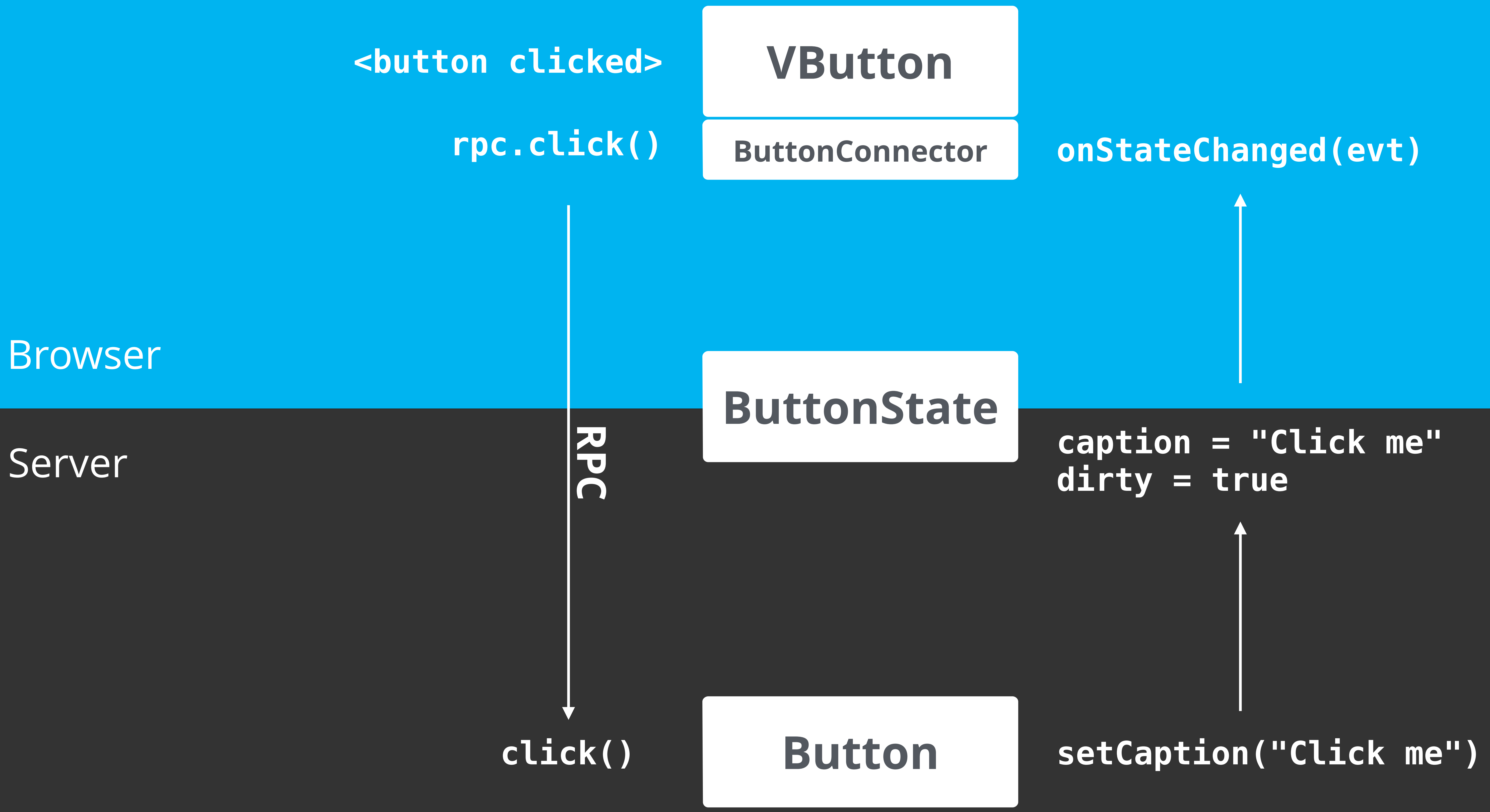
Basics

<code-demo>

How does this work?



Browser

Server

<button clicked>

VButton

rpc.click()

ButtonConnector

onStateChanged(evt)

ButtonState

RPC

caption = "Click me"
dirty = true

click()

Button

setCaption("Click me")

Pros and cons of server side

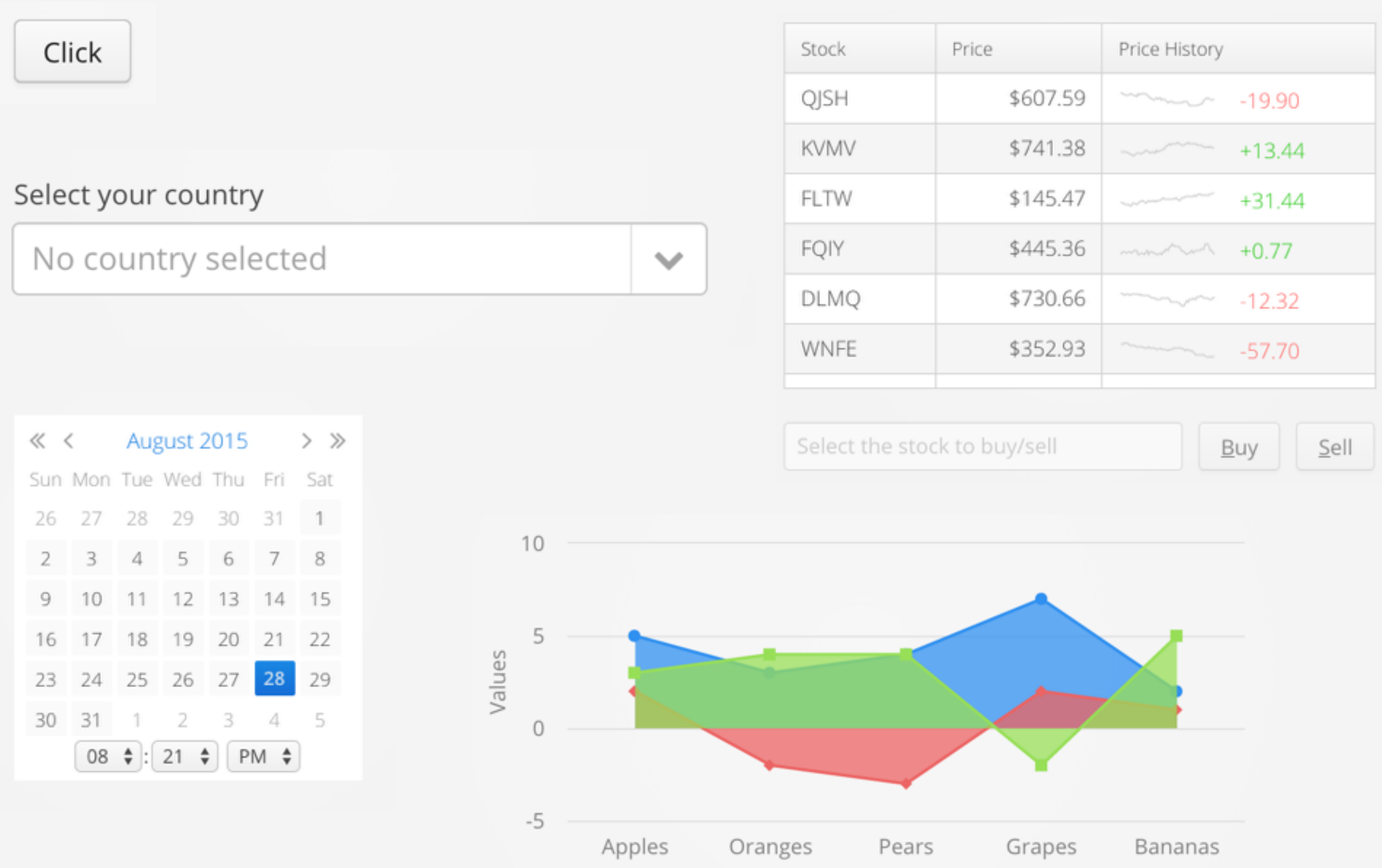
Main concepts

1. Everything is a **Component**

2. Components are organized by **Layouts**

3. **Data binding** ties Components to your data

Everything is a component



Click

Select your country

No country selected

Stock	Price	Price History
QJSH	$607.59	-19.90
KVMV	$741.38	+13.44
FLTW	$145.47	+31.44
FQIY	$445.36	+0.77
DLMQ	$730.66	-12.32
WNFE	$352.93	-57.70

Select the stock to buy/sell

Buy Sell

August 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

08 : 21 PM

vaadin }>

User interface

Add-ons

Data binding

Foundation

Common features

```
setWidth("100%");            setSizeFull();

setHeight("50px");           setSizeUndefined();

setEnabled(true);            setVisible(false);
```

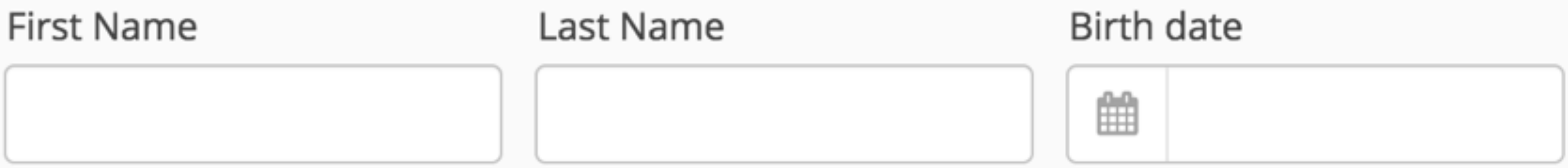
Handling user events

```
Button button = new Button("Cancel");
button.addClickListener(new Button.ClickListener() {
  @Override
  public void buttonClick(Button.ClickEvent clickEvent) {
    fieldGroup.discard();
  }
});

// Or..

button.addClickListener(e -> fieldGroup.discard());
```

Layouts determine how Components are shown

```java
Layout layout = new XLayout();
layout.addComponent(new TextField("First Name"));
layout.addComponent(new TextField("Last Name"));
layout.addComponent(new DateField("Birth date"));
```

```
Layout layout = new HorizontalLayout();
```

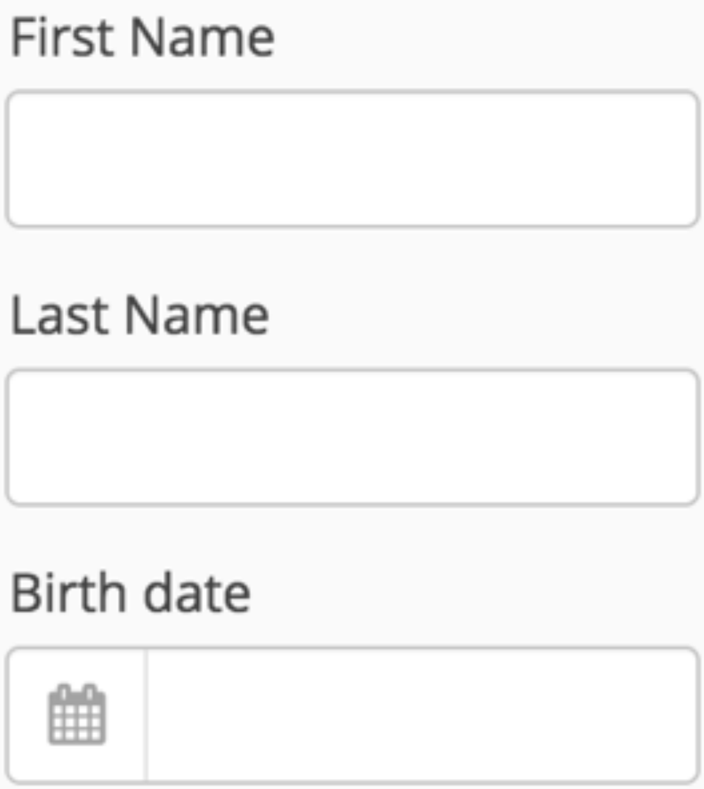


First Name

Last Name

Birth date

```
Layout layout = new VerticalLayout();
```

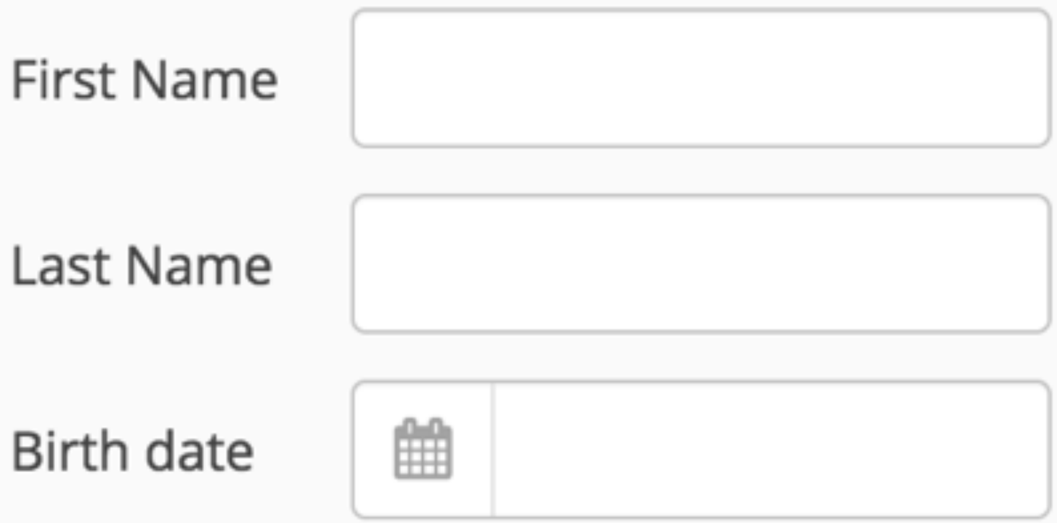


First Name

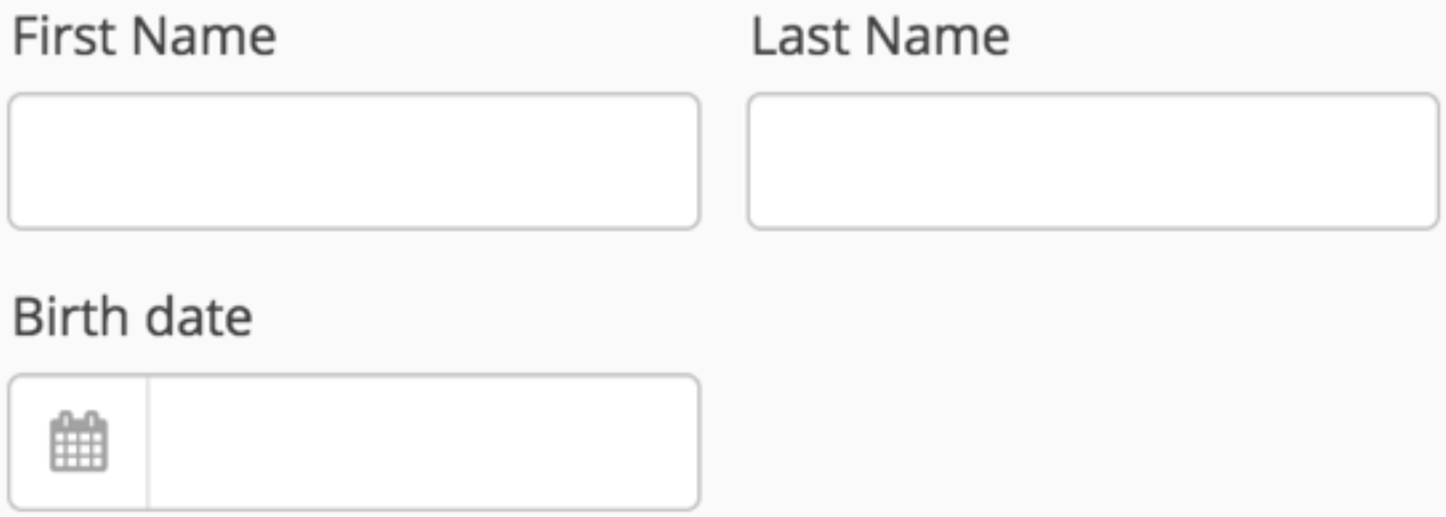
Last Name

Birth date

```
Layout layout = new FormLayout();
```



```java
Layout layout = new GridLayout(2,2);
```



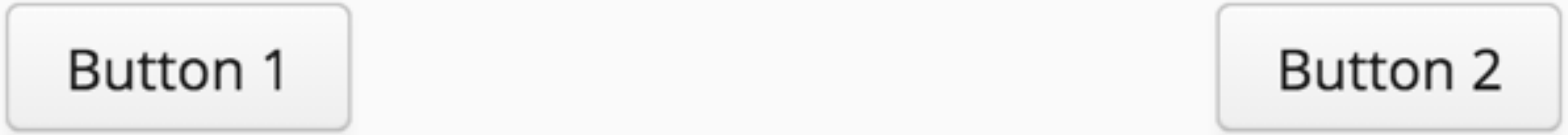
First Name

Last Name

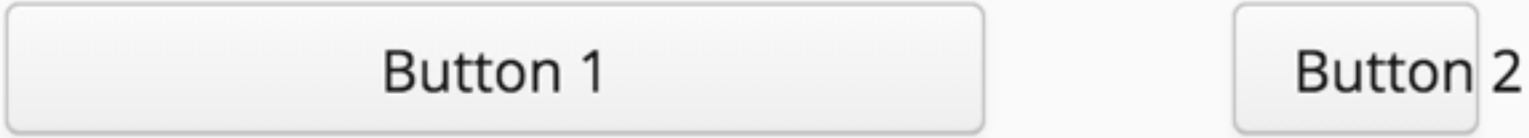
Birth date

Component sizing

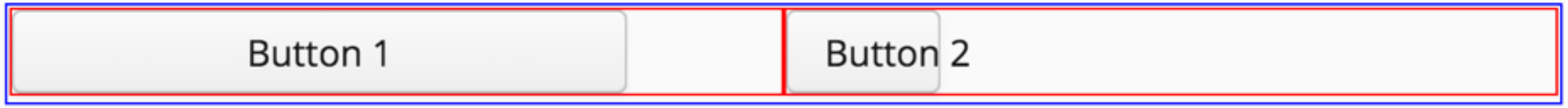
```
HorizontalLayout layout = new HorizontalLayout();
layout.setWidth("100%");
Button button1 = new Button("Button 1");
Button button2 = new Button("Button 2");
layout.addComponents(button1, button2);
```



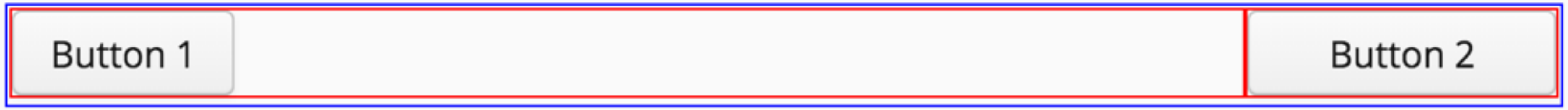
```
button1.setWidth("80%"); button2.setWidth("20%");
```



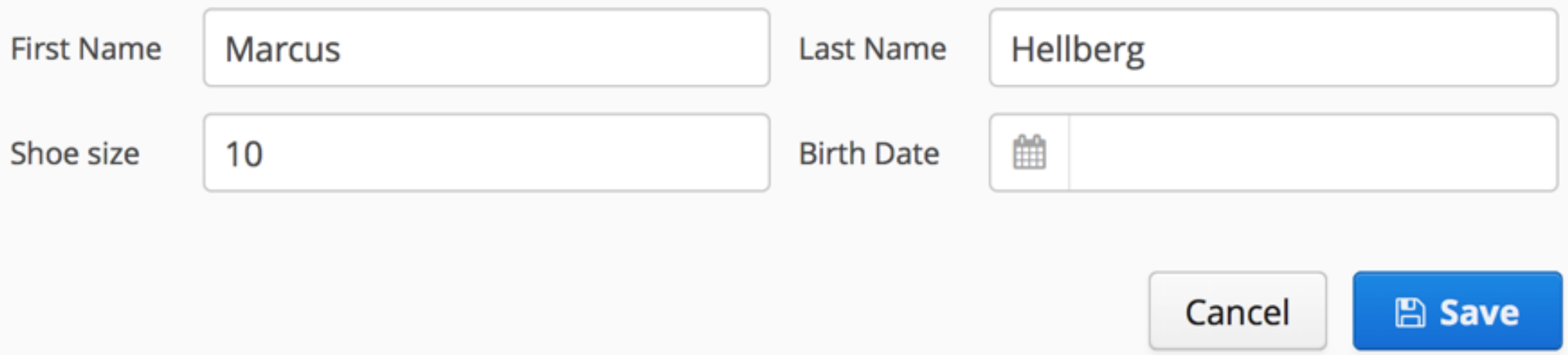
?!



```
layout.setExpandRatio(button1, 0.8f);
layout.setExpandRatio(button2, 0.2f);
button2.setWidth("100%");
```


Button 1
Button 2

Nest layouts to create more complex layouts



First Name Marcus

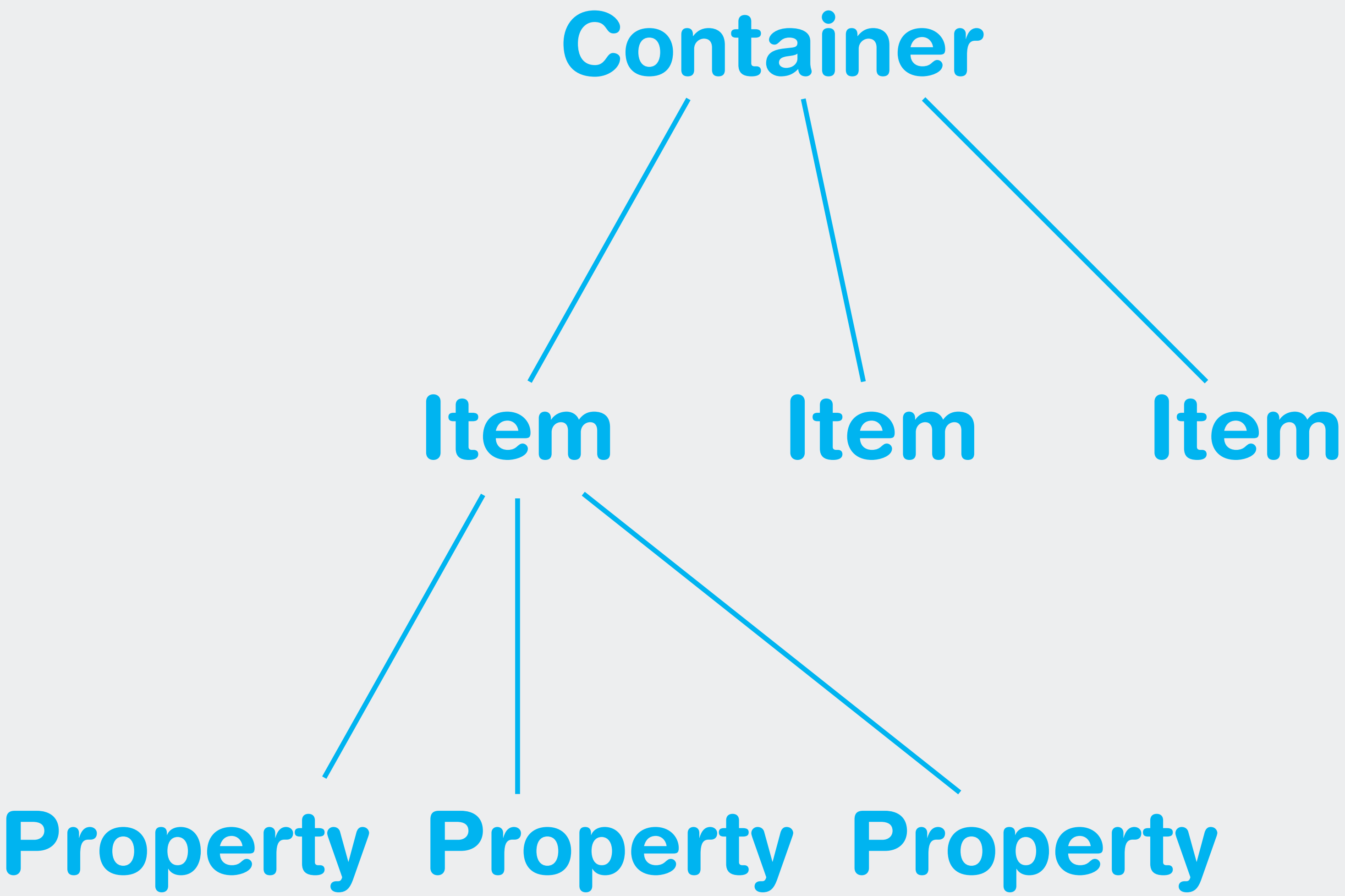
Last Name Hellberg

Shoe size 10

Birth Date

> demo

Binding components to data

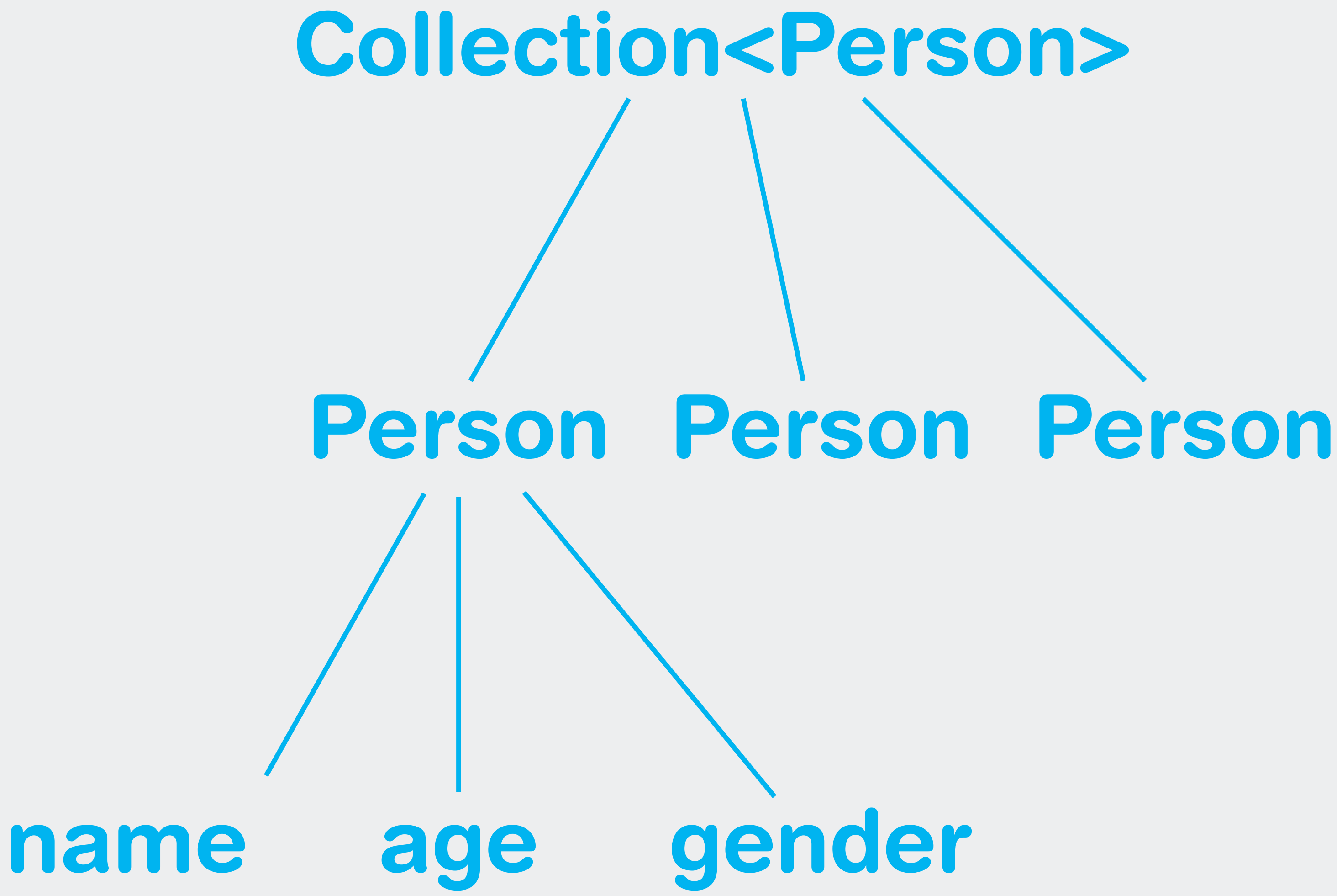


Container

Item Item Item

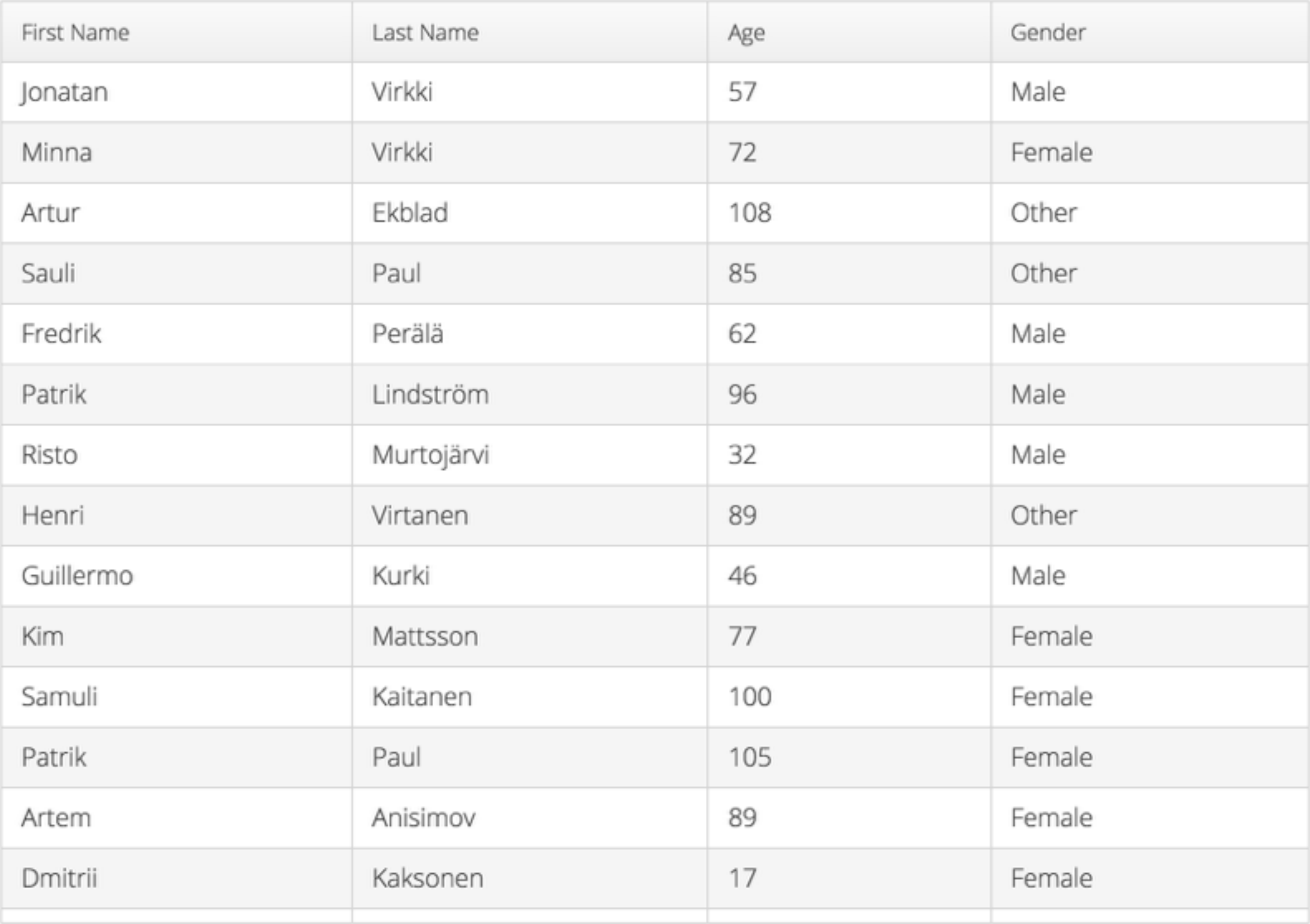
Property Property Property

Basically...



+ listeners

```java
@Inject
private PersonService personService;


@Override
protected void init(VaadinRequest request) {

  BeanItemContainer<Person> container =
    new BeanItemContainer<Person>(
         Person.class, personService.findAll());

    Grid grid = new Grid(container);

    setContent(grid);
}
```

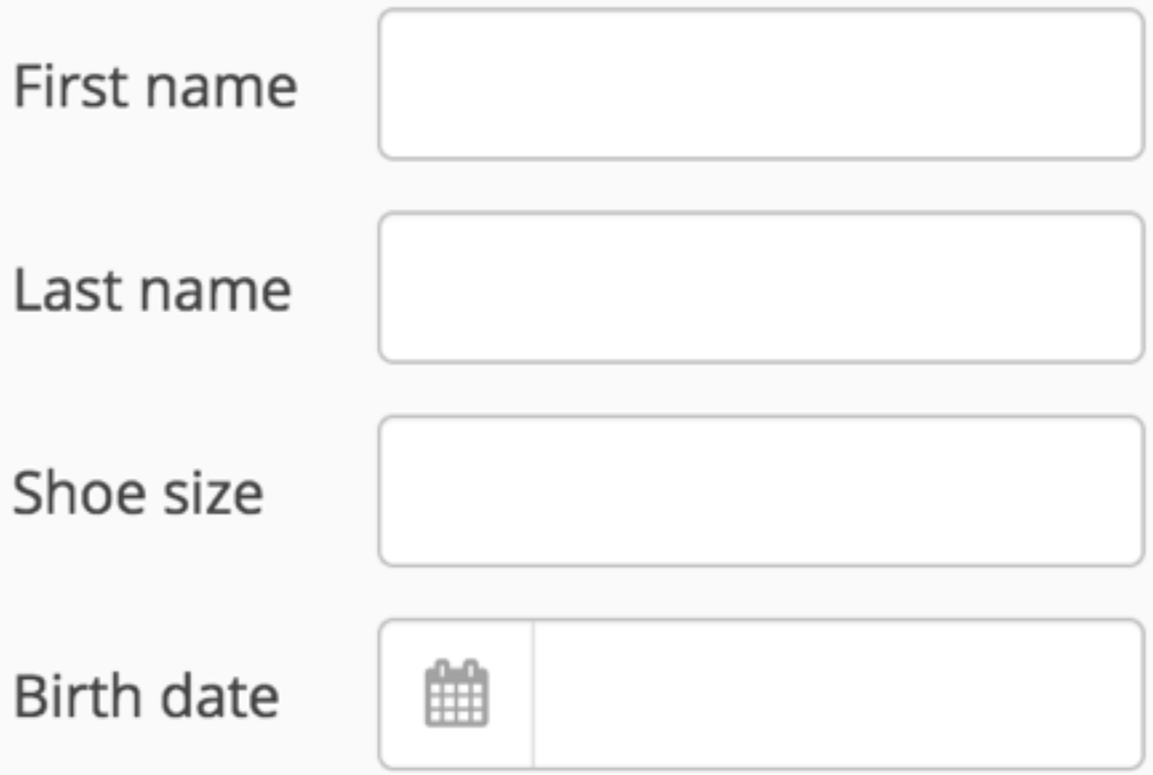


First Name	Last Name	Age	Gender
Jonatan	Virkki	57	Male
Minna	Virkki	72	Female
Artur	Ekblad	108	Other
Sauli	Paul	85	Other
Fredrik	Perälä	62	Male
Patrik	Lindström	96	Male
Risto	Murtojärvi	32	Male
Henri	Virtanen	89	Other
Guillermo	Kurki	46	Male
Kim	Mattsson	77	Female
Samuli	Kaitanen	100	Female
Patrik	Paul	105	Female
Artem	Anisimov	89	Female
Dmitrii	Kaksonen	17	Female

```
FieldGroup fieldGroup = new FieldGroup(new BeanItem<>(new Person()));
PersonForm formLayout = new PersonForm();
fieldGroup.bindMemberFields(formLayout);
```

```java
public class PersonForm extends FormLayout {

    private final TextField firstName;
    private final TextField lastName;
    private final TextField shoeSize;
    private final DateField birthDate;

    public PersonForm() {
        firstName = new TextField("First name");
        lastName = new TextField("Last name");
        shoeSize = new TextField("Shoe size");
        birthDate = new DateField("Birth date");

        addComponents(firstName, lastName, shoeSize, birthDate);
    }
}
```

```java
public class Person {

    private String firstName;
    private String lastName;
    private int shoeSize;
    private Date birthDate;

    // Getters and setters
}
```



First name

Last name

Shoe size

Birth date

Field methods

```java
TextField emailField = new TextField();
emailField.setRequired(true);
emailField.setReadOnly(false);

emailField.addValidator(new EmailValidator("Incorrect email"));
emailField.addValidator(new BeanValidator(Person.class, "email"));
```