement | null {  
    return <Switch defaultChecked />;  
  }  
}  
customElements.define('react-switch', ReactSwitchElement);  
  
@NpmPackage(value = "@mui/material", version = "^7.1.0")  
@JsModule("./components/demo/react-switch.tsx")  
@Tag("react-switch")  
public class SwitchComponent extends ReactAdapterComponent { }
```

Browser



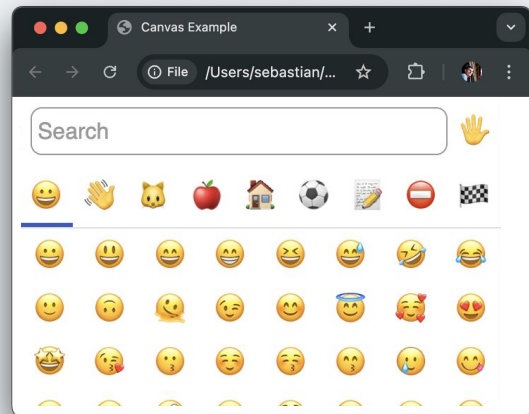
Let's take a look at other options!

WebComponent

NPM + TSX

```
npm install emoji-picker-element  
  
import 'emoji-picker-element';  
  
<emoji-picker></emoji-picker>
```

Browser



Native WebComponent support in Vaadin

→ Vaadin Flow

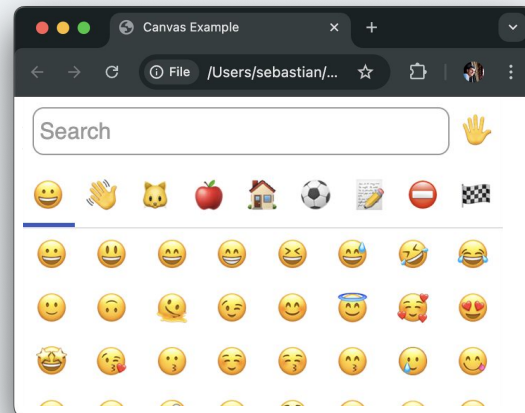
Vaadin

```
@NpmPackage(value = "emoji-picker-element", version = "^1.26.3")
@JsModule("emoji-picker-element/index.js")
@Tag("emoji-picker")
public class EmojiComponent extends Component
    implements HasSize {

    public EmojiComponent() {

    }
}
```

Browser



Let's code

<http://bit.ly/44M4Ysp>

seb@vaadin.com



WE'D LIKE TO TALK

Contact us

Website

<https://vaadin.com>

Email

seb@vaadin.com

Twitter

[@vaadin](https://twitter.com/vaadin)

