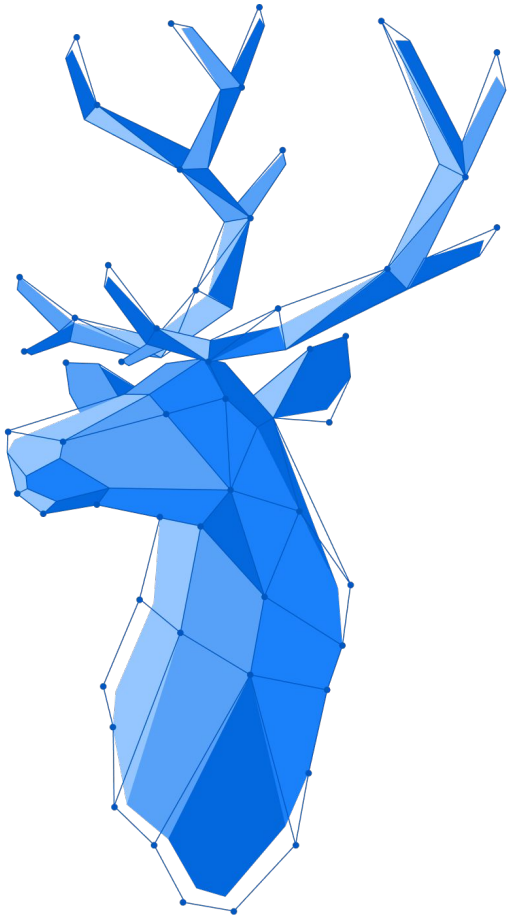
vaadin

Introduction to Vaadin Flow



What is Vaadin good for?

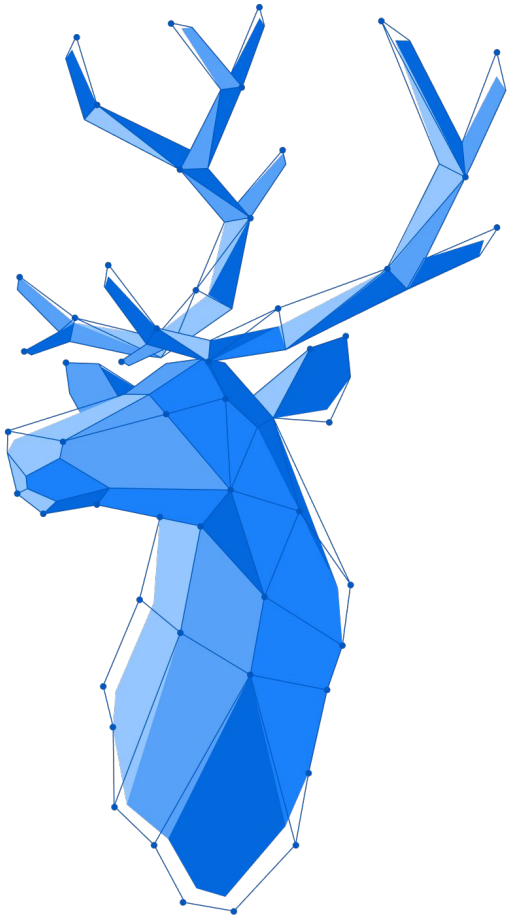
Vaadin training set

Vaadin Foundation

- **Introduction**
- Layouting
- Creating Forms
- Data Lists with Grid
- Routing and Navigation
- Theming and Styling Applications

Agenda

- What is Vaadin?
- Developing in Java
- Components
- Exercise



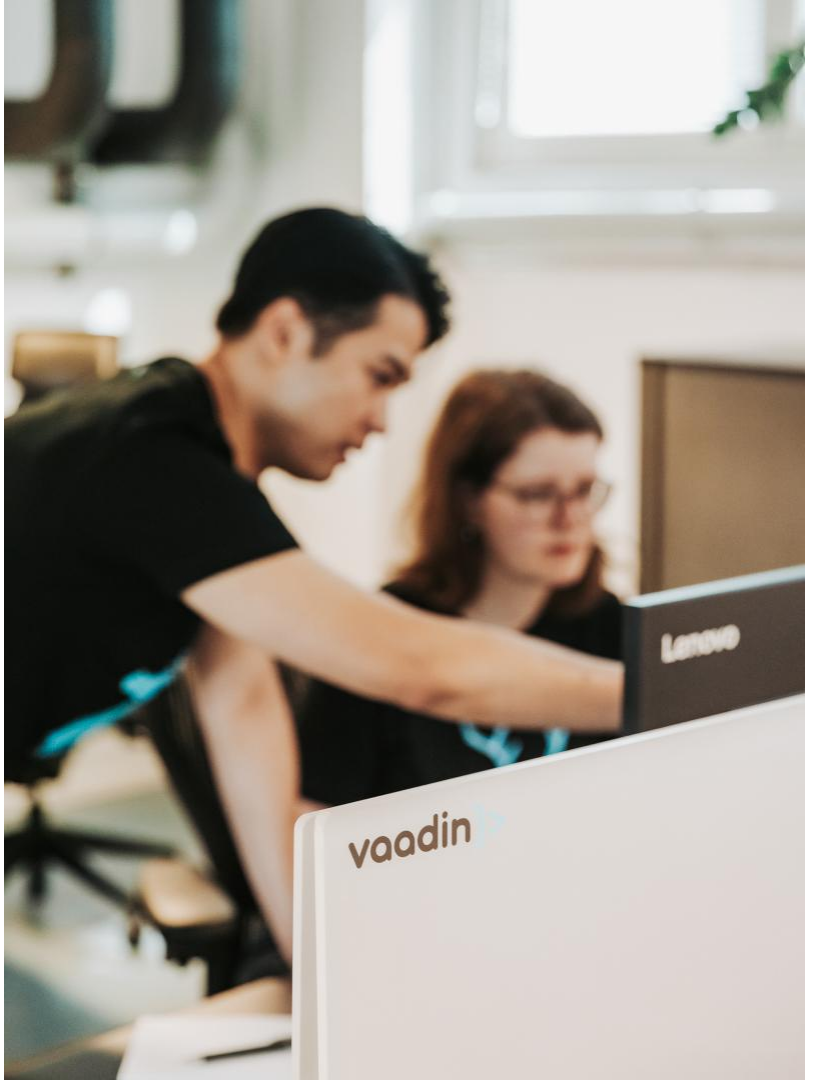
What is Vaadin?

Vaadin

Finnish word meaning
“female reindeer”

Vaadin

A company creating Open Source web development tools and components



What is Vaadin Flow?

Vaadin Flow is a web framework for Java. It comes with a suite of components and tools for building web applications.

What is Hilla?

Hilla is another web framework developed by Vaadin. Hilla focuses on **TypeScript** development with a **Spring Boot** backend. In this training, we'll focus on Flow.

How is Vaadin Flow different?

Flow is a framework focused on building **Progressive Web Applications** with **Java**

Typesafe and **secure** end-to-end integration gives unparalleled developer experience

Vendor-backed open source is a **stable** foundation to build on

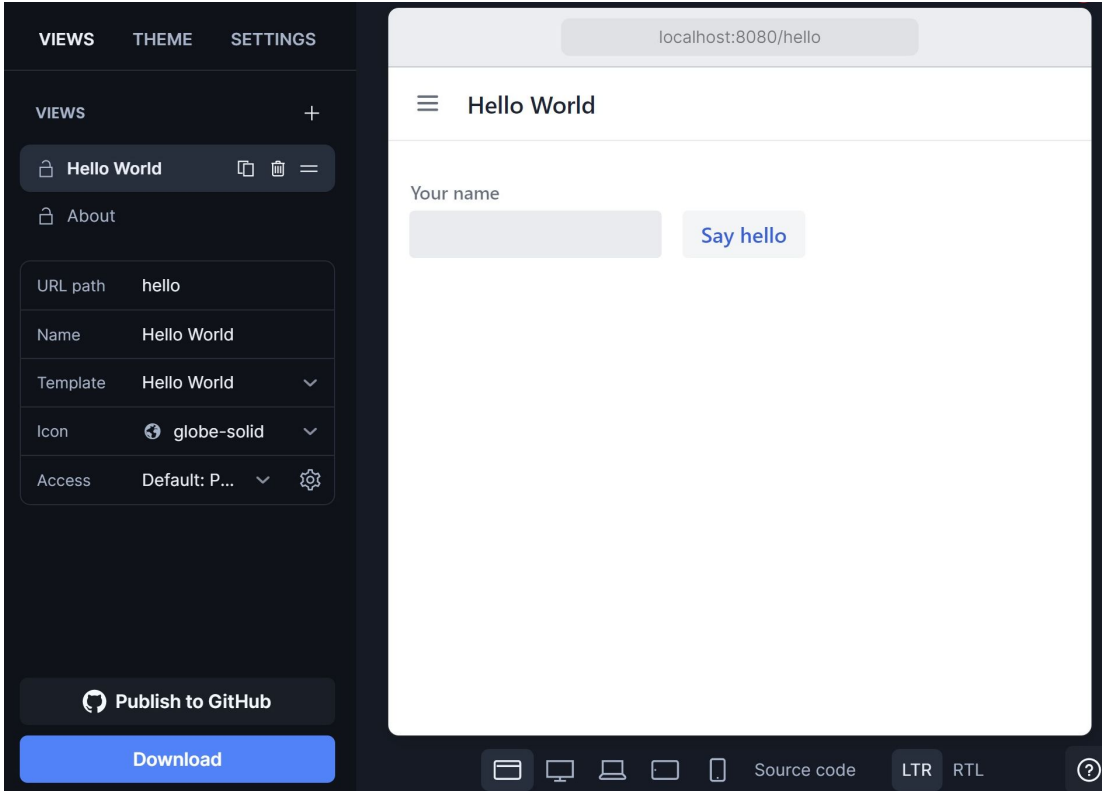
Best-in-class **component** set makes building highly **usable** and **accessible** applications easy

Developing in Java

vaadin

https://start.vaadin.com/

Getting started



Project structure

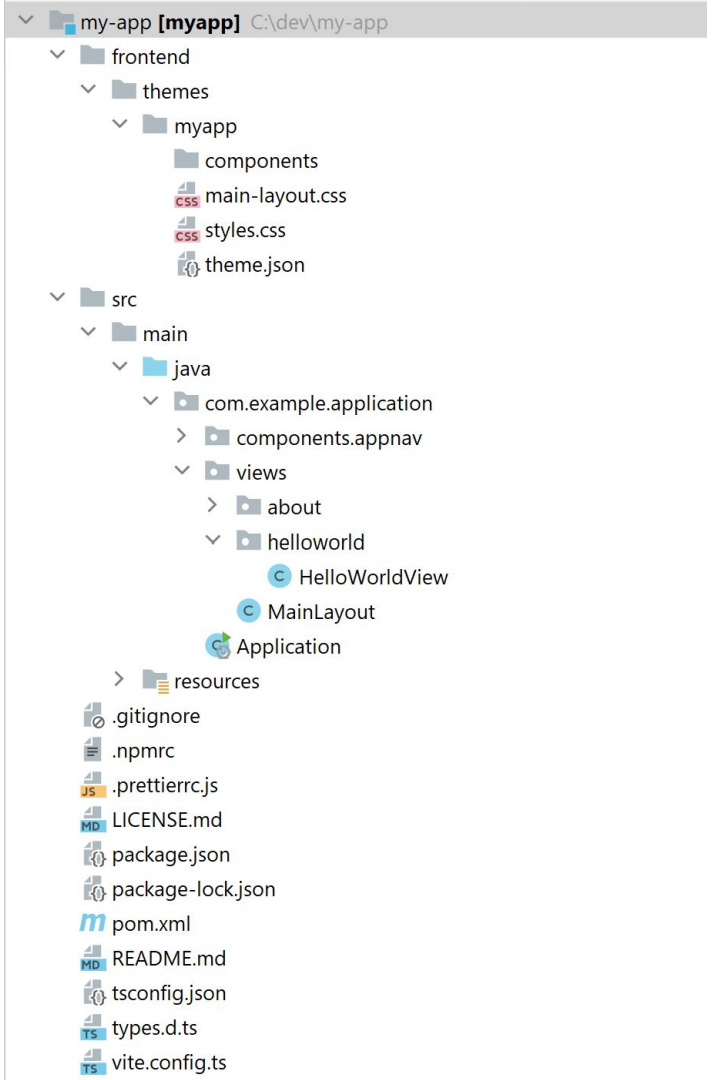
src/main/java for Java sources

pom.xml Maven project configuration

frontend for client side resources, e.g. js, HTML, CSS files

node _modules, package.json, package-lock.json for npm

vite.config.ts auto generated file for Vite, frontend build tooling



Java UI

Hello World!

```
@Route(value = "")
public class HelloWorldView extends VerticalLayout {
   public HelloWorldView() {
        Button sayHello = new Button("Say hello");
        sayHello.addClickListener(e -> {
              Notification.show("Hello world!");
        });
        add(sayHello);
   }
}
```

Java UI with a backend service

Hello World!

```
@Route(value = "")
public class HelloWorldView extends VerticalLayout {

    private MessageBean messageBean = new MessageBean();

    public HelloWorldView() {
        Button sayHello = new Button("Say hello", (e ->
                Notification.show("Hello " + messageBean.getMessage())
        ));
        add(sayHello);
    }
}
```

vaadin

How to run the application?

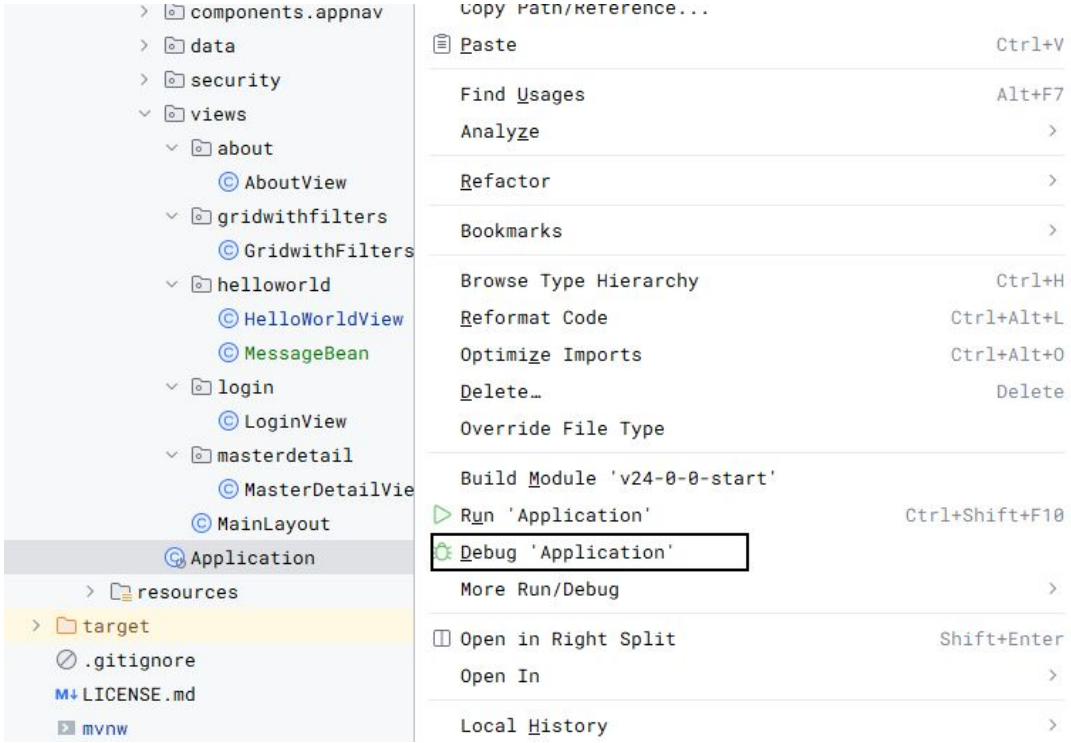
1. Run the main method of Application.java
2. Wait for the application to start
3. Open http://localhost:8080 in a browser to view the application

Alternatively:

1. Run the spring-boot:run Maven goal
2. Wait for the application to start
3. Open http://localhost:8080 in a browser to view the application

Debug (IntelliJ)

- Find Application.java
- Right-click and choose “Debug ‘Application’”

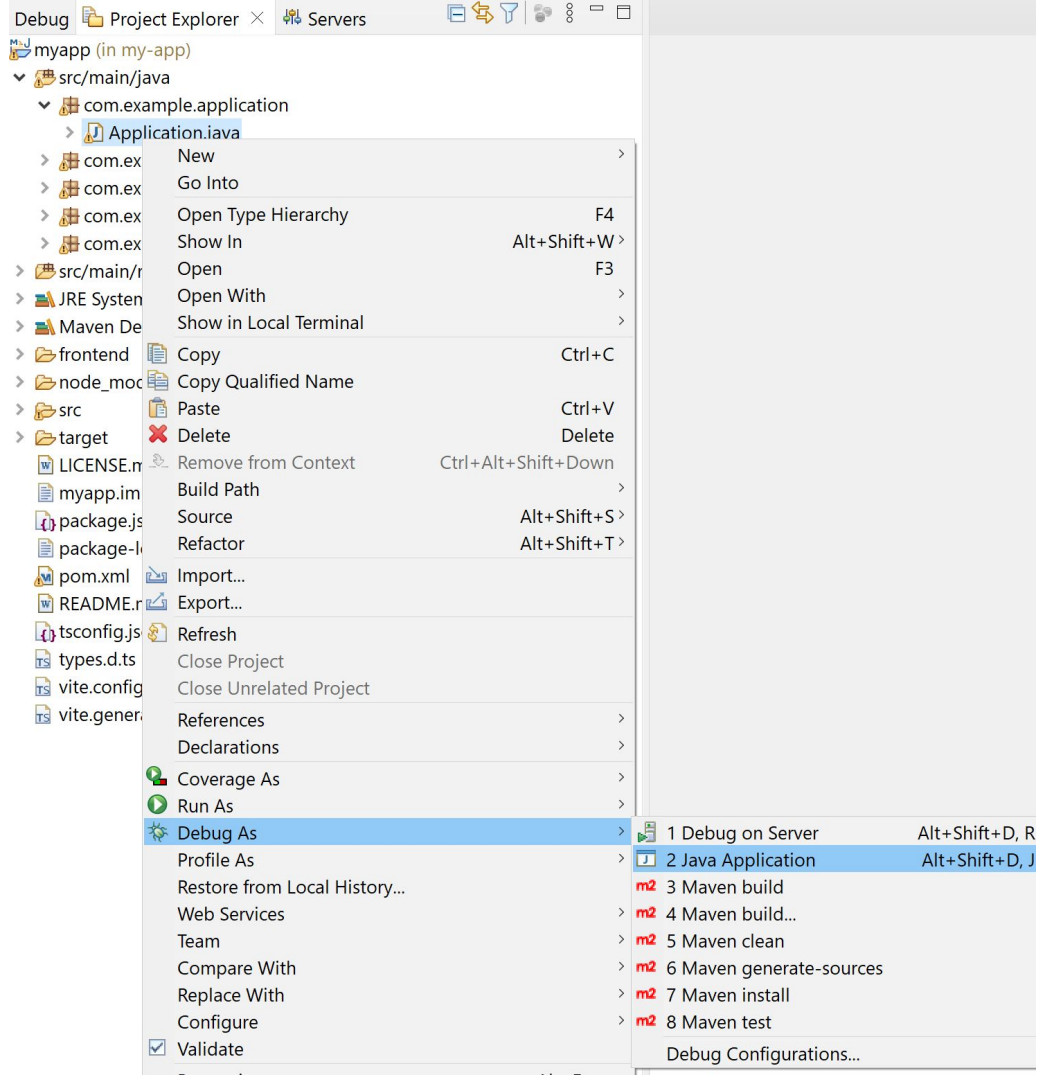


vaadin

Debug (Eclipse)

- Find Application.java
- Right-click and choose “Debug As” → “Java Application”

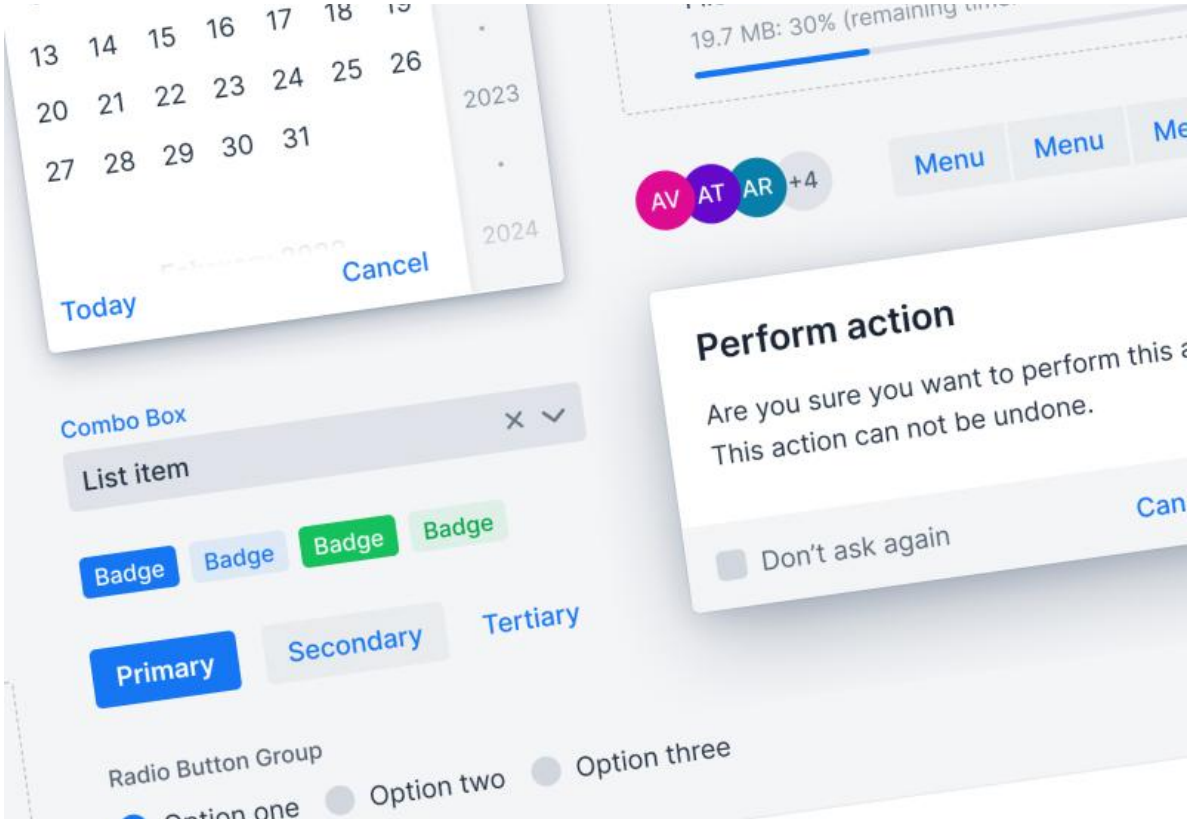
Note: if you’re running plain Eclipse, you’ll need to continue past a SilentExitException.
You can install the Spring Tools 4 Plugin from Eclipse Marketplace to work around this issue



Components

Components

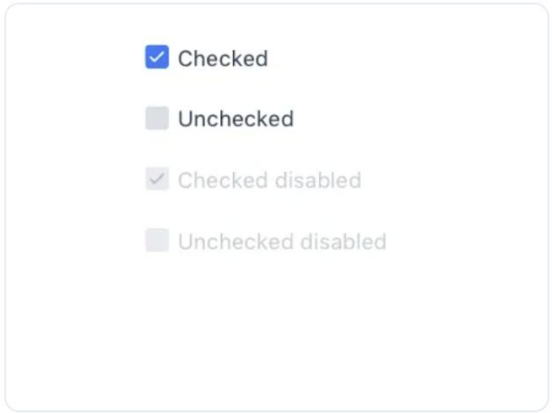
- Built-in components
- Architecture
- Web components



Built-in components

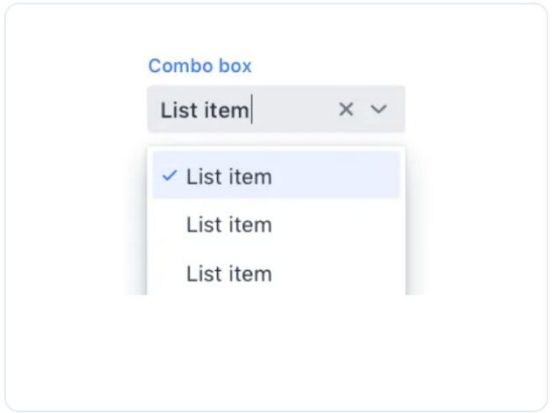
- Form inputs
- Visualization and interaction
- Layouts
- HTML components

Form inputs



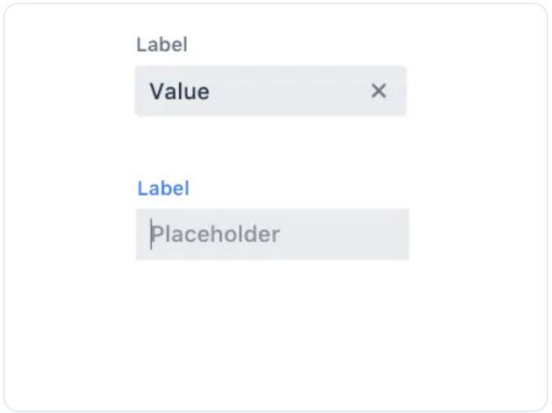
Checkbox

Checkbox is an input field representing a binary choice.



Combo Box

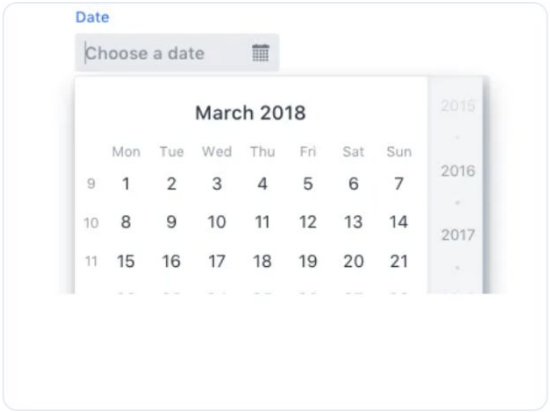
Combo Box allows the user to choose a value from a filterable list of options presented in an overlay.



Text Field

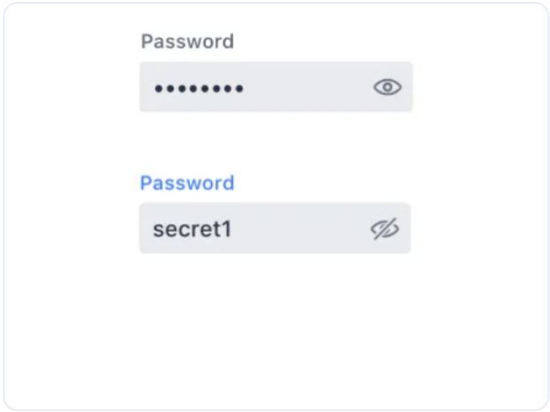
Text Field allows the user to input and edit text.

More form inputs



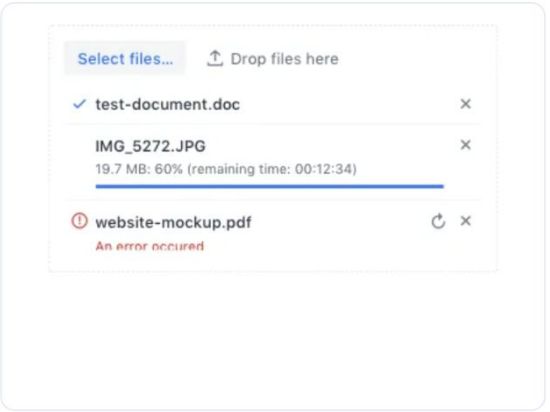
Date Picker

Date Picker is an input field that allows the user to enter a date by typing or by selecting from a calendar overlay.



Password Field

Password Field is an input field for entering passwords.

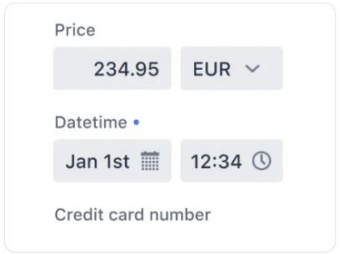


Upload

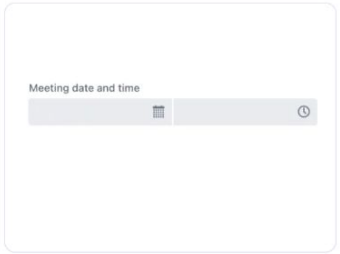
Upload is a component for uploading one or more files.

vaadin

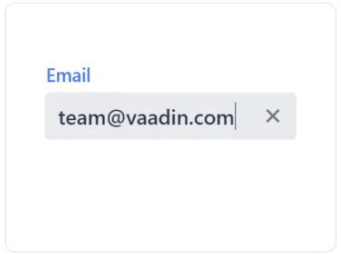
Even more form inputs



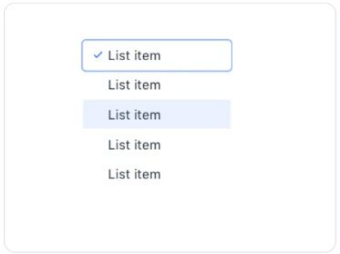
Custom Field



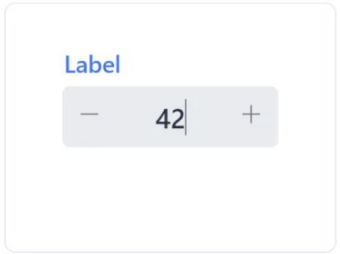
Date Time Picker



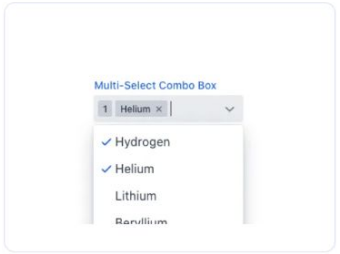
Email Field



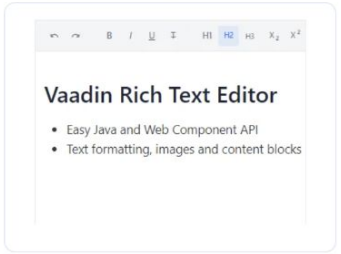
List Box



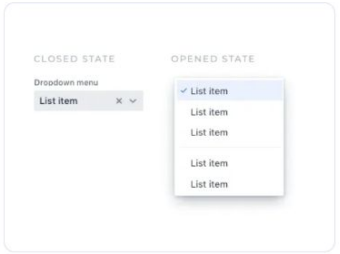
Number Field



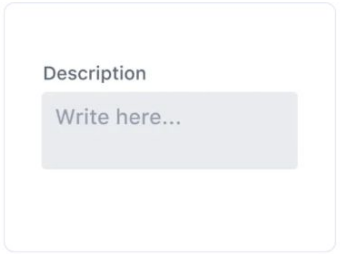
Multi-Select Combo Box



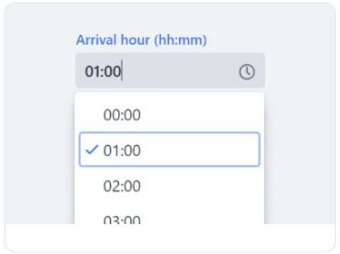
Rich Text Editor



Select



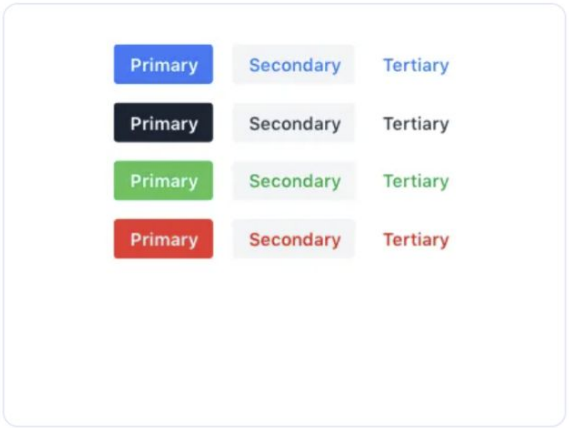
Text Area



Time Picker

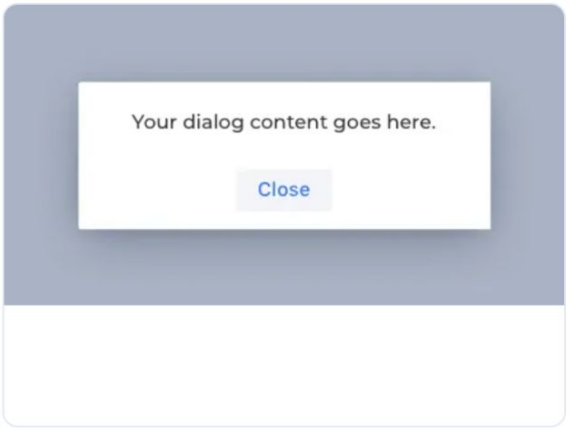
vaadin

Visualization & interaction



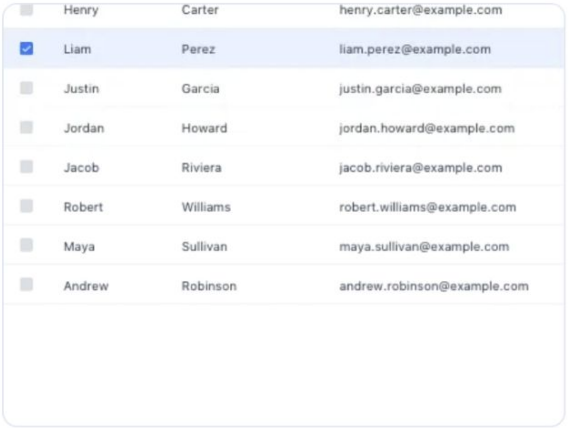
Button

The Button component allows users to perform actions.



Dialog

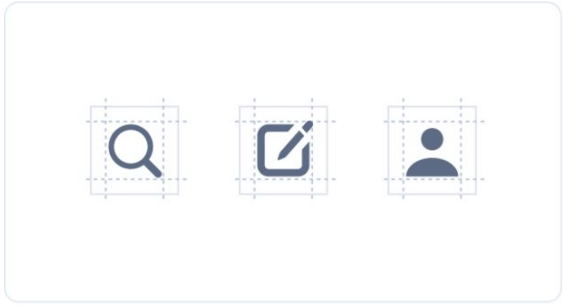
Dialog is a small window that can be used to present information and user interface elements in an overlay.



Grid

Grid is a component for showing tabular data.

More visualization & interaction

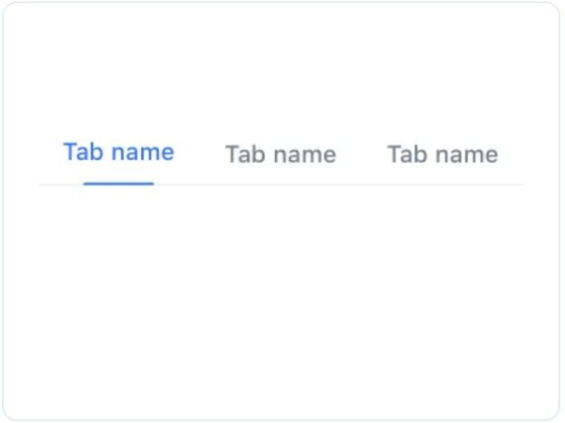


Icons

Over 600 built-in icons for business applications.

Progress Bar

Progress Bar shows the completion status of a task or process.

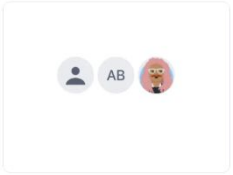


Tabs

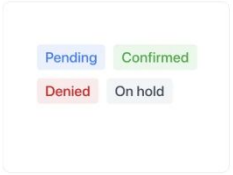
Tabs are used to organize and group content into sections that the user can navigate between.

Even more visualization & interaction

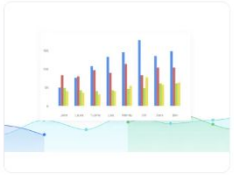
Accordion



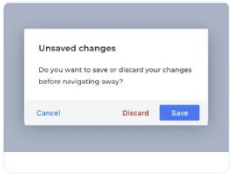
Avatar



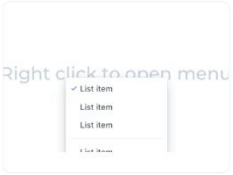
Badge



Charts



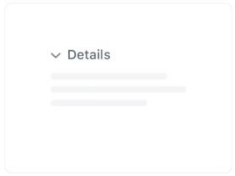
Confirm Dialog



Context Menu

Cookie Consent

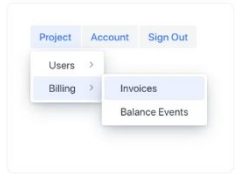
CRUD



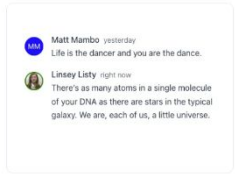
Details

Grid Pro

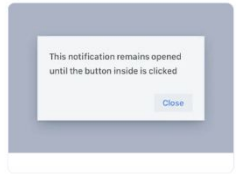
Map



Menu Bar



Message List



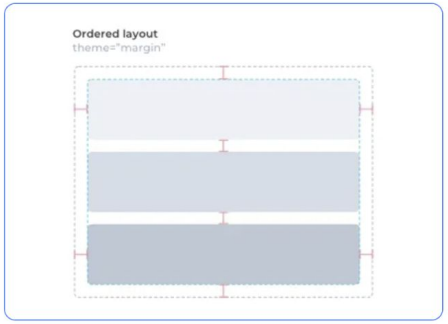
Notification

Scroller

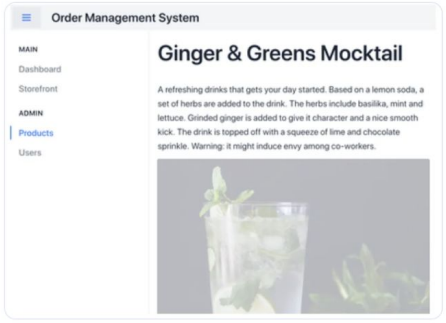
Virtual List

vaadin

Layouts

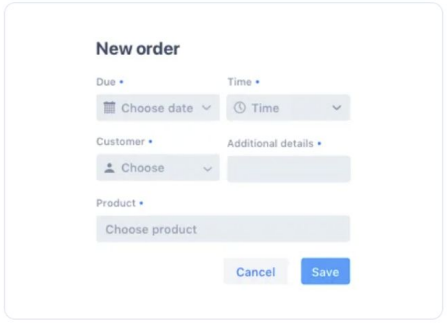


Basic Layouts

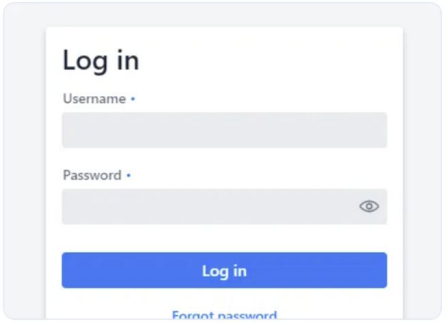


App Layout

Board



Form Layout



Login

Split Layout

vaadin

HTML components

Anchor
Article
Aside
Description
List
Div
Emphasis

H1
H2
H3
H4
H5
H6
Hr

Header
Footer
IFrame
Input
Label
ListItem
Main

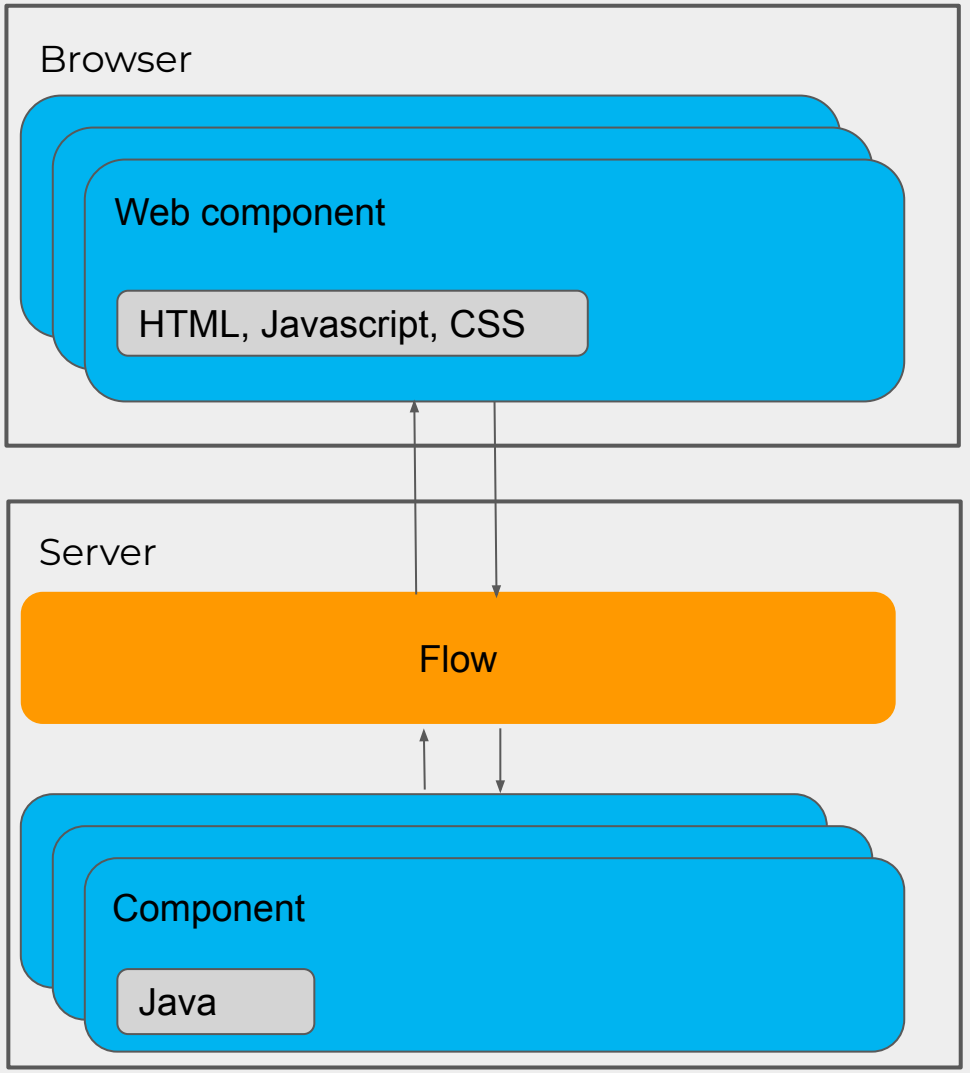
NativeButton
Nav
OrderedList
Paragraph
Section
Span
UnorderedList

Architecture

Client side: Web Components

Server side: Java

Communication: Flow



Web components?

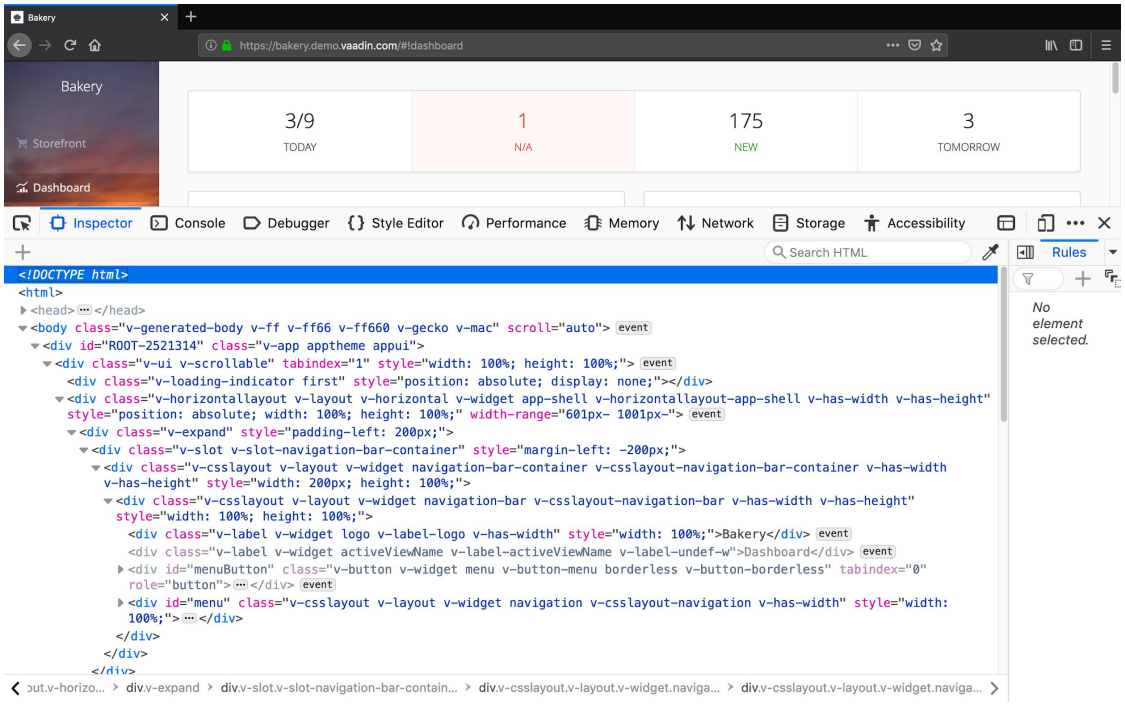
Custom web components build on existing web standards that work across modern browsers

The standards allow easily extending HTML with new tags with encapsulated styling and custom behavior

Semantics

Before:

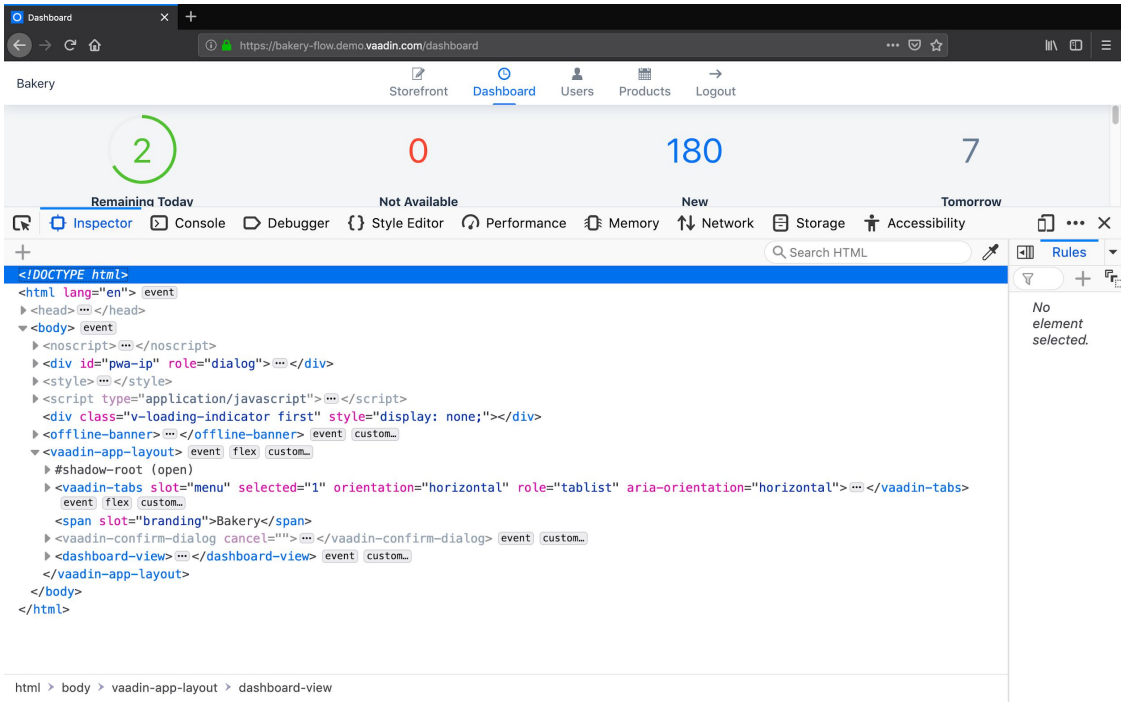
- soup



Semantics

Now:

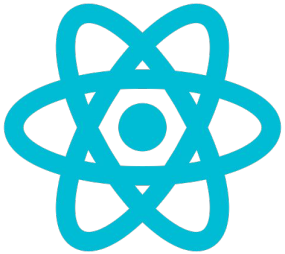
- New tags for your browser
- Client-side component model to match server-side components



Reusability

Web components

- Based on properties and DOM events
- Framework agnostic



Stability

Web components

- Based on standards
- Web standards live long

Exercise

Tools

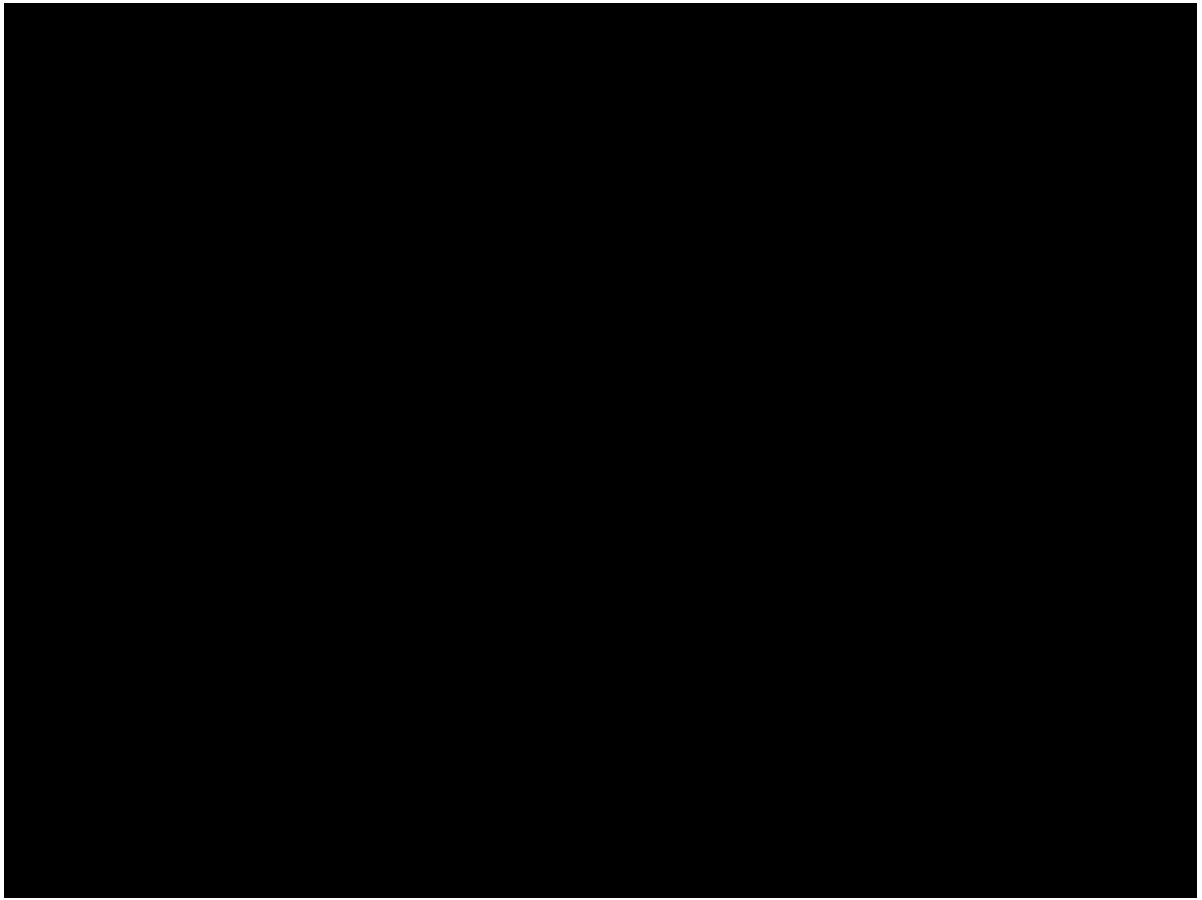
Tools

TestBench
Designer
Multiplatform Runtime
Collaboration Kit
Design System Publisher
Observability Kit
Azure Cloud Kit
SSO Kit
Swing Kit

TestBench

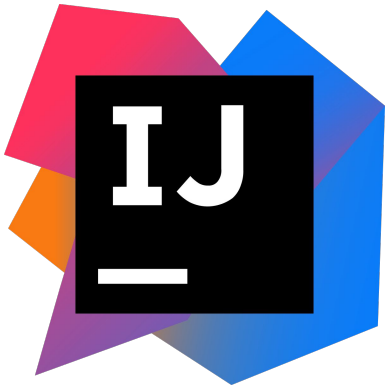
- Vaadin TestBench is a tool for creating and running browser-based integration tests for your Vaadin application
- The UI unit testing feature also allows browser-less and container-less testing directly in the JVM

TestBench in action

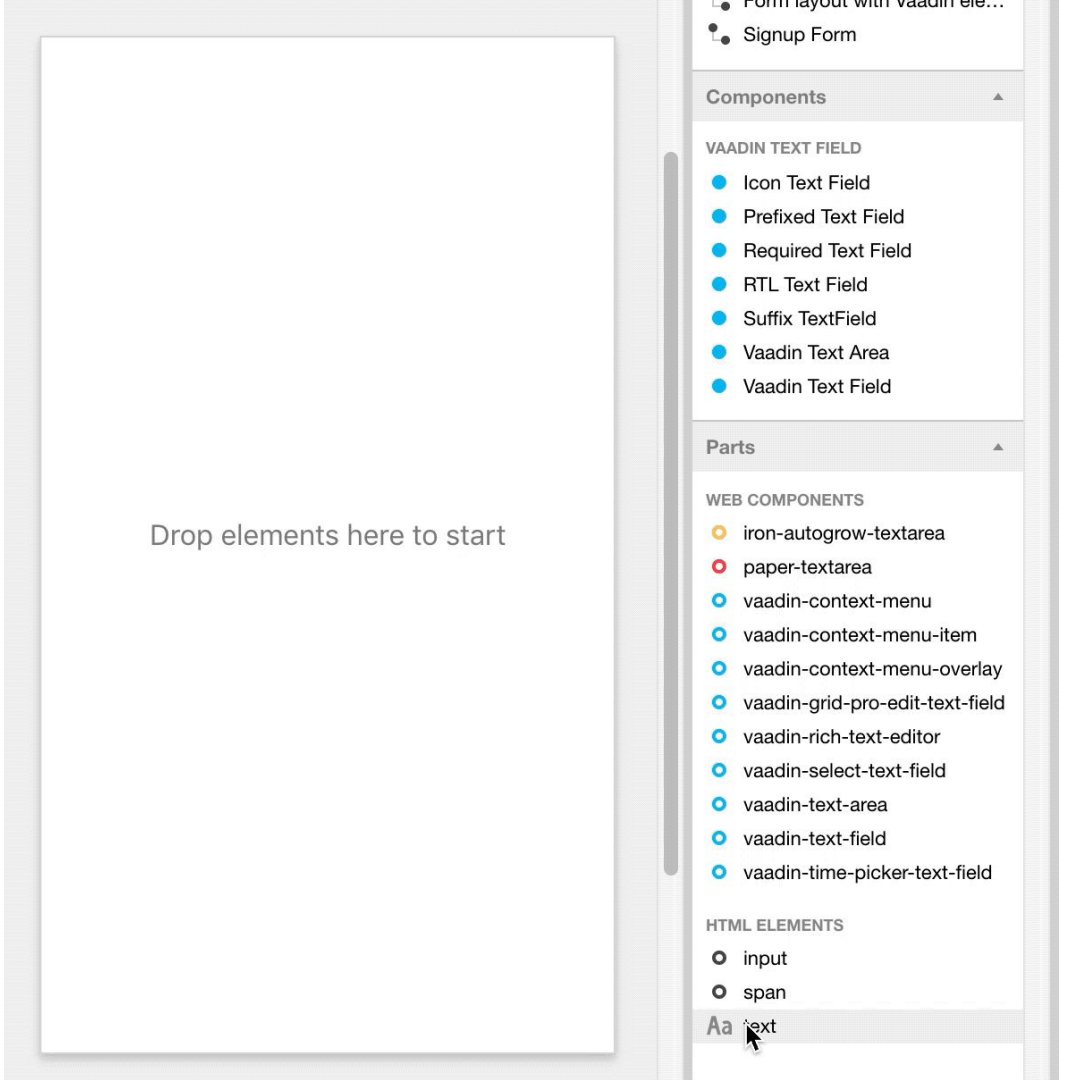


Vaadin Designer

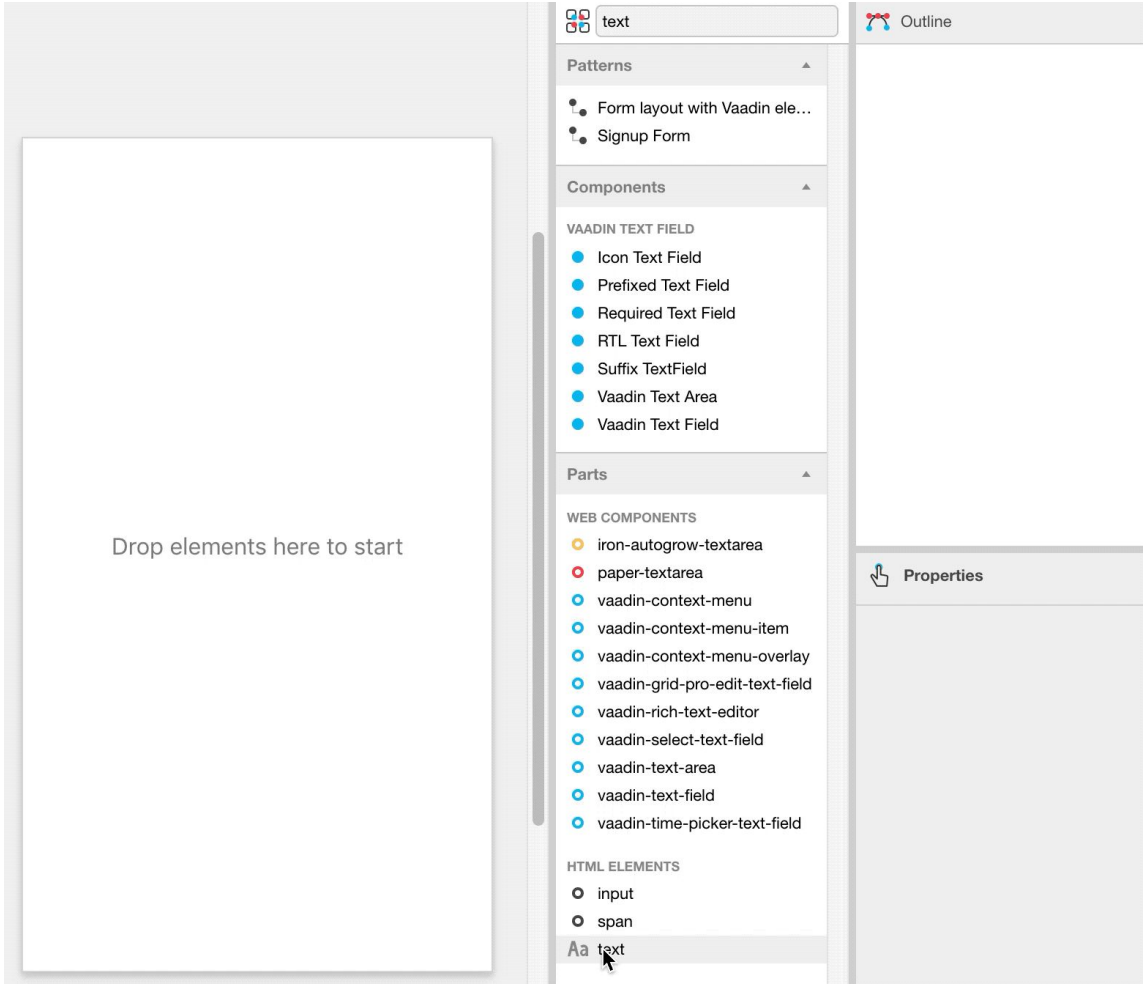
IDE plugin for Eclipse and IntelliJ Idea for building Vaadin applications with Drag&Drop



Drag and drop to canvas



Drag and drop to outline



Progressive Web Apps

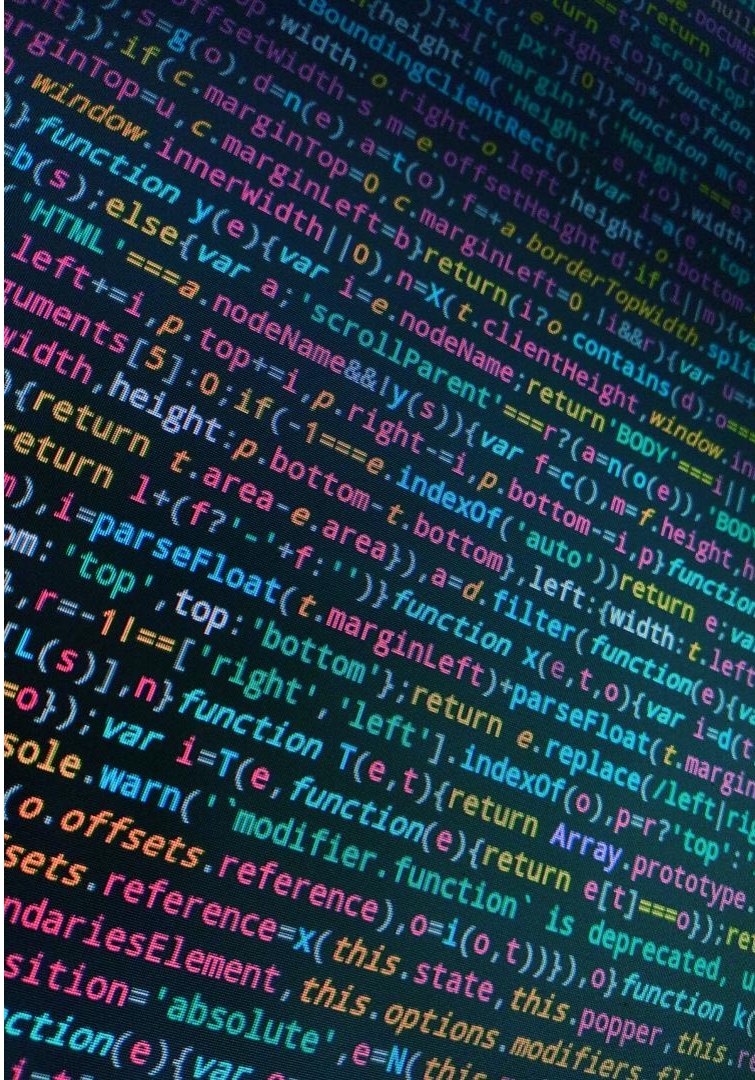
Web apps with native-like experiences

vaadin

vaadin

PWA How

- Web App Manifest
- Service Worker
- Icons
- Offline Support
- Header information
- Installation prompt



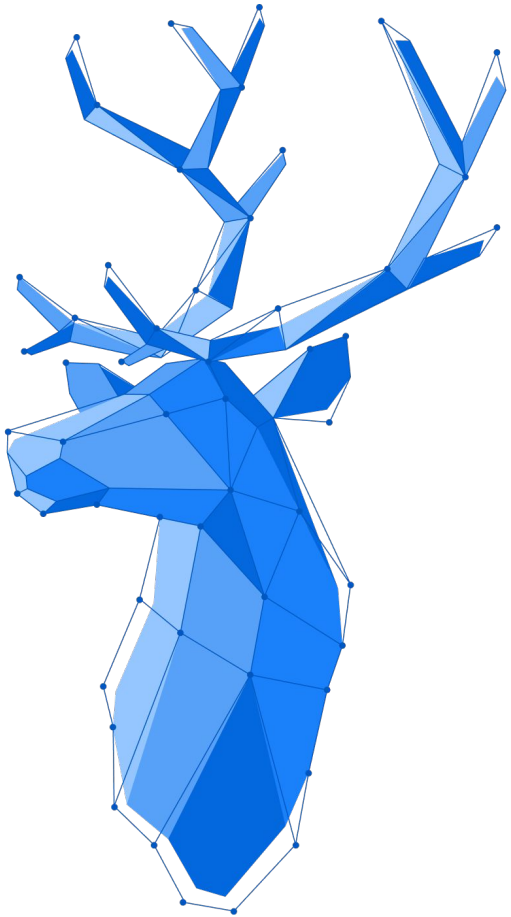
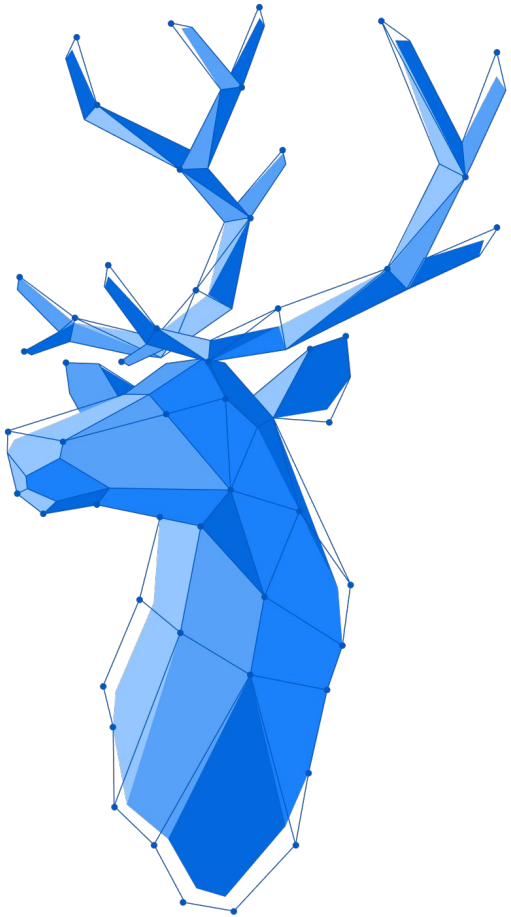
Create PWA with Vaadin

Just by adding @PWA, a Vaadin application becomes a PWA

```
@Route
@PWA(name = "My Progressive Web Application", shortName = "MyPWA")
public class MainView extends VerticalLayout {
   public MainView() {
       Button button = new Button("Click me",
               e -> Notification.show("Hello Worlld!"));
       add(button);
   }
}
```

Summary

- What is Vaadin
- Java
- Components
- Tools
- Progressive Web Application



Thank you!