Introduction

Vaadin platform

Agenda

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

What is Vaadin?

Vaadin

Finnish world

Means female **reindeer**

What is Vaadin?

Vaadin is a web framework for **Java**. It comes with a suite of **components** and **tools** for building **Progressive Web Apps** your users will ❤️

How is Vaadin different?

The only framework focused on building **Progressive Web Apps** on the **Java** platform

Typesafe end-to-end integration gives unparalleled developer experience.

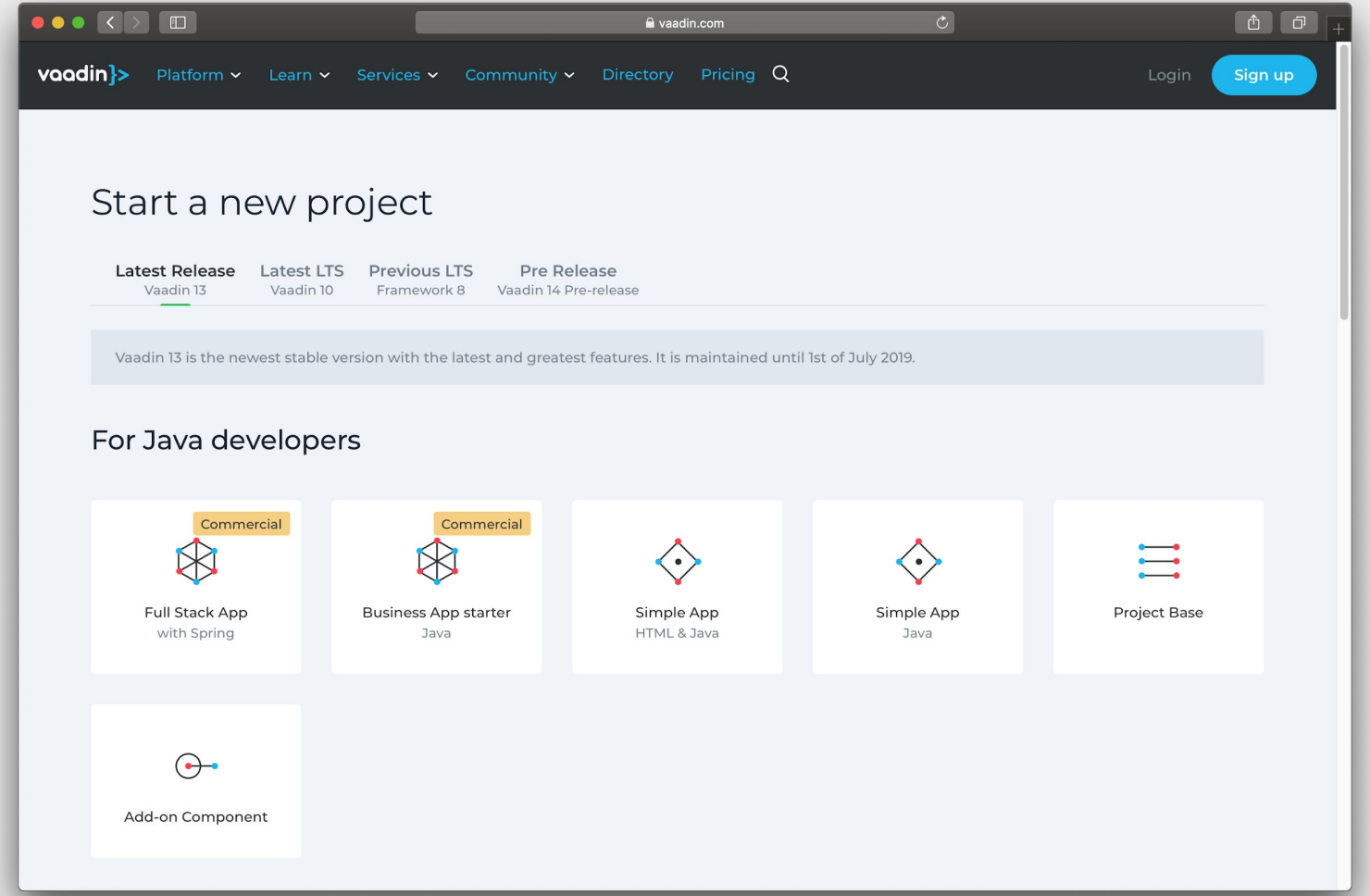
Vendor-backed open source is a **stable** foundation to build on.

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

Java

vaadin}>

https://vaadin.com/start



Project Structure

src/main/java for Java sources

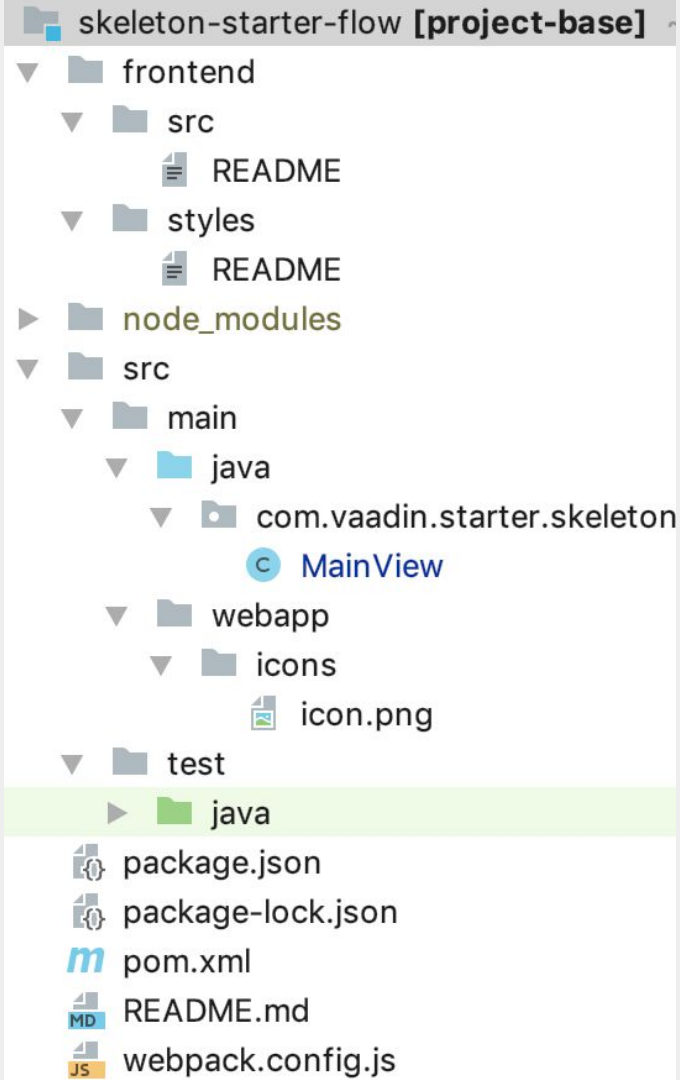
src/test/java for test sources

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

webpack.config.js auto generated file for webpack



Hello World!

```
@Route("")
public class MainView extends VerticalLayout {
   public MainView() {
       Button button = new Button("Click me",
               e -> Notification.show("Hello World!"));
       add(button);
   }
}
```

Hello World!

Integrate with backend service

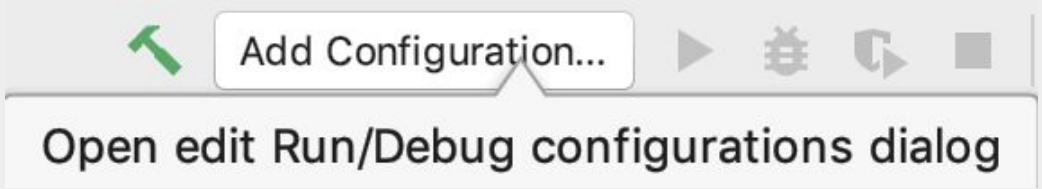
```
@Route("")
public class MainView extends VerticalLayout {
   private MessageBean messageBean = new MessageBean();

   public MainView() {
       Button button = new Button("Click me",
               e -> Notification.show(messageBean.getMessage()));
       add(button);
   }
}
```

How to run the application?

1. Run mvn jetty:run
2. Wait for the application to start
3. Open http://localhost:8080 to view the application

Debug/Run (IntelliJ)

Create a new Run/Debug configuration

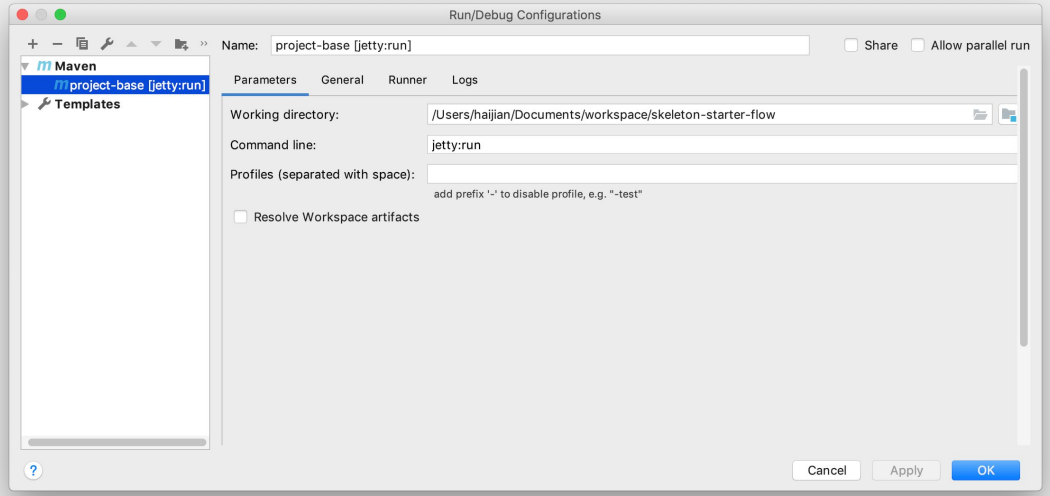


Debug/Run (IntelliJ)

Add a new Maven configuration

Specify jetty:run as the command line

Click OK



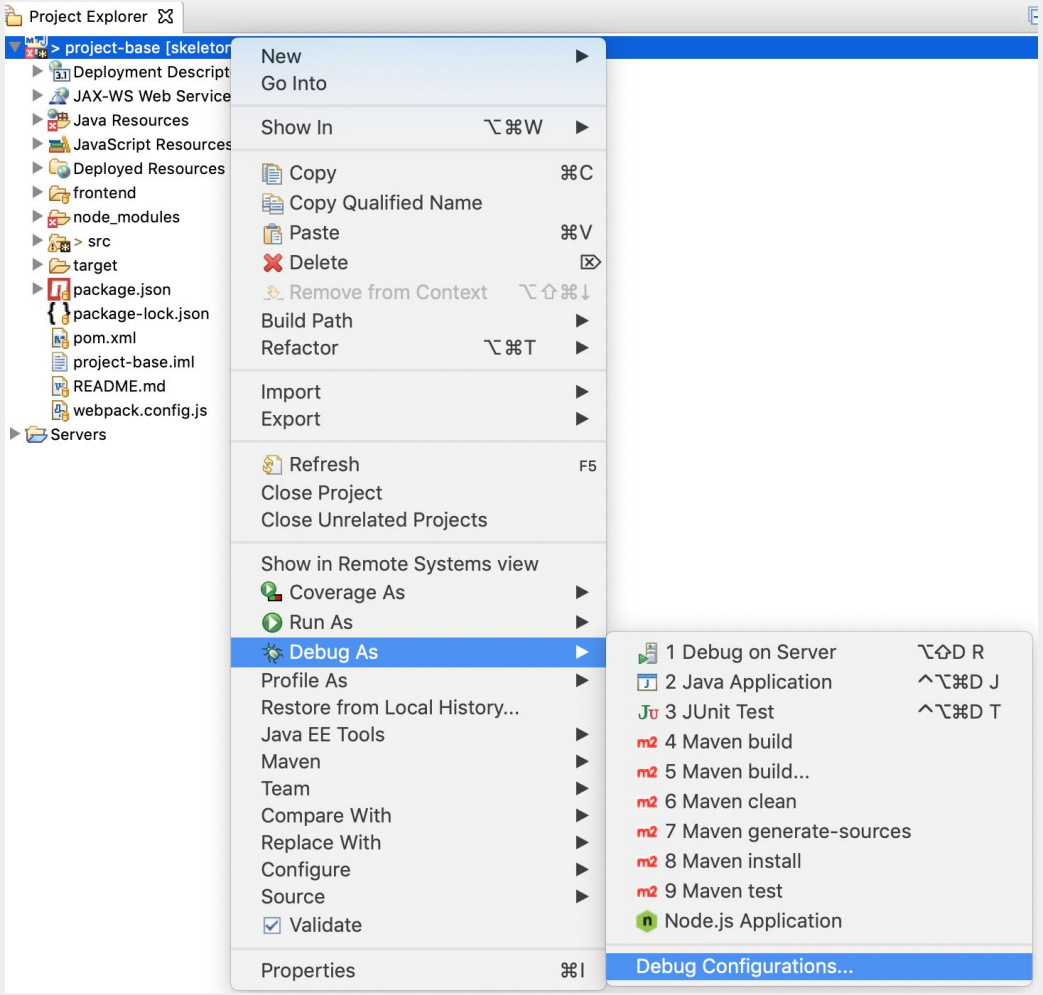
Debug/Run (IntelliJ)

Click the debug/run button in the toolbar

Debug/Run (Eclipse)

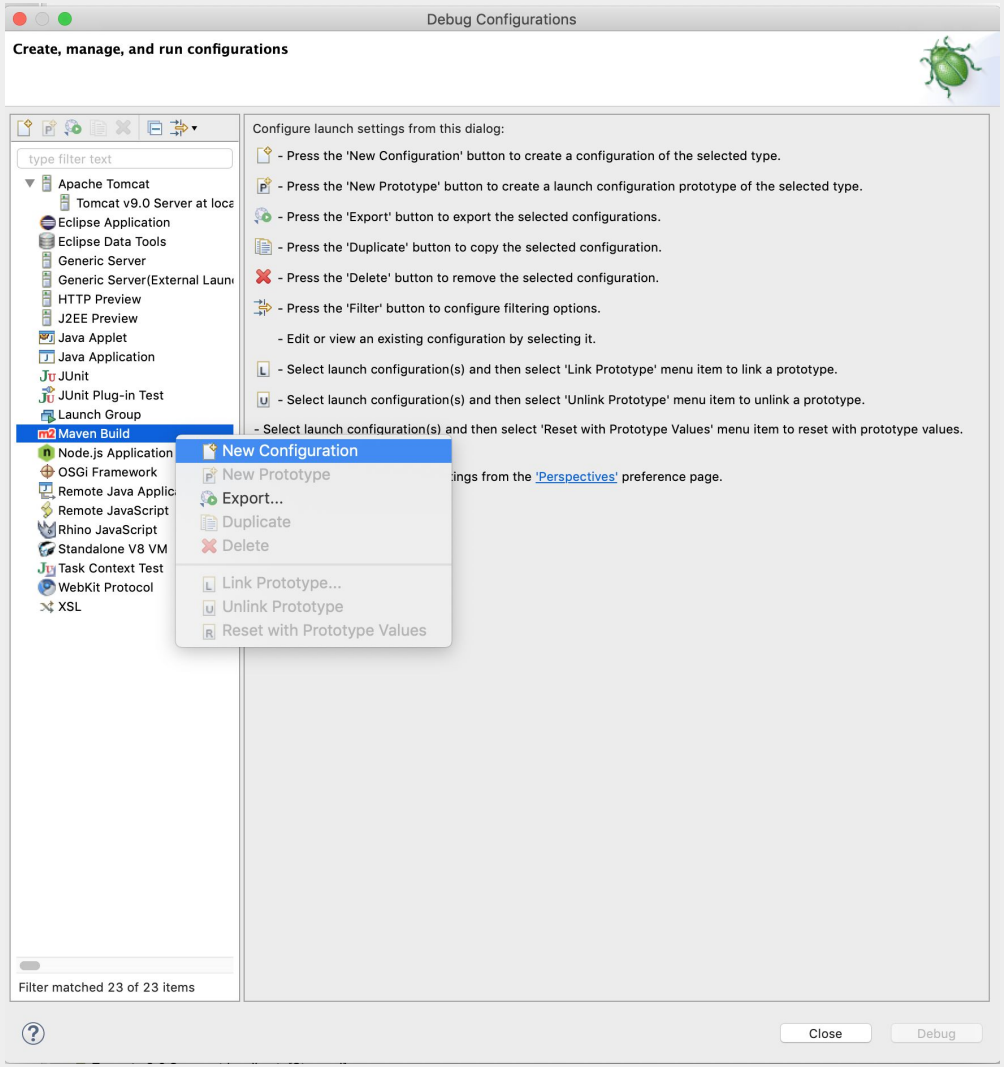
Right click on the project

Debug As -> Debug Configurtion...



Debug/Run (Eclipse)

In the popup window, right click
Maven Build -> New Configuration



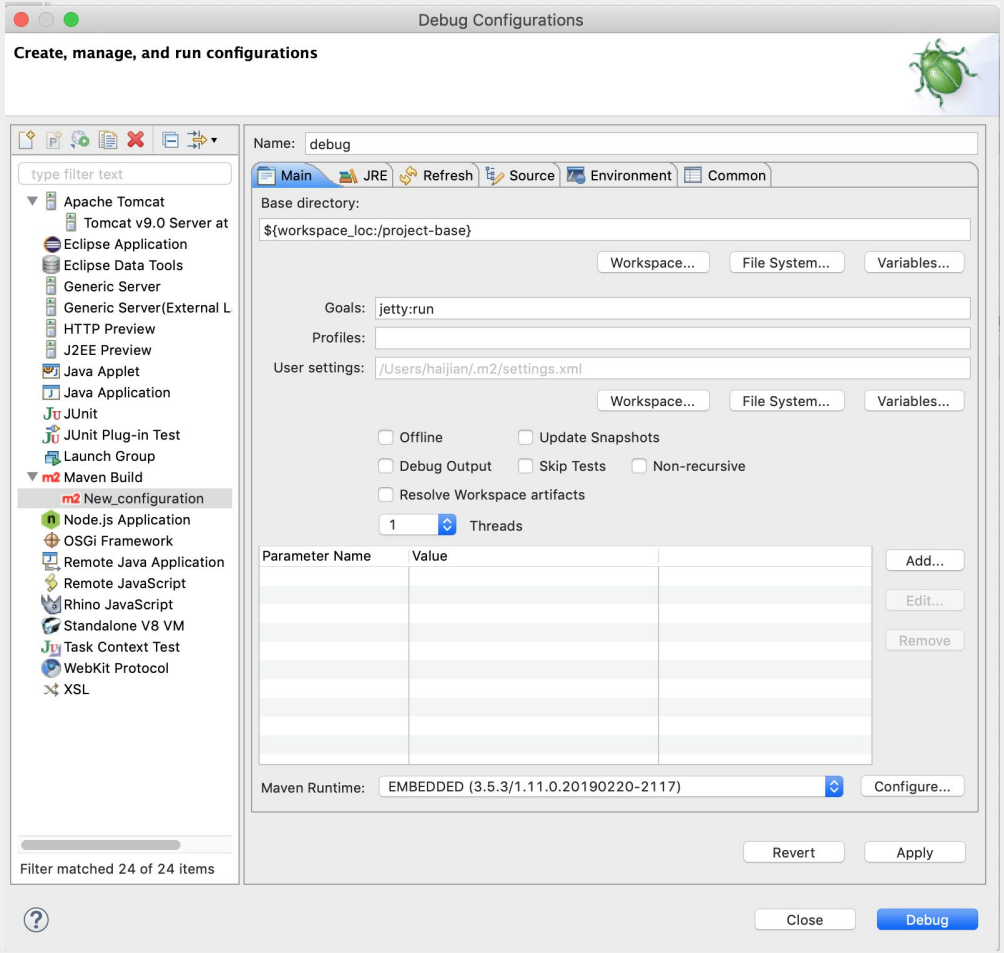
Debug/Run (Eclipse)

Give it a name

Set the Base directory as the project directory

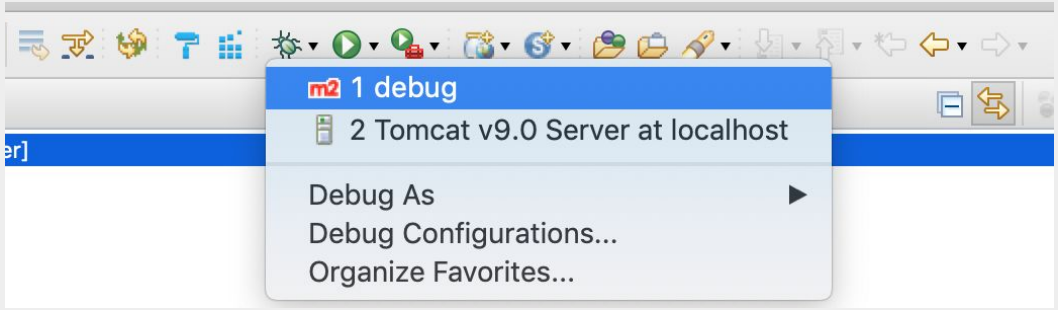
Set the goal as jetty:run

Click Apply



Debug/Run (Eclipse)

Click the debug button in the toolbar



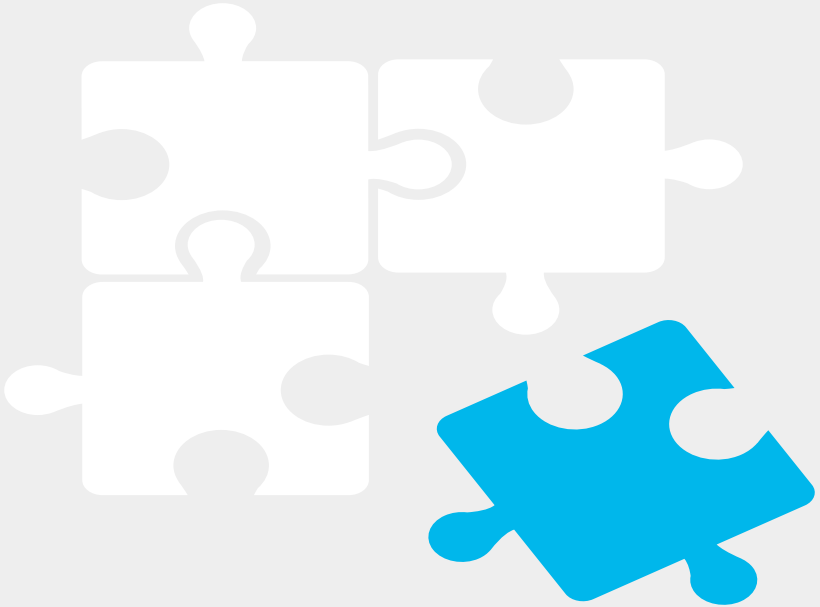
Components

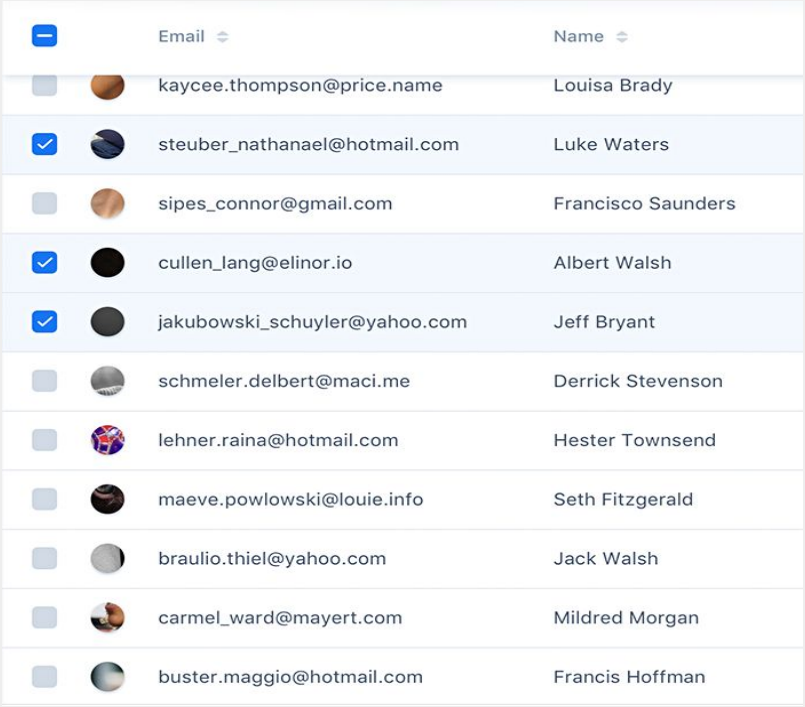
Components

Built-in Components

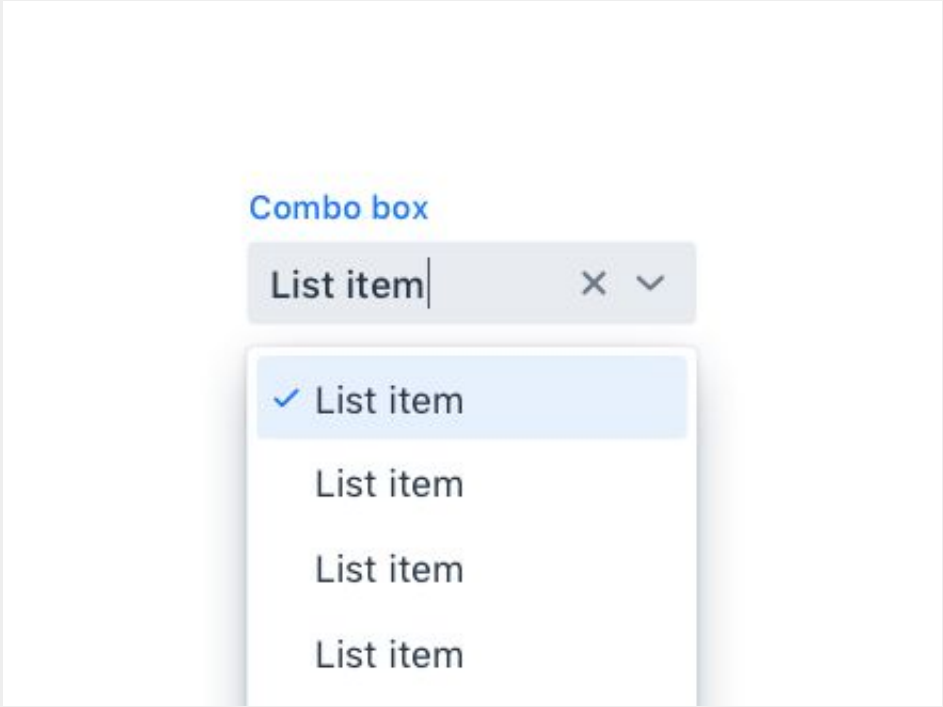
Architecture

Web Components

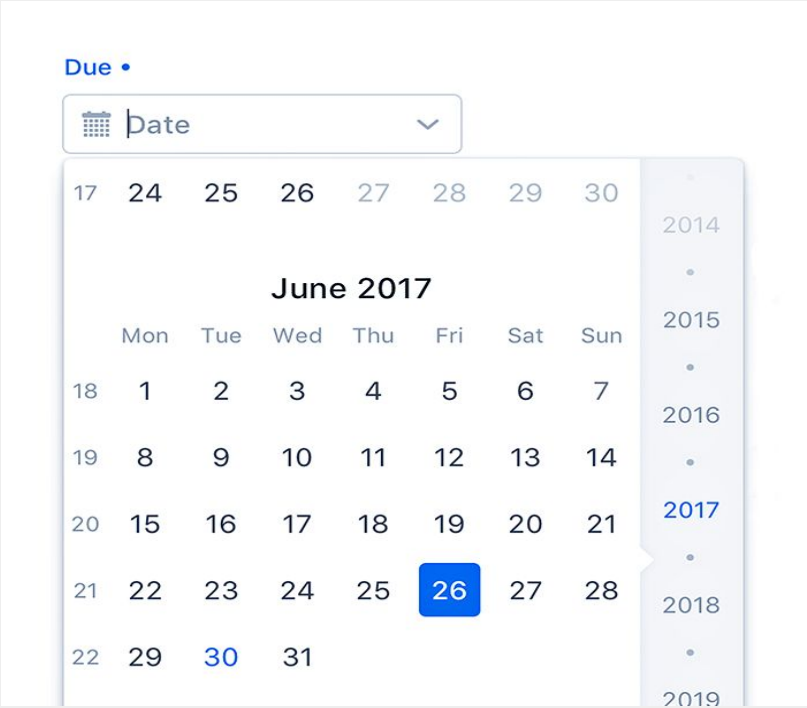




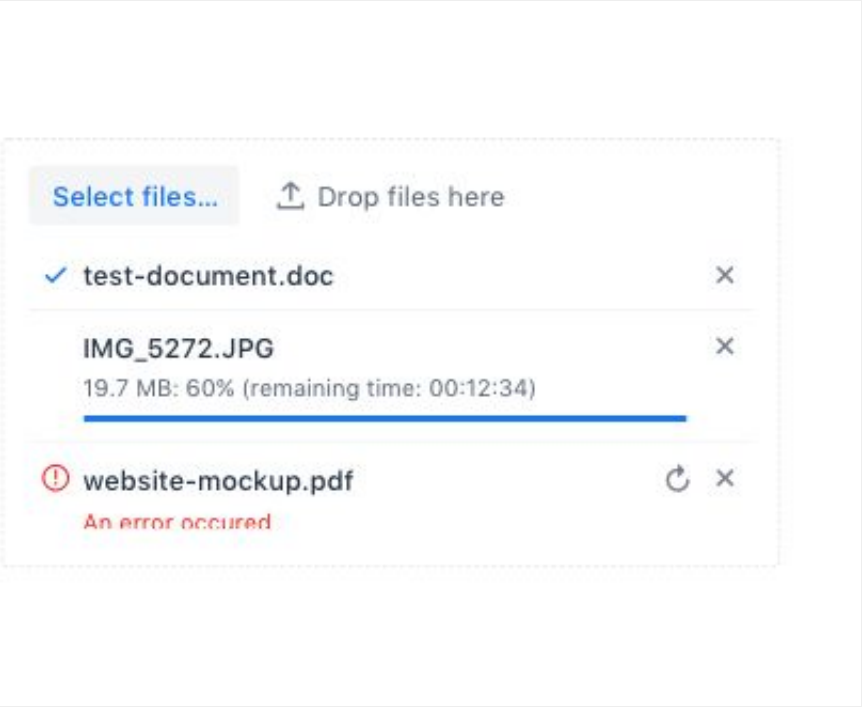
Grid



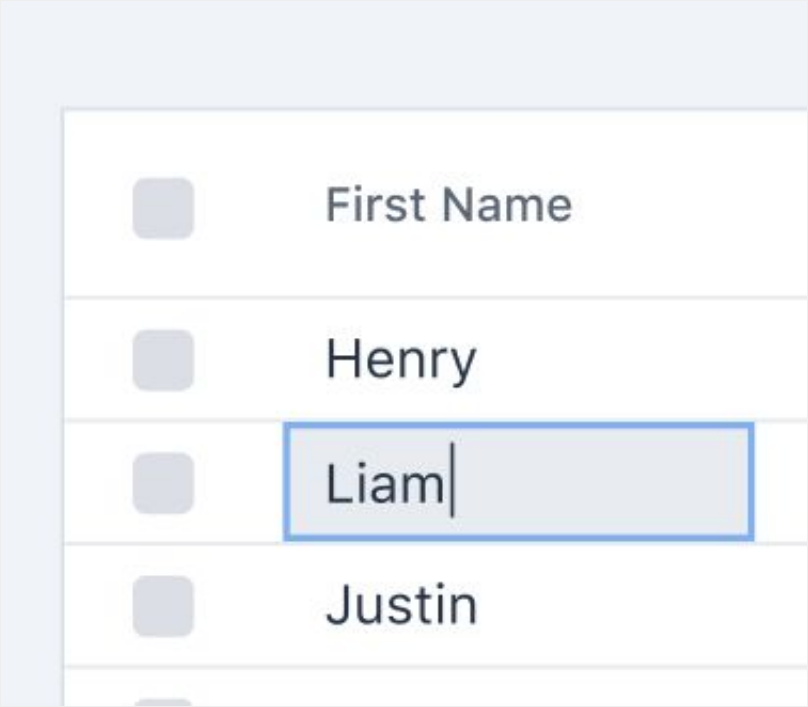
Combobox



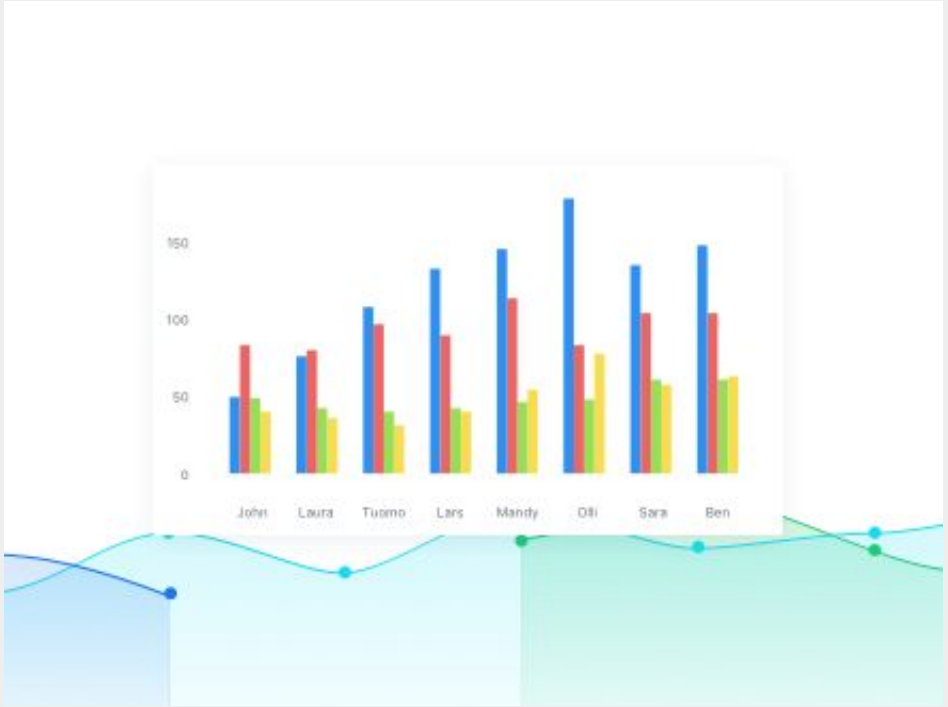
Date Picker



Upload



Grid Pro



Charts

Form Inputs

Checkbox
Combo Box
Custom Field
Date Picker Email Field
List Box
Number Field
Password Field
Radio Button
Select
Text Field
Time Picker
Upload

Visualization & Interaction

Accordion Button
Context Menu
Details
Dialog
Grid
Icons
Item
Notification
Progress Bar
Tabs
TreeGrid

Layouts

Horizontal Layout
Vertical Layout
Ordered Layout
Form Layout

Split Layout
App Layout
Login

Pro Components

Board
CRUD
Charts
Confirm Dialog
Cookie Consent
Grid Pro
Rich Text Editor Spreadsheet

HTML Components

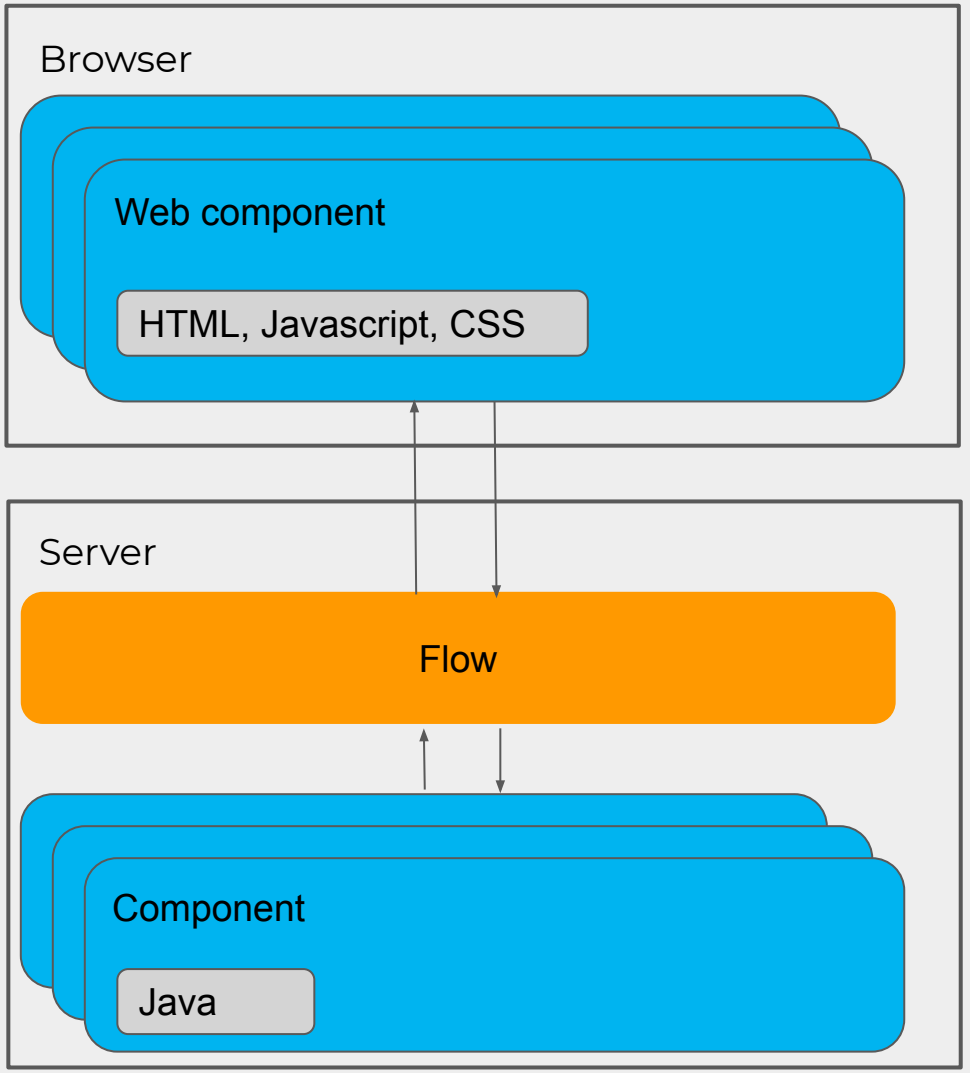
- Anchor
- Article
- Aside
- Description
- List
- Div
- Emphasis
- H1
- H2
- H3
- H4
- H5
- H6
- Hr
- Header
- Footer
- IFrame
- Input
- Label
- ListItem
- Main
- NativeButton
- Nav
- OrderedList
- Paragraph
- Section
- Span
- UnorderedList

Architecture

Client side: Web Components

Server side: Java

Communication: Flow



Web Components

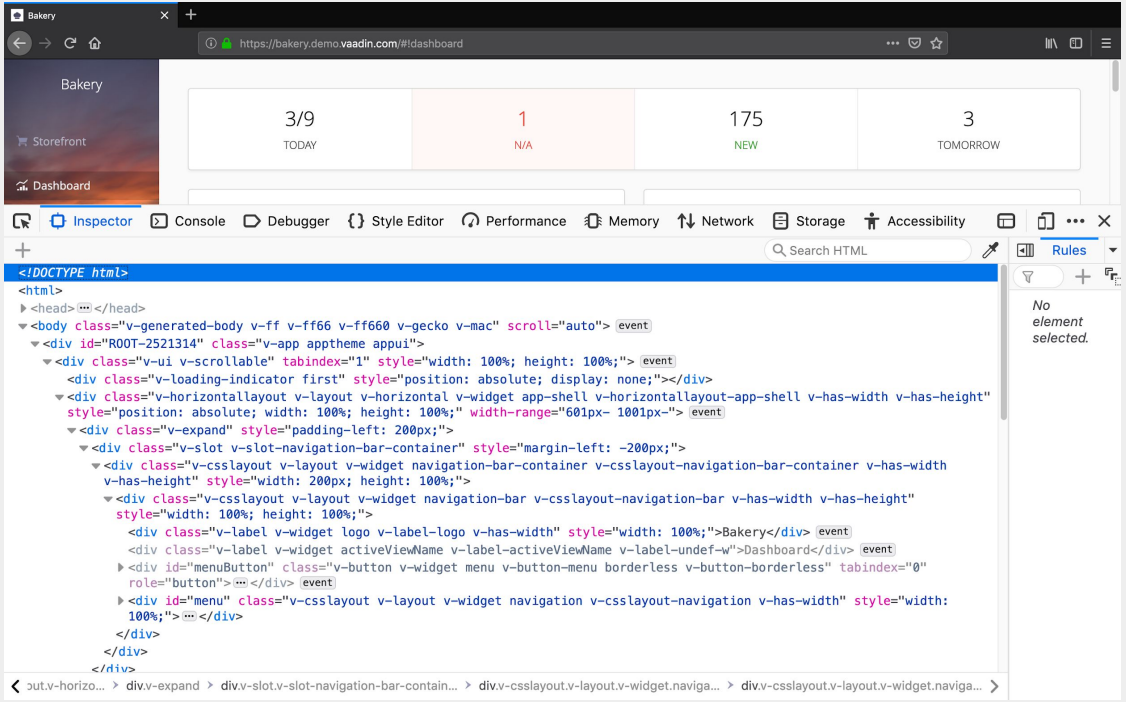
Based on web standards

Allows to create new tags

Semantics

New tags for your browser

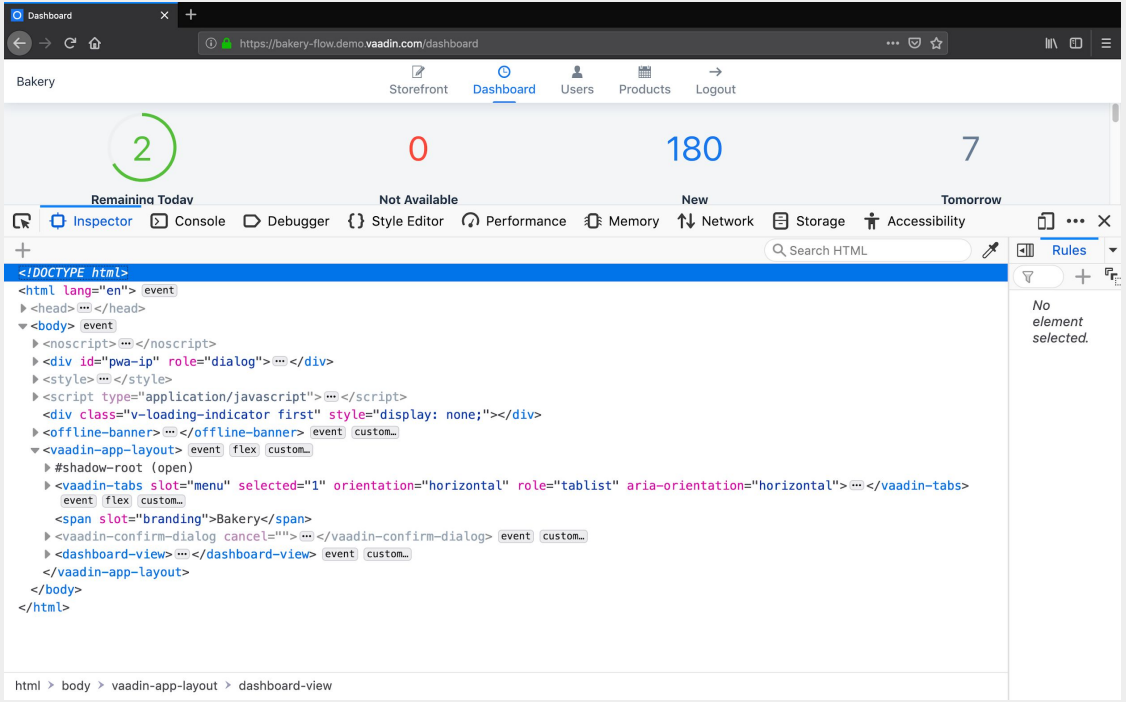
Component model for apps



Semantics

New tags for your browser

Component model for apps



True CSS Scoping

Shadow DOM

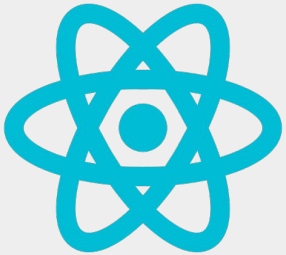
Current Solutions:

- Naming conventions: BEM, SMACSS
- Framework solutions: Angular, Vue
- CSS modules, CSS-in-JS

Reusability

Based on properties and DOM events

Framework agnostic



LitElement

Stability

Based on standards

Web standards live long

Use external components

With npm

Browse elements

2037 Elements

multiselect-combo-box

A multi select combo box web component based on Polymer 2.x and the vaadin-combo-box

★ 11 2

chemical-element-visualisation

A Web Component for visualizing an chemical element

★ 8 2

actor-layout

★ 1 0

a-wc-router

Routing Web Component

★ 4 0

flip-card

A Lit-Element component based on code-it notes by Dan Harding
https://www.instagram.com/same_dev_different_day/?hl=en

★ 0 0

paper-pagination

★ 0 0

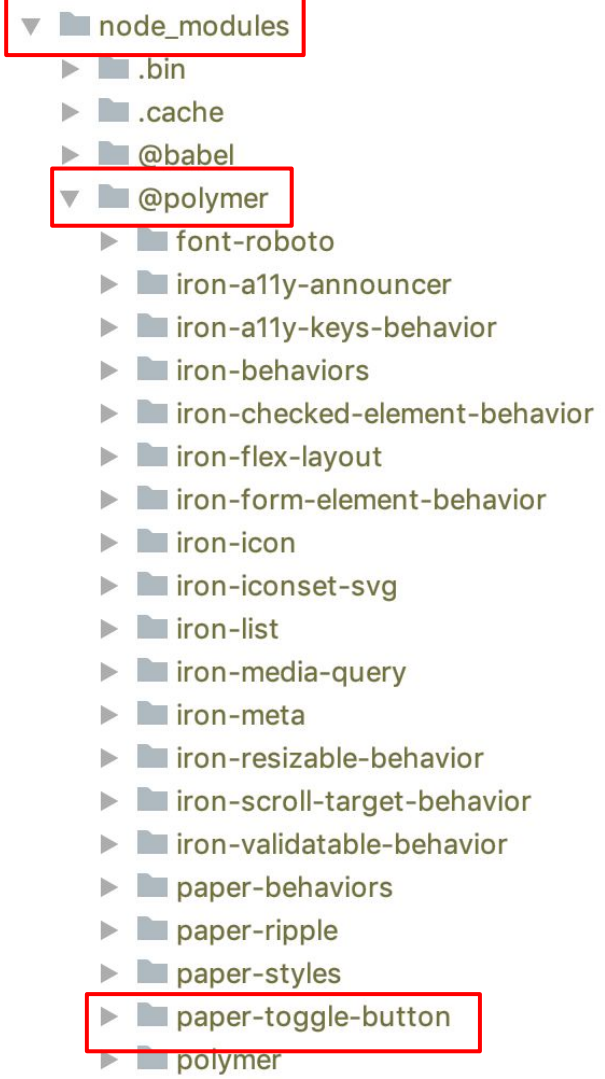
COPY

```
<paper-toggle-button>Toggle</paper-toggle-button>
<paper-toggle-button checked>Toggle</paper-toggle-button>
<paper-toggle-button disabled>Disabled</paper-toggle-button
<paper-toggle-button invalid>Invalid</paper-toggle-button>
```

Install

Run npm install on the project's root directory

```
npm install --save @polymer/paper-toggle-button
```



Connect to Java

```
@Tag("paper-toggle-button")
@JsModule("@polymer/paper-toggle-button.js")
public class PaperToggleButton extends Component {
}
```

Use as if it were built-in

```
@Route
public class MainView extends VerticalLayout {

   public MainView() {
       add(new PaperToggleButton());
   }

}
```

Exercise

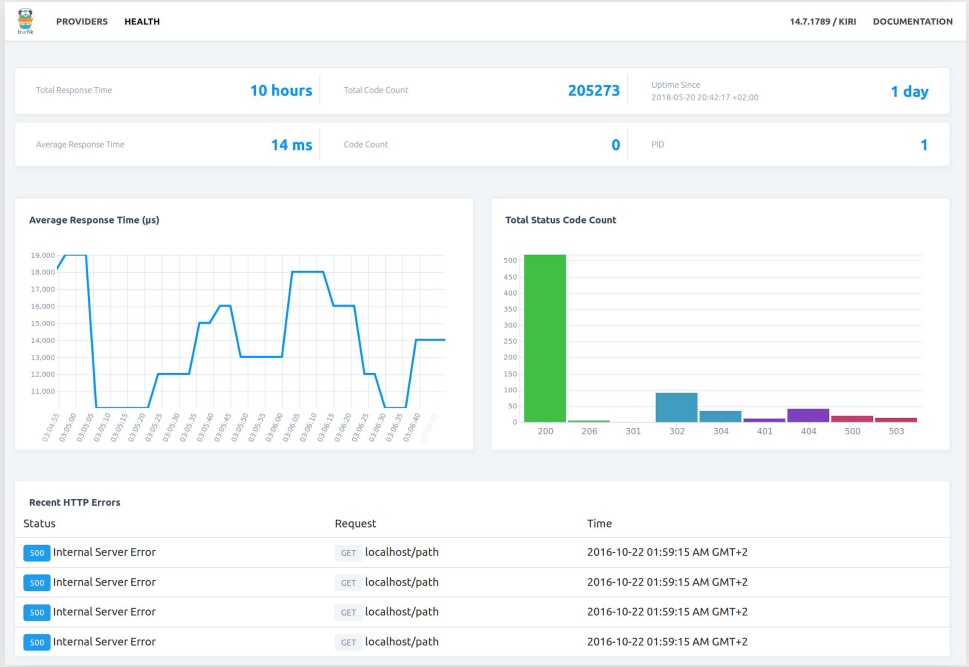
Tools

Tools

TestBench

Designer

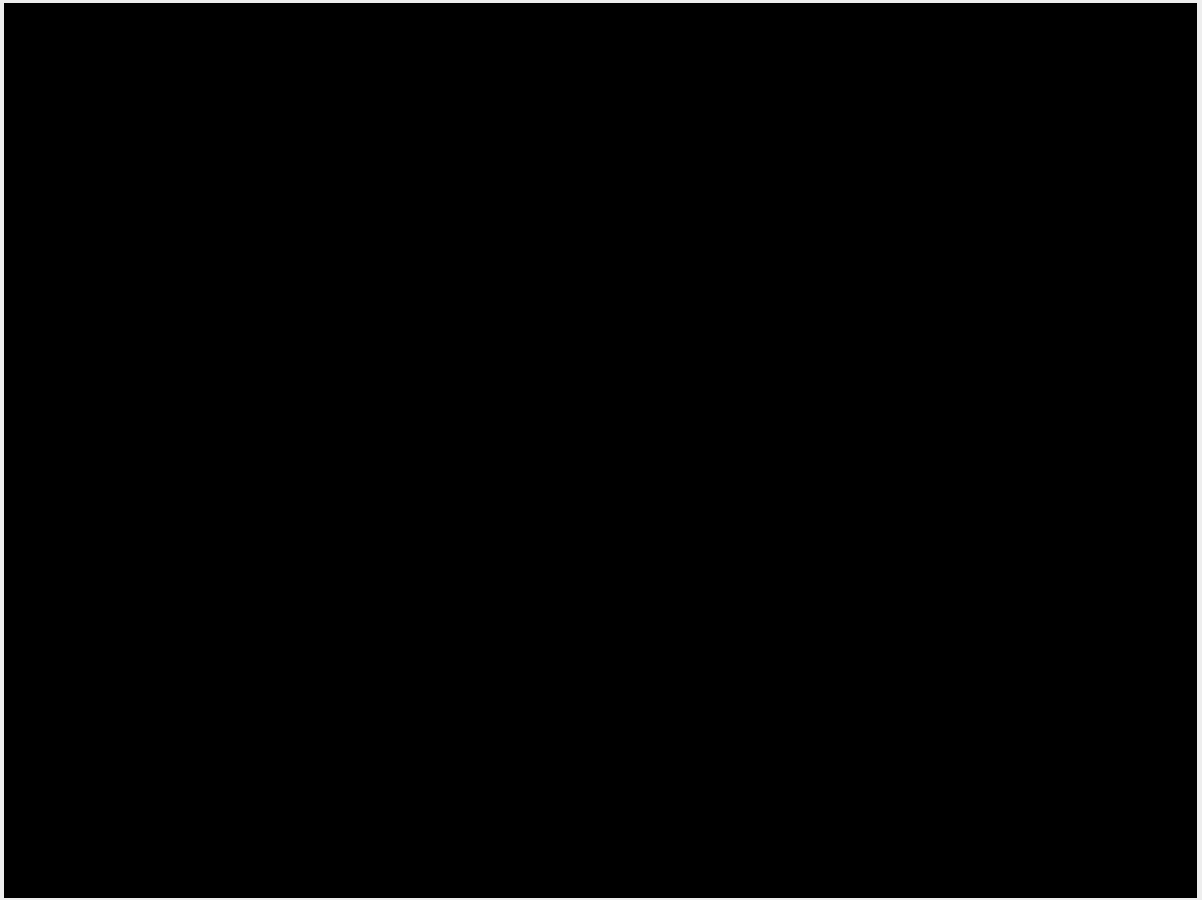
Multiplatform Runtime



What is TestBench?

Vaadin TestBench is a tool for creating and running browser-based integration tests for your Vaadin application

How does TestBench work?



Testbench vs Selenium

API for Vaadin Components

Testbench

```
//selects the row with index 5 in a grid
grid.select(5);
```

Selenium

```
xpath=//div[@id='list']/div[3]/table/tbody/tr[2]/td
```

vaadin}>

Server roundtrip

Testbench waits until the server-side is ready.

Selenium has to wait manually

vaadin}>

Shadow DOM

Vaadin TestBench has support for piercing Shadow DOM.

vaadin}>

Screen Comparison

When screen comparison fails, the differences are highlighted.

How does TestBench work?

Simulates a user of your application

performs the tasks specified using **Java code**

Verifies that the expected actions take place in the application.

The code

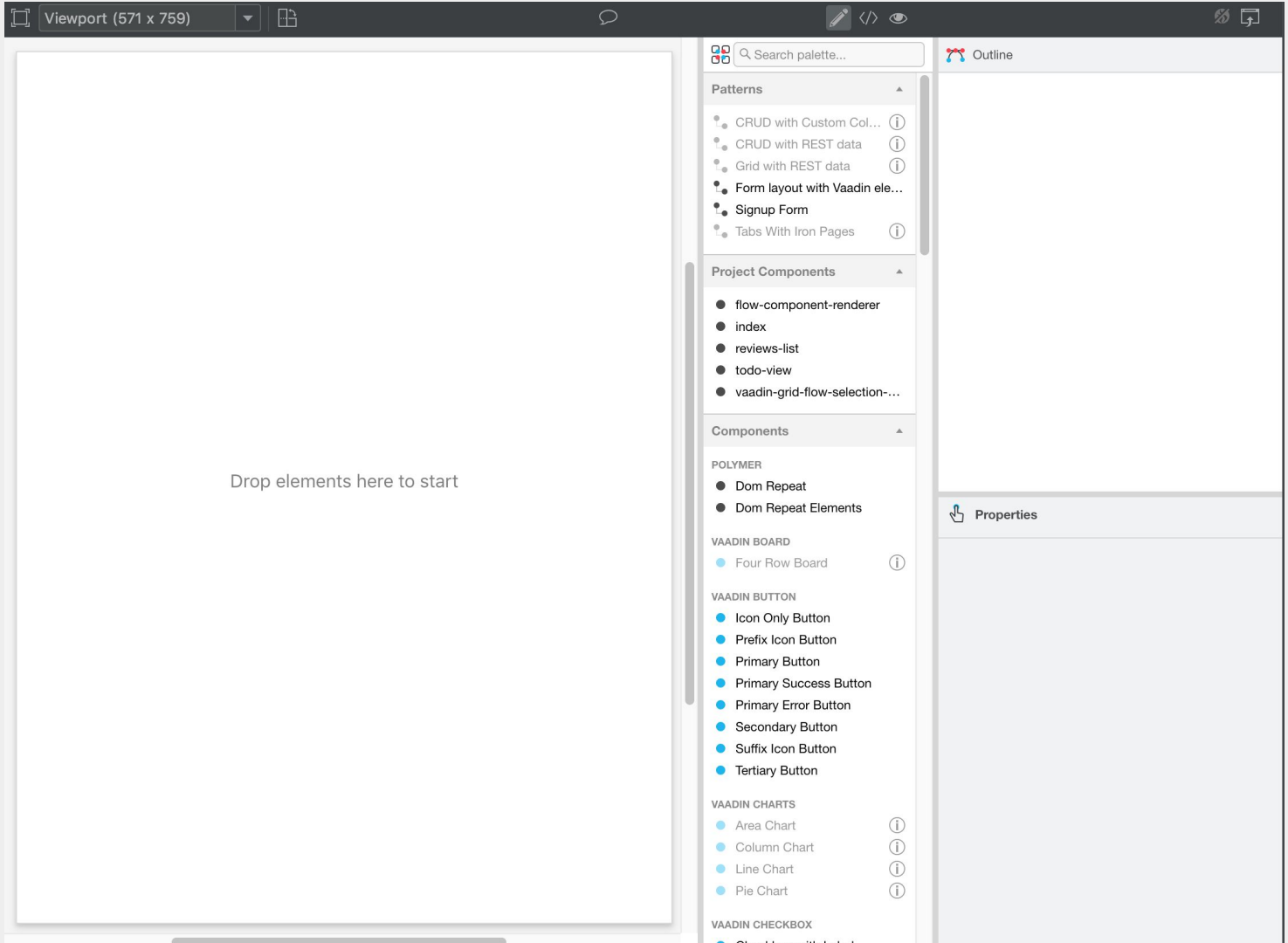
```
public class SimpleIT extends TestBenchTestCase {

    ...

    @Test
    public void clickButton() {
        // Find the first button (<vaadin-button>) on the page
        ButtonElement button = $(ButtonElement.class).first();

        // Click it
        button.click();

        Assert.assertTrue($(NotificationElement.class).exists());
        Assert.assertEquals("Clicked", $(NotificationElement.class).first().getText());
    }


}
```

What is Vaadin Designer?

Designer is an IDE **plugin**

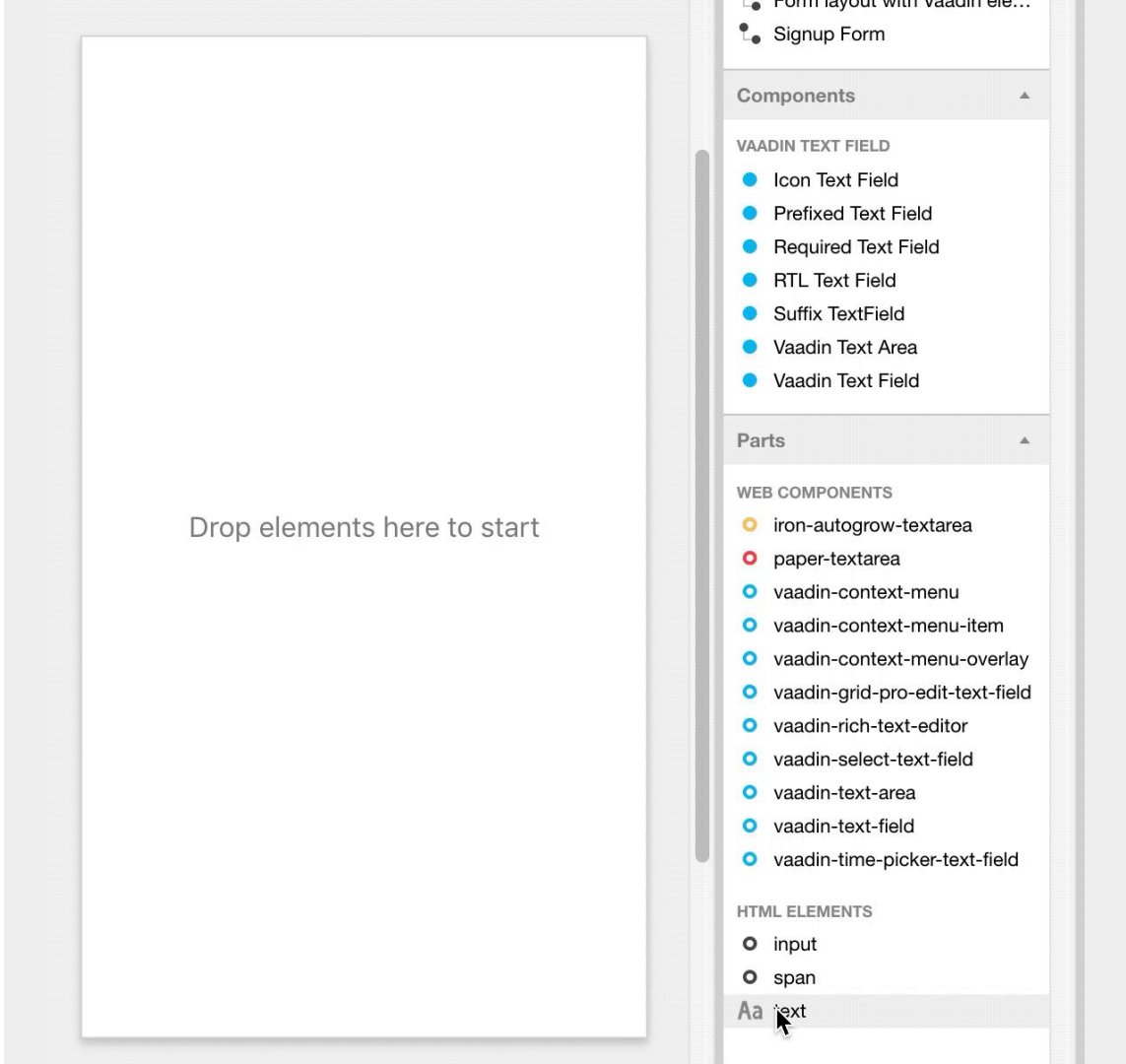
A **visual tool** for building Vaadin applications with **Drag&Drop**

vaadin}>


Viewport (571 x 759)
Drop elements here to start
Search palette...
Patterns
CRUD with Custom Col...
CRUD with REST data
Grid with REST data
Form layout with Vaadin ele...
Signup Form
Tabs With Iron Pages
Project Components
flow-component-renderer
index
reviews-list
todo-view
vaadin-grid-flow-selection-...
Components
POLYMER
Dom Repeat
Dom Repeat Elements
VAADIN BOARD
Four Row Board
VAADIN BUTTON
Icon Only Button
Prefix Icon Button
Primary Button
Primary Success Button
Primary Error Button
Secondary Button
Suffix Icon Button
Tertiary Button
VAADIN CHARTS
Area Chart
Column Chart
Line Chart
Pie Chart
VAADIN CHECKBOX
Outline
Properties

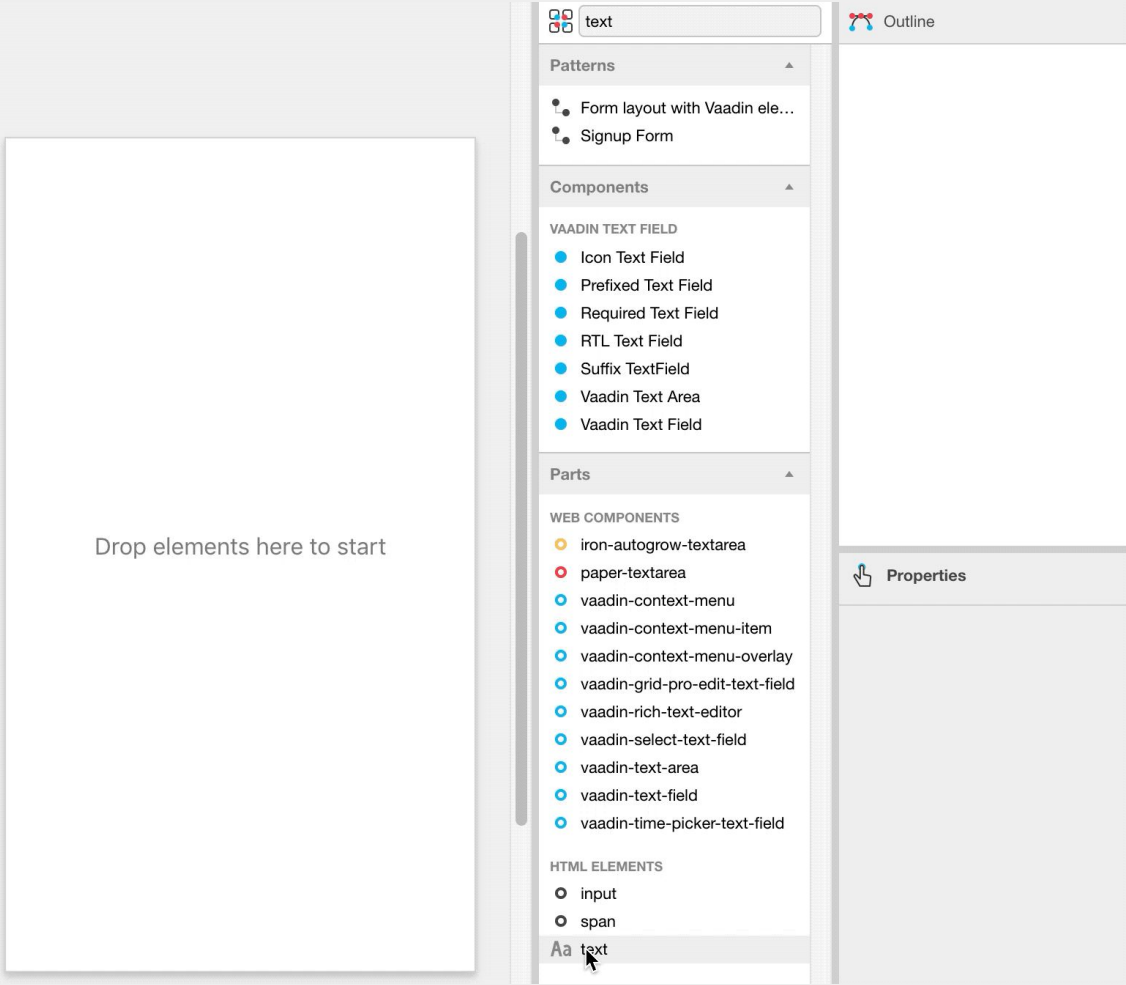
Add Components

Drop to canvas



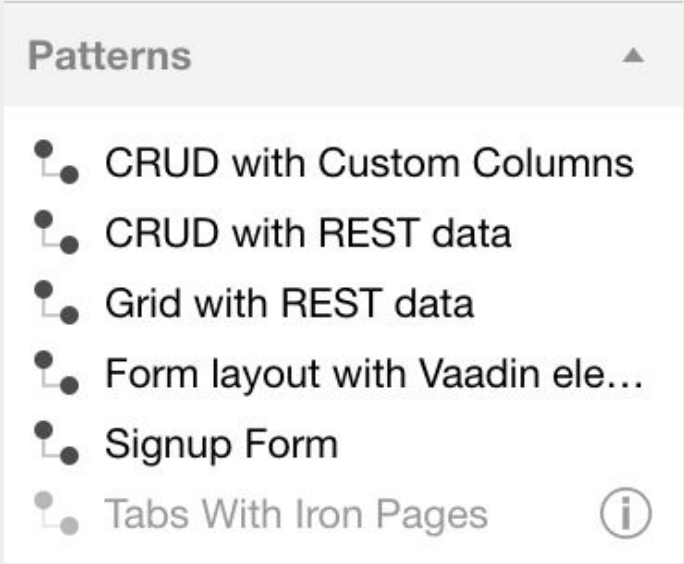
Add Components

Drop to the Outline view



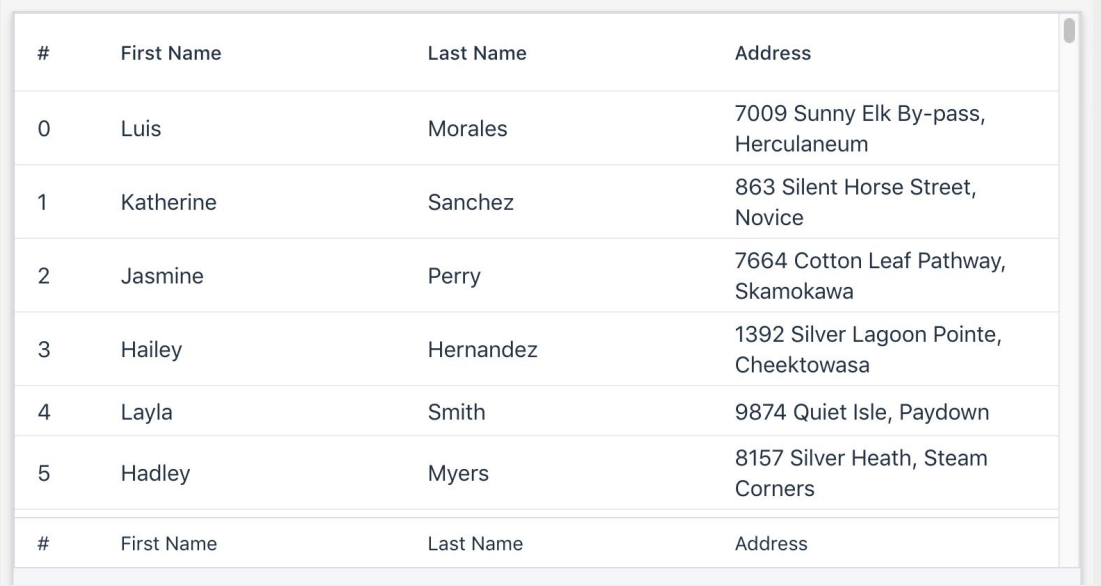
Patterns

Quick-start solutions to certain design tasks



Patterns

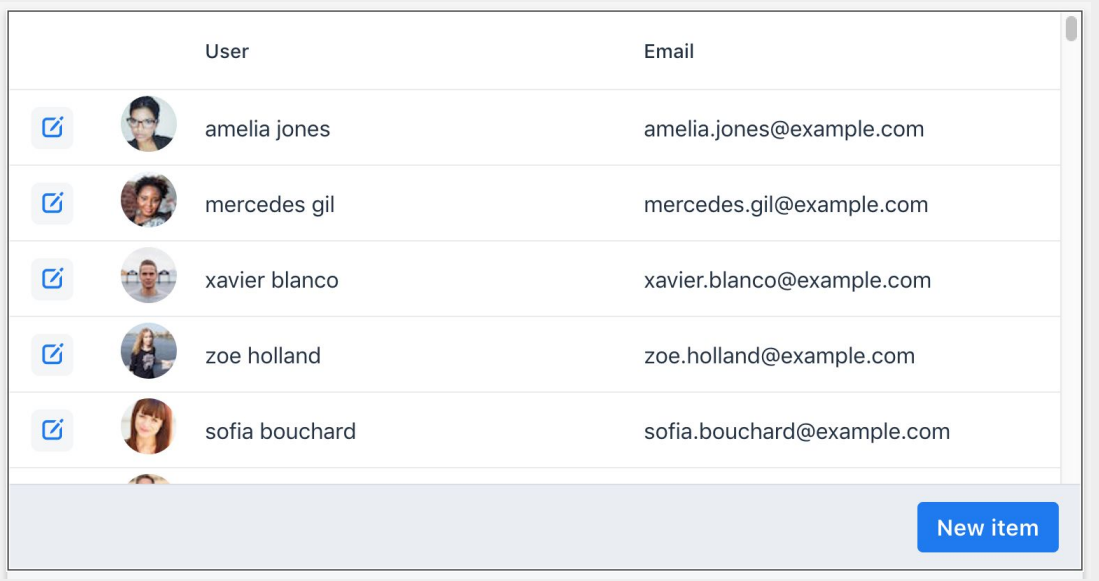
Grid with REST data



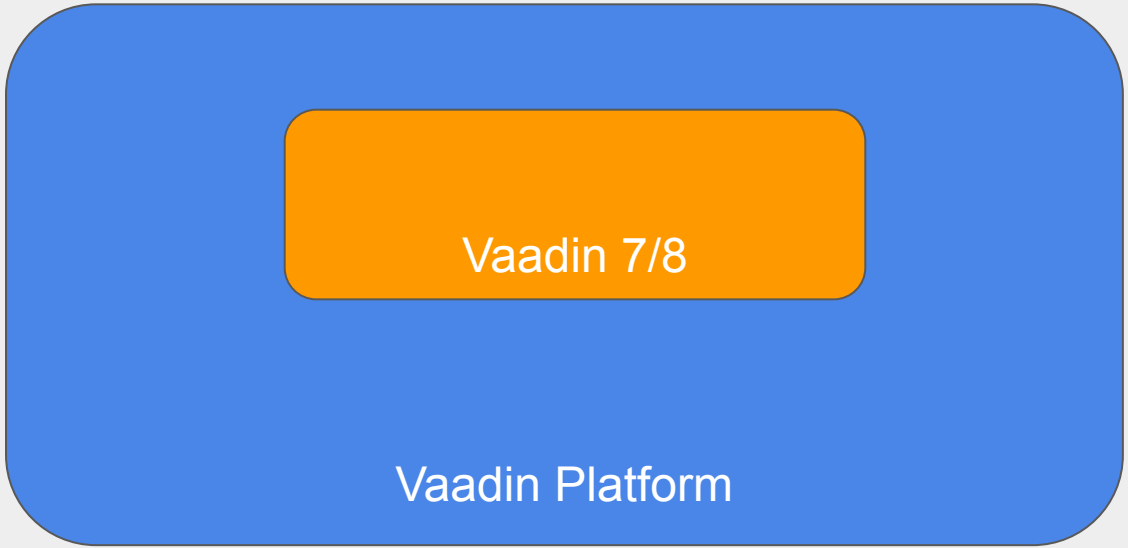
#	First Name	Last Name	Address
0	Luis	Morales	7009 Sunny Elk By-pass, Herculaneum
1	Katherine	Sanchez	863 Silent Horse Street, Novice
2	Jasmine	Perry	7664 Cotton Leaf Pathway, Skamokawa
3	Hailey	Hernandez	1392 Silver Lagoon Pointe, Cheektowasa
4	Layla	Smith	9874 Quiet Isle, Paydown
5	Hadley	Myers	8157 Silver Heath, Steam Corners
#	First Name	Last Name	Address

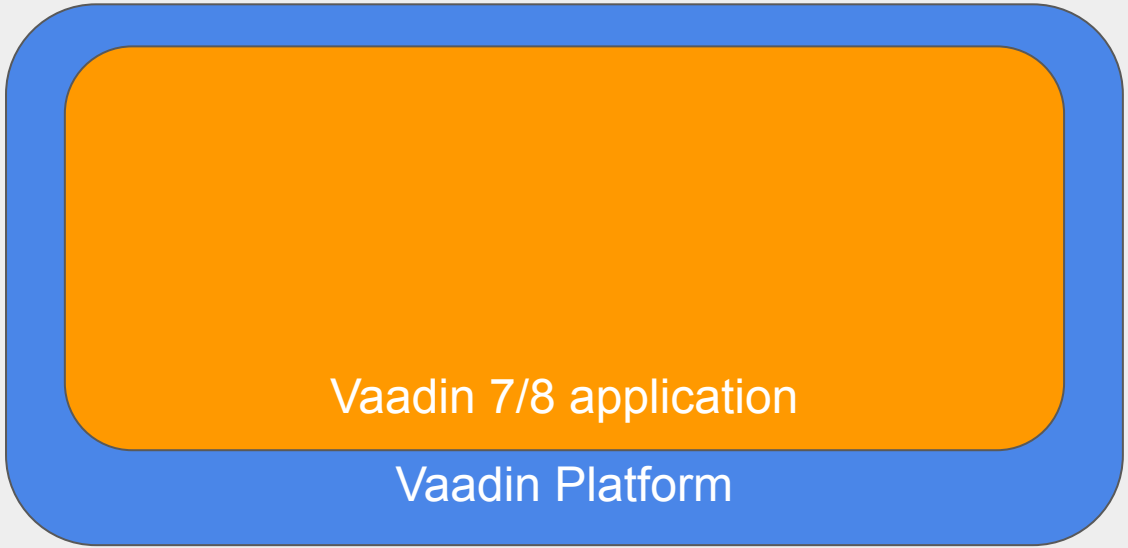
Patterns

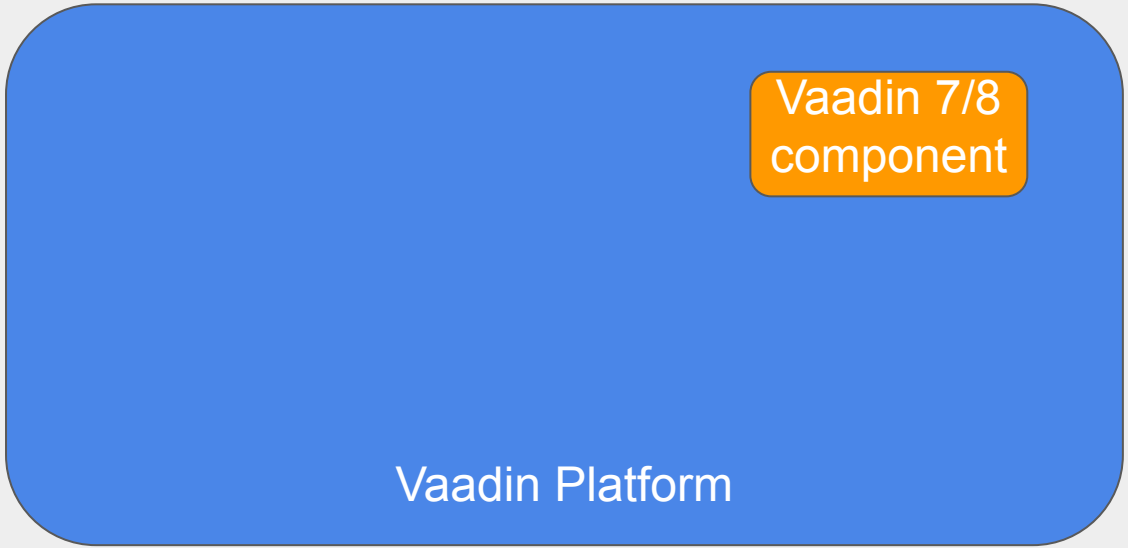
CRUD with custom columns



Multiplatform Runtime


Vaadin 7/8
Vaadin Platform


Vaadin 7/8 application
Vaadin Platform


Vaadin 7/8
component
Vaadin Platform

How does it work?

Cannot use legacy components directly in the new application

```
Button legacyButton = new Button("Legacy button");
VerticalLayout flowLayout = new VerticalLayout();
flowLayout.add(legacyButton);
```

How does it work?

Wrap legacy component with a legacy wrapper

```
Button legacyButton = new Button("Legacy button");
VerticalLayout flowLayout = new VerticalLayout();
flowLayout.add(new LegacyWrapper(legacyButton));
```

Progressive Web Apps

Web apps with native-like experiences

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

vaadin}>

Feedback

bit.ly/vaadin-training