Vaadin Workshop

Simon Martinelli
@simas_ch
martinelli.ch

About Me

1972

1995

COBOL

2000

2007

Bern University of Applied Sciences

2009

2011

JSR-352
Java Batch

2012

2013

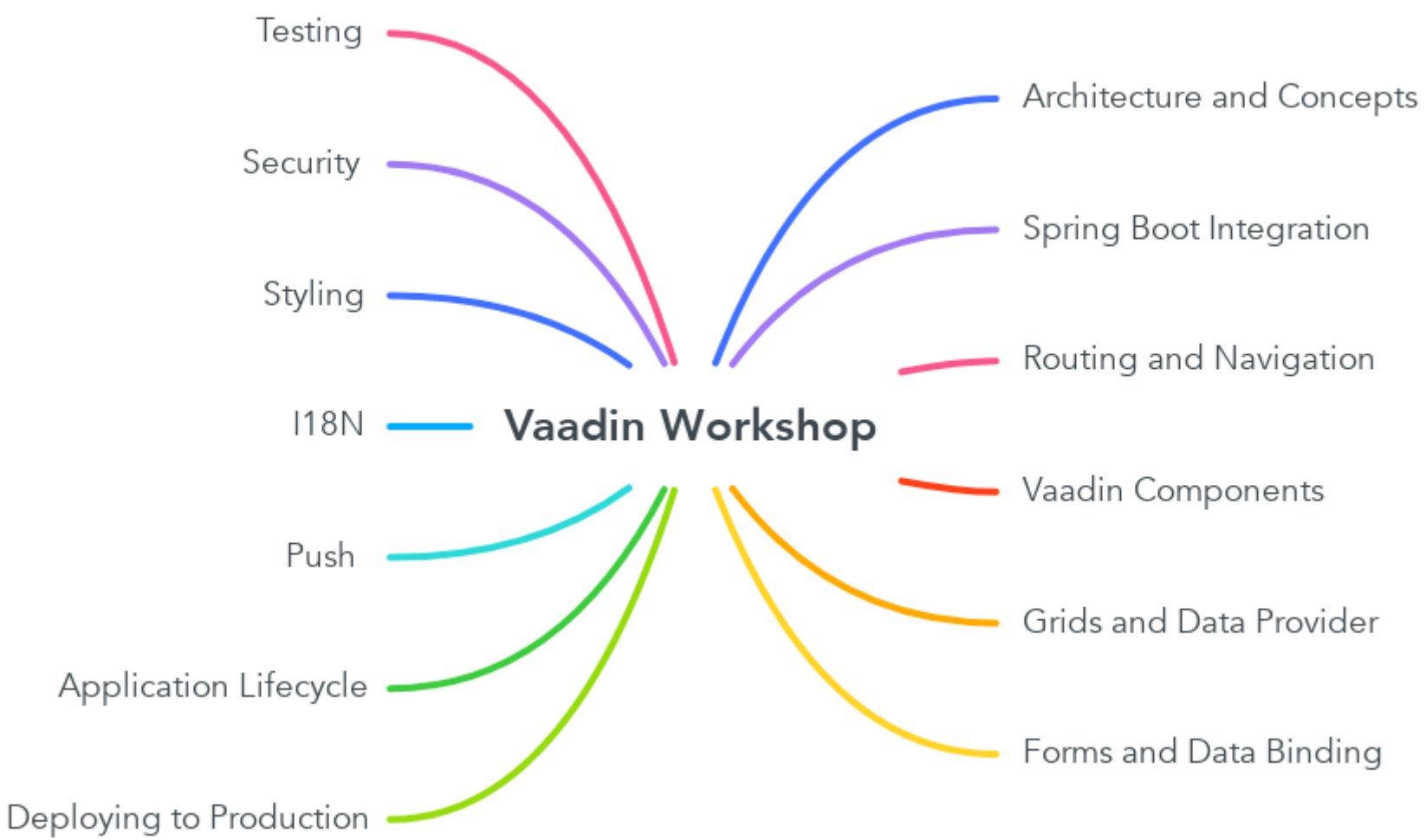
JSR-354
Money/Currency

2021

About You

- What’s your name?
- What’s your day job?
- Did you ever use Vaadin? If yes, which version?
- Do you have experience with Spring Boot?
- What are your expectations?

Workshop Topics

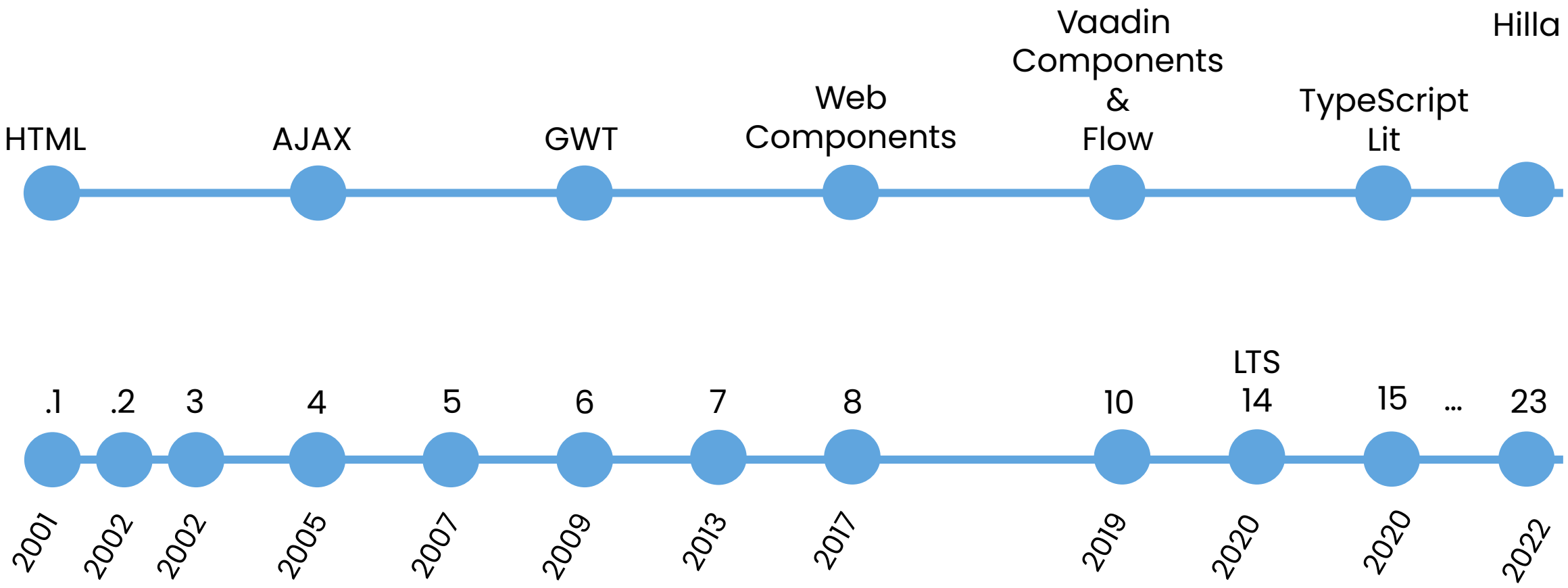



vaadin

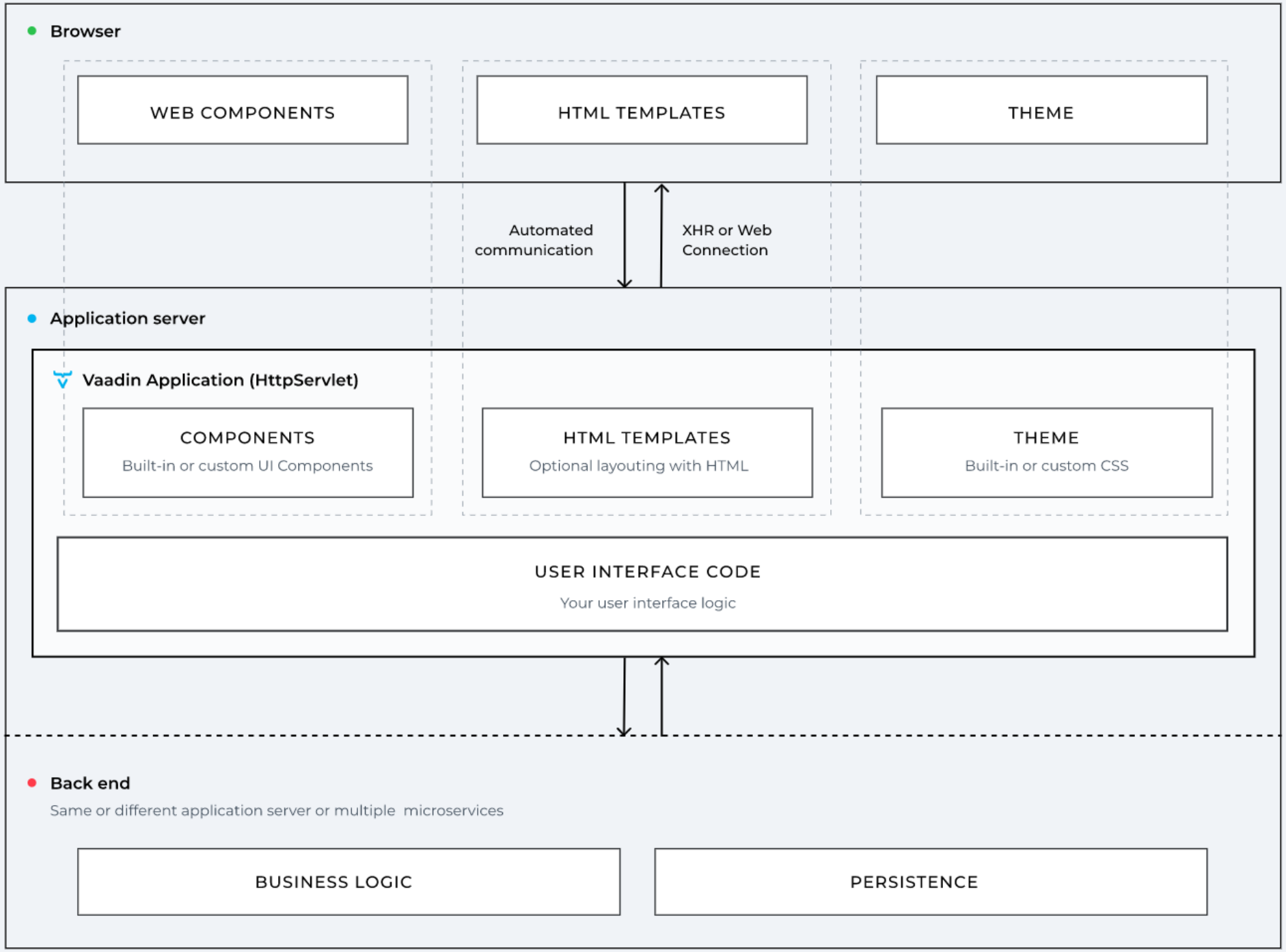
Helpful Resources

- https://vaadin.com/
- https://discord.com/invite/vaadin
- https://stackoverflow.com/

History


HTML
AJAX
GWT
Web Components
Vaadin Components & Flow
TypeScript Lit
Hilla
.1
.2
3
4
5
6
7
8
10
LTS 14
15
...
23
2001
2002
2002
2005
2007
2009
2013
2017
2019
2020
2020
2022

Architecture and Concepts


Browser
WEB COMPONENTS
HTML TEMPLATES
THEME
Automated communication
XHR or Web Connection
Application server
Vaadin Application (HttpServlet)
COMPONENTS
Built-in or custom UI Components
HTML TEMPLATES
Optional layouting with HTML
THEME
Built-in or custom CSS
USER INTERFACE CODE
Your user interface logic
Back end
Same or different application server or multiple microservices
BUSINESS LOGIC
PERSISTENCE

Getting Started

Build

- Default is Maven

- Gradle
  - https://vaadin.com/docs/latest/guide/start/gradle

Integrations

- Spring/Spring Boot
- Quarkus
- CDI (Java EE)
- OSGi
- Portlet (commercial)

Spring Integration

- Scopes
  - https://vaadin.com/docs/latest/integrations/spring/scopes
- Routing and Dependency Injection
  - https://vaadin.com/docs/latest/integrations/spring/routing
- Configuration
  - https://vaadin.com/docs/latest/integrations/spring/configuration

How to Start?

- https://start.vaadin.com
- https://start.spring.io

Exercise: Getting Started

1. Register for a Vaadin account
2. Go to https://start.vaadin.com
3. Download the application
4. Unzip and import the project into your favorite IDE
5. Run the app
   - Either with the IDE or

```
./mvnw spring-boot:run
```

Example Application

https://github.com/simasch/vaadin-workshop

Workshop Manager

MENU SUCHE KONTAKT

CH Open
Source | Business | Community

Dienstag, 06. September 2022 (9:00 – 17:00 Uhr)

Titel	Thema	Referenten	Status
ATDD mit Spring Boot & Karate	Softwareentwicklung mit Open Source	Thorben Stangenberg	
IoT Embedded Programmierung mit Zephyr OS	Mobile und IoT	Thomas Amberg	CANCELLED
Securing your Microservices with Spring Security, OAuth 2 and OpenID Connect (OICD)	Softwareentwicklung mit Open Source	Patrick Baumgartner	
Java Full-Stack Entwicklung mit Vaadin	Softwareentwicklung mit Open Source	Simon Martinelli	
Ansible Basics	Open Source Systeme und Applikationen	Jérôme Witt	
Mob Programming: Erlebe Kollaborations- und Kreativitätstechniken hautnah	Methodik & Soft Skills	Danilo Biella Barbara Dravec	
Functional Programming Idioms and Practices	Softwareentwicklung mit Open Source	Venkat Subramaniam	AUSGEBUCHT!

Requirements

- Create an application to manage workshops and participants
- A workshop has many participants, and a participant can attend one workshop
- The list of workshops is public
- To manage workshops and participants, the user must log in

Routing and Navigation

https://vaadin.com/docs/latest/routing

Route

```
@Route("")
public class HelloWorldView extends Div {

    public HelloWorldView() {
        setText("Hello world");
    }
}
```

Navigation

- Programatically

```
UI.getCurrent().navigate(HelloWorldView.class);
```

- RouterLink

```
new RouterLink("Home", HomeView.class);
```

Exercise: Routing and Navigation

1. Create views according to the requirements
2. Programmatically navigate from one view to another view
3. Implement BeforeEnterObserver and display the current timestamp

4. Bonus: Try out Route Templates

Vaadin Components

https://vaadin.com/docs/latest/components

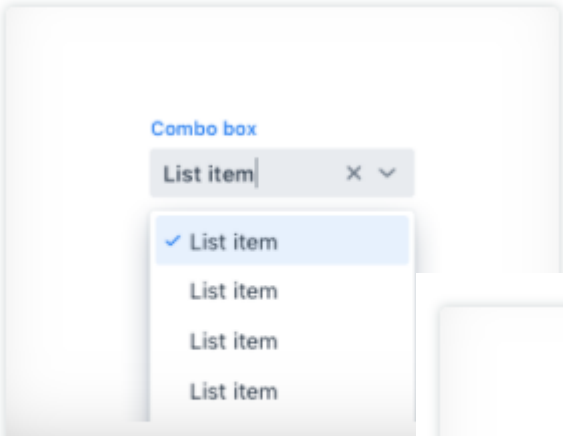
Exercise: Components

1. Get familiar with the Vaadin Components
2. Add one of the basic layouts (Horizontal or Vertical)
3. Try out
    1. Alignment
    2. Spacing
    3. Padding
    4. Expanding

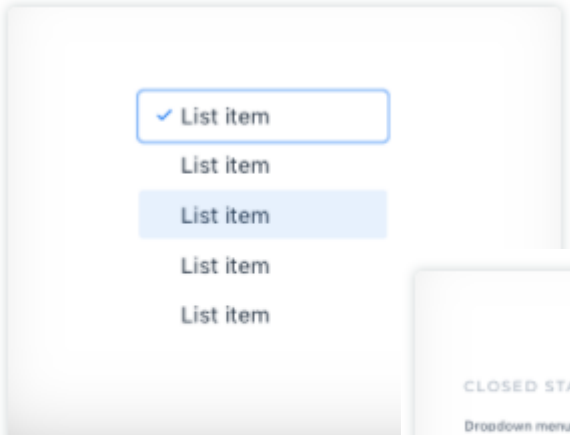
4. Bonus: Checkout the 3rd party components

Grids and Data Providers

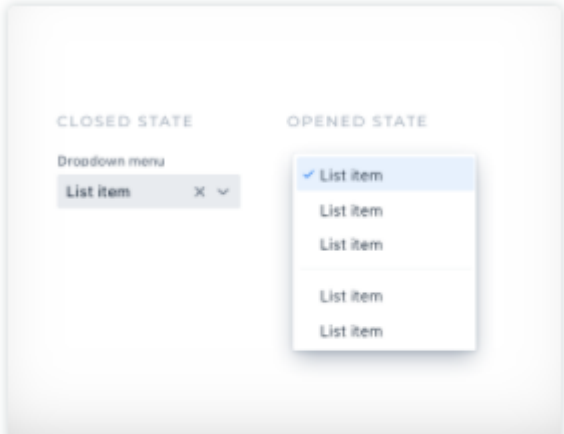
https://vaadin.com/docs/latest/binding-data/data-provider



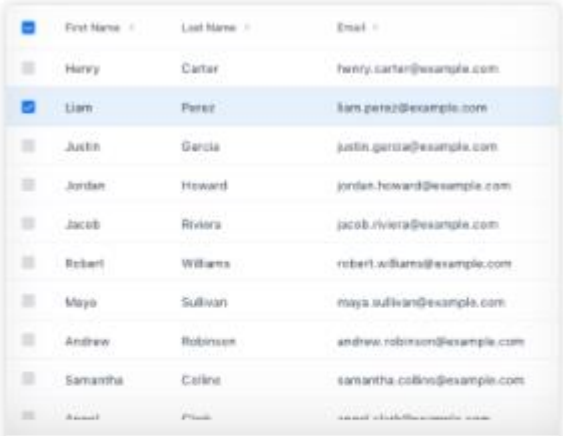
Combo Box
Latest v6.0.1



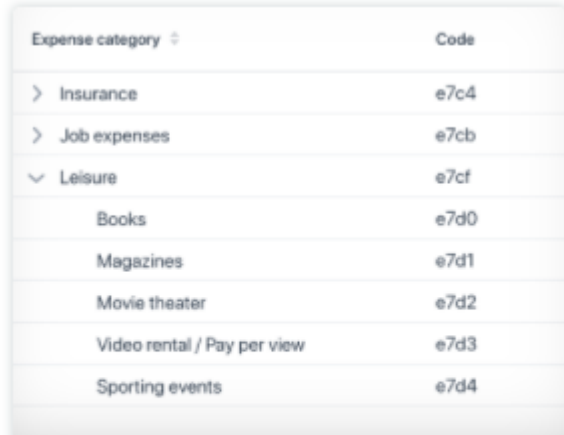
List Box
Latest v2.0.0



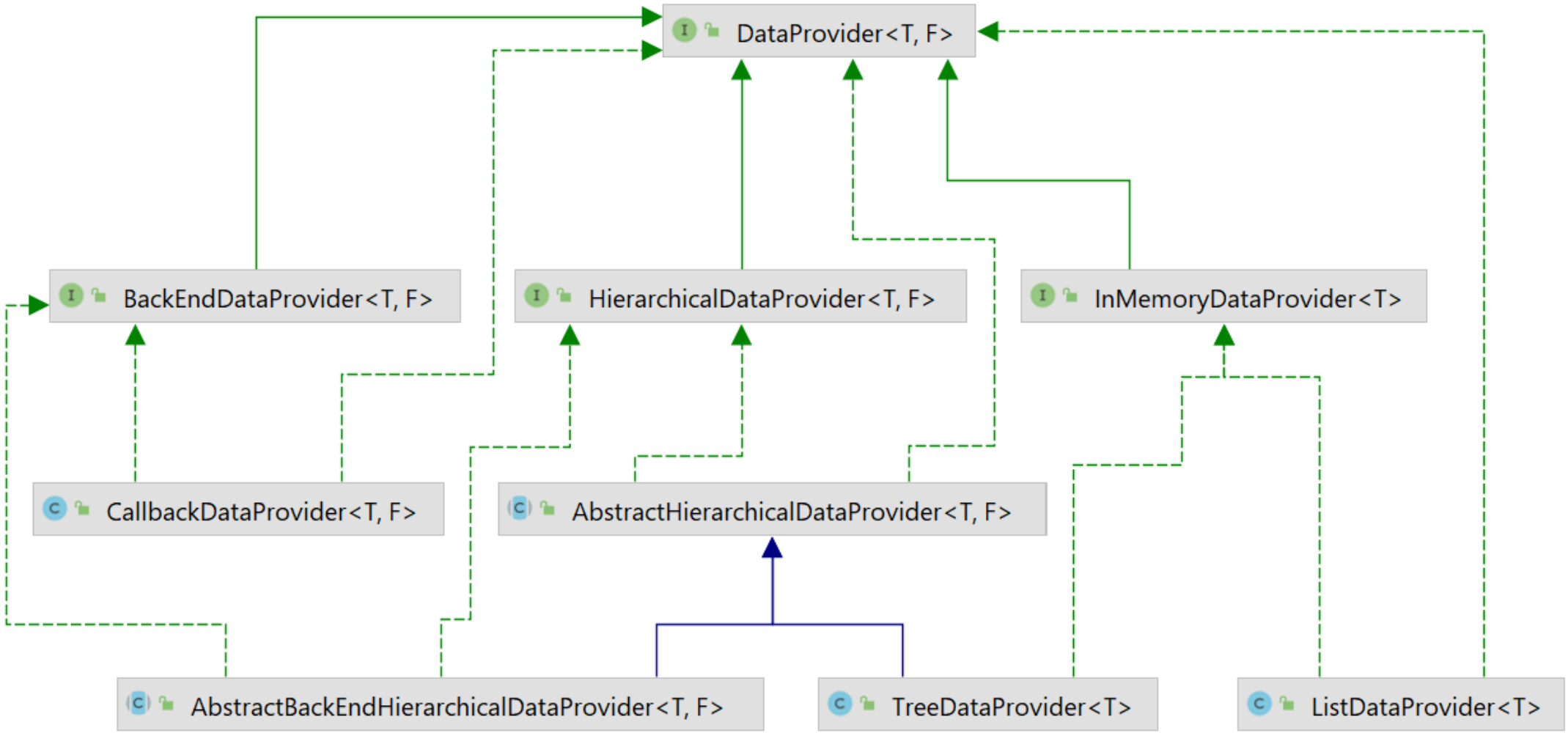
Select
Latest v3.0.0



Grid
Latest v6.0.2



TreeGrid
Latest v6.0.2


DataProvider<T, F>
BackEndDataProvider<T, F>
HierarchicalDataProvider<T, F>
InMemoryDataProvider<T>
CallbackDataProvider<T, F>
AbstractHierarchicalDataProvider<T, F>
AbstractBackEndHierarchicalDataProvider<T, F>
TreeDataProvider<T>
ListDataProvider<T>

DataProvider Variations

- https://github.com/72services/vaadin-database-performance

Exercise: Grids and Data Providers

1. Add grids on all three views to display Workshops and Participants
2. The grids must be sortable

3. Bonus: Make the grids filterable
4. Bonus: Safe the sort order of the columns

Forms and Data Binding

https://vaadin.com/docs/latest/binding-data/components-binder

Binder

- The Binder class allows you to define how the values in a business object are bound to fields in the UI
- Binder reads the values in the business object and converts them from the format expected by the business object to the format expected by the field.
- It also handles the reverse process, converting values from UI fields to the format expected by the business object
- Binder can only bind components that implement the HasValue interface, for example, TextField and ComboBox.
- It is also possible to validate user input and present the validation status to the user in different ways.

Binder Types

- Binder

- BeanValidationBinder
  - Must use property name!

```
.bind("name");
```

Loading From and Saving to Business Objects

- Reading and Writing Automatically
  - `setBean`

- Reading Manually
  - `readBean`
  - `writeBean`
  - `writeBeanIfValid`

Exercise: Forms and DataBinding

1. Add Bean Validation constraints to Workshop and Participant
2. Add a FormLayout to the workshop and participants view
3. Add input fields and a save button
4. Workshop status and topic must be Select fields
5. Participant workshop must be a ComboBox
6. Use the BeanValidationBinder
7. Populate the form when the users selects an item in the Grid

Testing

Karibu Testing

- Testing without browser
- Runs Vaadin in a mock environment

- https://github.com/mvysny/karibu-testing

Testbench

- End-to-end testing with browser
- Based on Selenium

- https://vaadin.com/docs/latest/testing

E2E Other

- Playwright
  - https://playwright.dev/
- Cypress
  - https://www.cypress.io/

Exercise: Testing

1. Add at least one Karibu test for a View

2. Add at least one TestBench test for a View

Security

https://vaadin.com/docs/latest/security

Spring Security Integration

- Extend `VaadinWebSecurityConfigurerAdapter`
- Enables annotation-based security on views

Exercise: Security

1. Add a dependency
    1. `org.springframework.boot:spring-boot-starter-security`
2. Extend VaadinWebSecurityConfigurerAdapter and configure basic auth
3. Add a LoginView
4. The public workshop view should be anonymous allowed
5. Workshop and participant view are protected (role USER)

Styling

https://vaadin.com/docs/latest/styling

Theming

- Base on default Lumo theme

```
@Theme("my-theme")
```

- Folder structure
  - `frontend`
    - `themes`
      - `my-theme`
        - `components`
        - `styles.css`

Exercise: Theming

1. Change the primary color and the default font
2. Add a light-yellow background to all TextFields

Localization

https://vaadin.com/docs/latest/advanced/i18n-localization

I18N

- Implement `I18NProvider`
- `Use: Component.getTranslation()`

- BeanValidationBinder
  - You must use ValidationMessages.properties
  - or https://martinelli.ch/vaadin-beanvalidationbinder-with-custom-resource-bundle/

Exercise: I18N

1. Implement I18NProvider
2. Add a resource bundle for German and English
3. Use getTranslation() in the views

Push

https://vaadin.com/docs/latest/advanced/server-push

Use Cases

- PDF generation takes long
  - Display a progress indicator
  - Display a notification with a download link when finished

- Notify other users when the data has been changed

Push

- Activate with `@Push`
- Uses Atmosphere

- Asnychronous UI update

```
ui.access(() -> label.setText(text));
```

Exercise: Push

1. Notify the users that are on the public workshop view when a new workshop has been added

- Example
  - https://github.com/simasch/vaadin-spring-events

Application Lifecycle

https://vaadin.com/docs/latest/advanced/application-lifecycle

UI

- When a browser first accesses a URL mapped to the servlet of a particular UI class, the Vaadin servlet generates a loader page. The page loads the client-side engine, which in turn loads the UI in a separate request to the Vaadin servlet
- A UI instance is created when the client-side engine makes its first request
- UI instances are cleaned up if no communication is received from them after a certain time

Deploying to Production

https://vaadin.com/docs/latest/production

Production Build

```
mvn package -Pproduction
```

- Builds a JAR or WAR file with all the dependencies and transpiled front-end resources, ready to be deployed.

Exercise: Deploying to Production

1. Create a production build

2. Run the executable JAR

Thank you!

- Web https://martinelli.ch
- E-Mail simon@martinelli.ch
- Twitter simas_ch
- LinkedIn https://linkedin.com/in/simonmartnelli