



BUILDING WEB APPS IN JAVA

 @MARCUSHELLBERG

vaadin}>

Apache 2.0

Apache 2.0

150k+ active developers

40% of Fortune 100 companies use Vaadin

Apache 2.0

150k+ active developers

40% of Fortune 100 companies use Vaadin

Apache 2.0

150k+ active developers

#2 Java Web Framework

according to Open HUB stats

vaadin}>



things you'll learn

1. What it takes to build a modern web app
2. The building blocks of Vaadin
3. How to build a full stack app in 30 minutes

MODERN WEB APPS

demo.vaadin.com/dashboard

WELCOME

QuickTickets Dashboard

Username

Password

Sign In

☒ Remember me

WELCOME

QuickTickets Dashboard

Username

Password

Sign In

☒ Remember me

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

```
<nav>
    <a href="/analytics/age" routerLink="/analytics/age">Age</a>
    <a href="/analytics/doctor" routerLink="/analytics/doctor">Doctor</a>
    <a href="/analytics/gender" routerLink="/analytics/gender">Gender</a>
</nav>
...
```

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

```
<nav>
    <a href="/analytics/age" routerLink="/analytics/age">Age</a>
    <a href="/analytics/doctor" routerLink="/analytics/doctor">Doctor</a>
    <a href="/analytics/gender" routerLink="/analytics/gender">Gender</a>
</nav>
...

```

my-view.css

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

```
<nav>
    <a href="/analytics/age" routerLink="/analytics/age">Age</a>
    <a href="/analytics/doctor" routerLink="/analytics/doctor">Doctor</a>
    <a href="/analytics/gender" routerLink="/analytics/gender">Gender</a>
</nav>
...

```

my-view.css

```
nav {
    justify-content: center;
    text-transform: uppercase;
    margin-bottom: 50px;
} ...

```


MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

```
<nav>
    <a href="/analytics/age" routerLink="/analytics/age">Age</a>
    <a href="/analytics/doctor" routerLink="/analytics/doctor">Doctor</a>
    <a href="/analytics/gender" routerLink="/analytics/gender">Gender</a>
</nav>
...

```

my-view.css

```
nav {
    justify-content: center;
    text-transform: uppercase;
    margin-bottom: 50px;
} ...

```

my-view.js

MyController.java

```
@RestController
@RequestMapping("/analytics")
public class AnalyticsController {

    @Autowired
    PatientsRepository patientsRepository;

    @RequestMapping(path = "/age",
        method = RequestMethod.GET)
    public Response getByAge() {
        ...
    }

    @RequestMapping(path = "/length",
        method = RequestMethod.GET)
    public Response getByLength() {
        ...
    }
    ...
}
```

my-view.template.html

```
<nav>
    <a href="/analytics/age" routerLink="/analytics/age">Age</a>
    <a href="/analytics/doctor" routerLink="/analytics/doctor">Doctor</a>
    <a href="/analytics/gender" routerLink="/analytics/gender">Gender</a>
</nav>
...

```

my-view.css

```
nav {
    justify-content: center;
    text-transform: uppercase;
    margin-bottom: 50px;
} ...

```

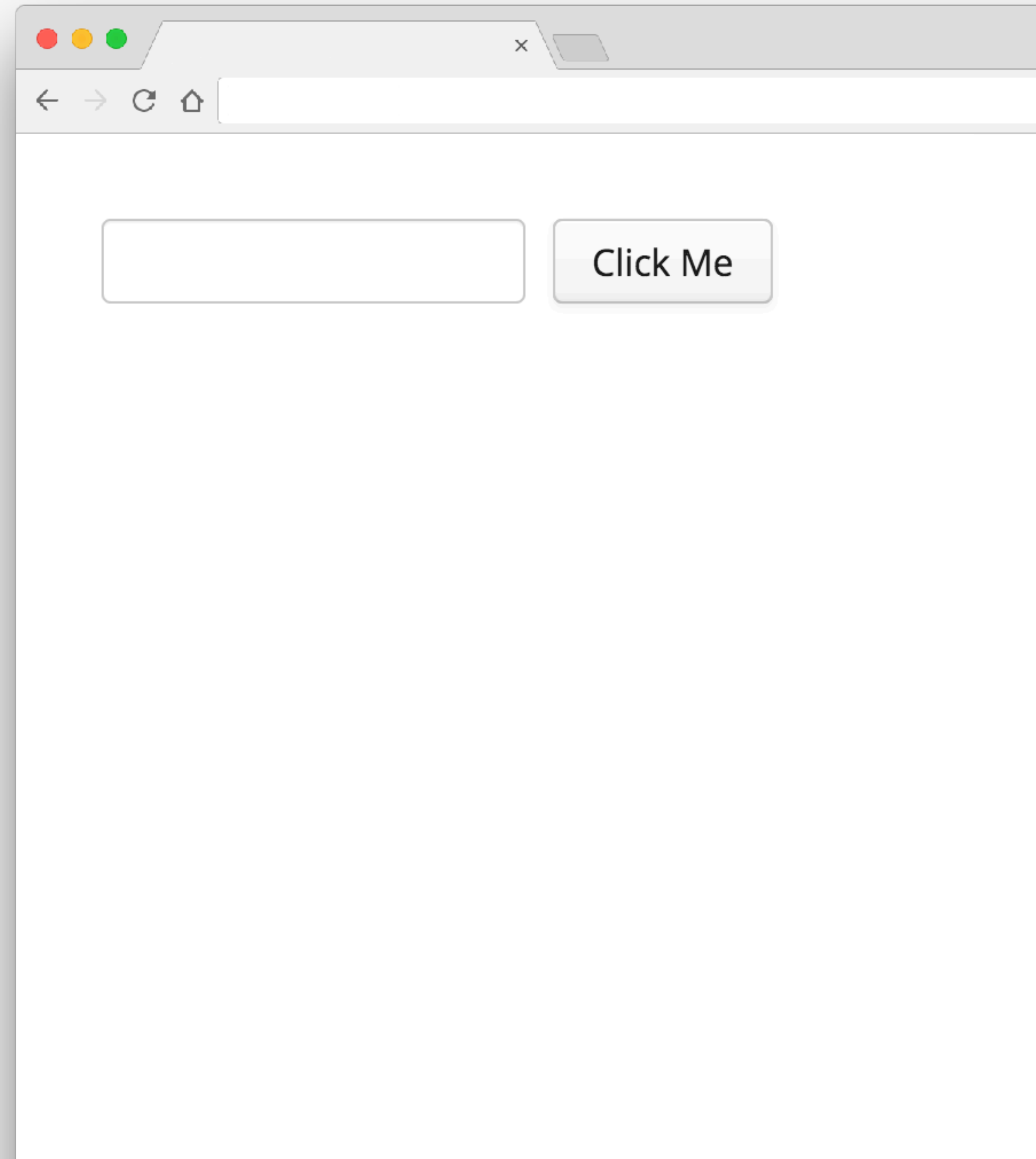
my-view.js

```
getAnalytics(grouping: string = 'age') {
    return this.http
        .get(`${this.config.API_URL}/analytics/${grouping}`)
        .map(res=>res.json()); vaadin}>
}
```

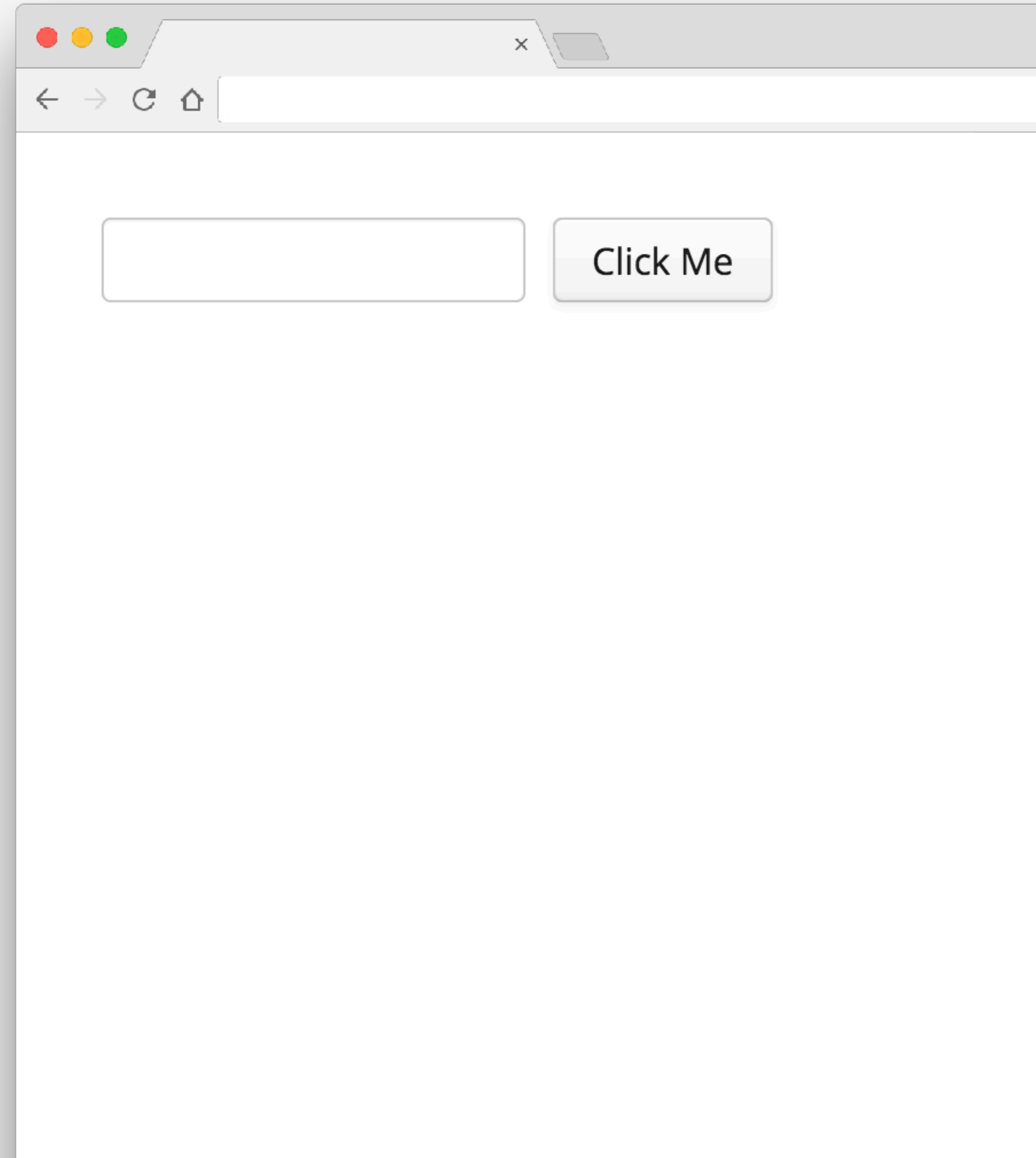
FFS

FIGHT FOR SIMPLICITY

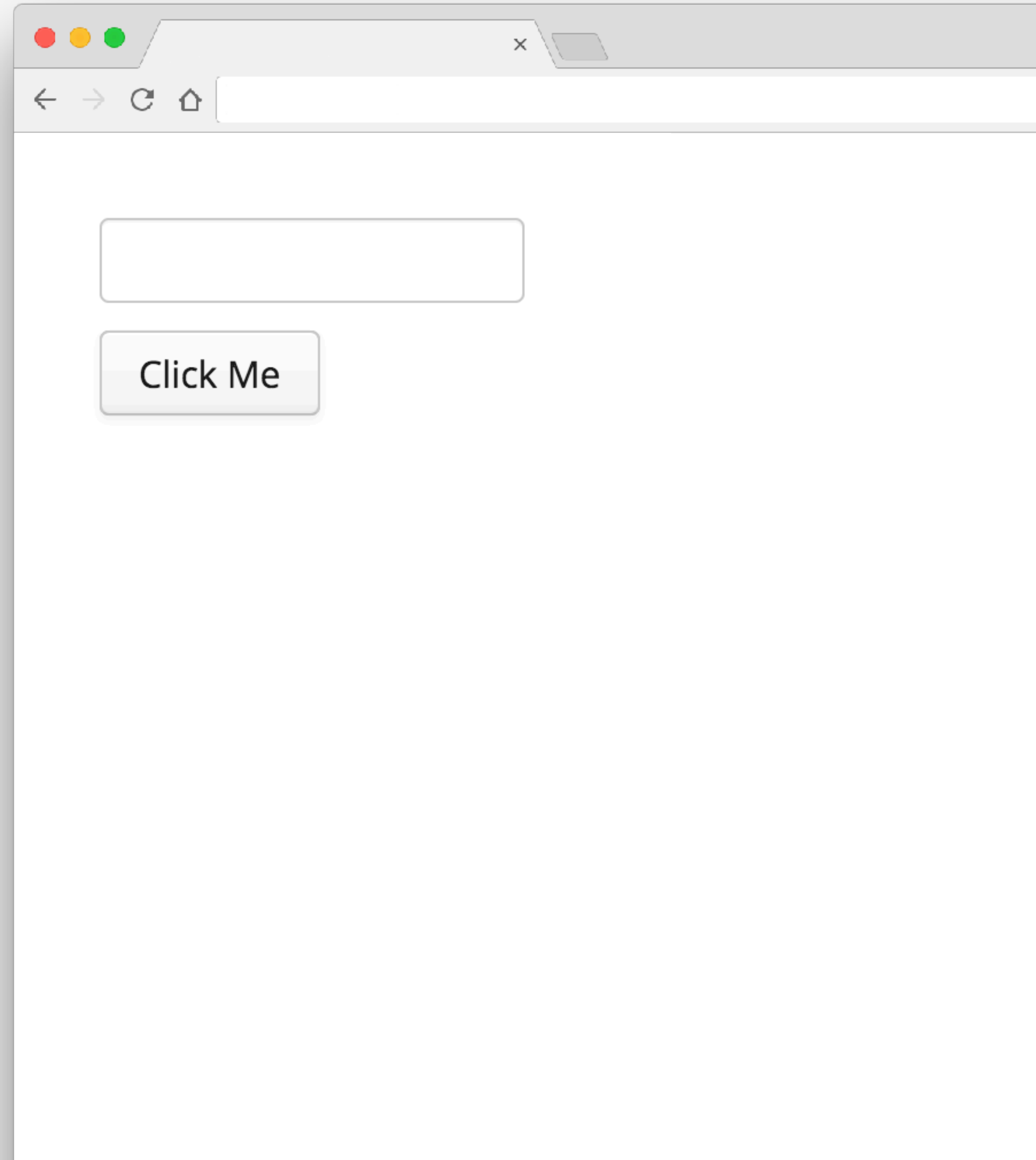
```
new TextField();  
new Button("Click me");
```



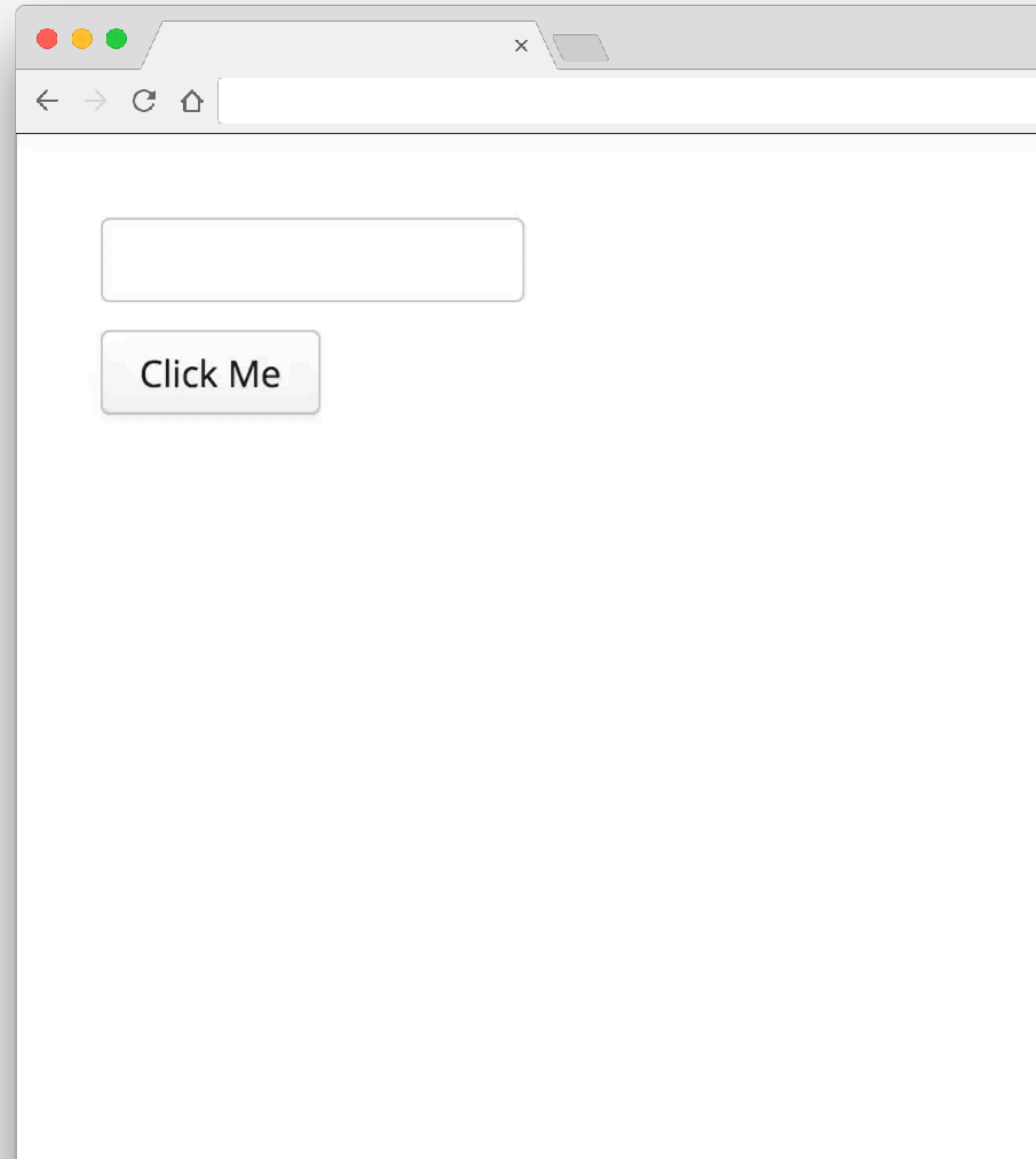
```
new HorizontalLayout(  
    new TextField(),  
    new Button("Click me")  
);
```



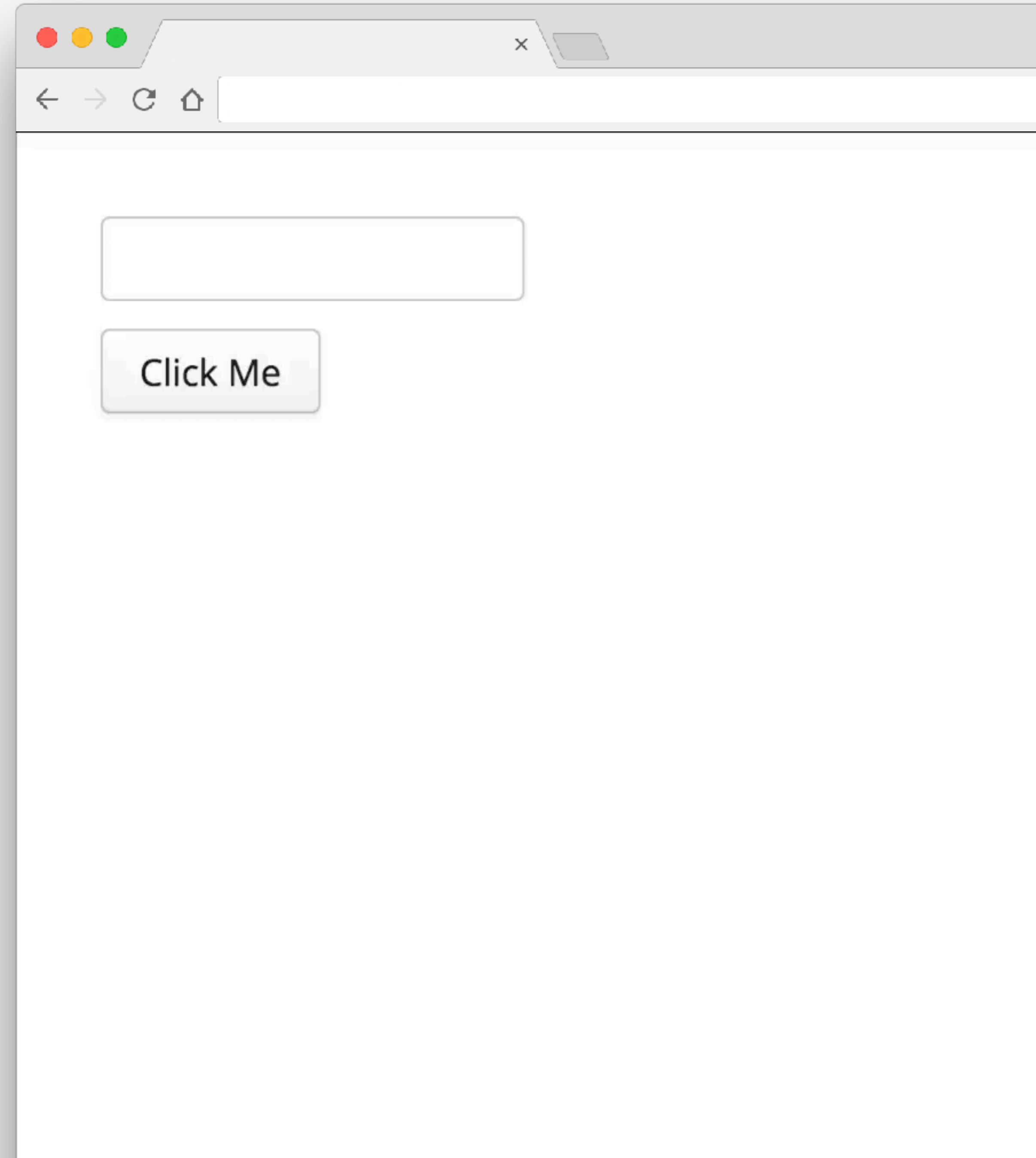
```
new VerticalLayout(  
    new TextField(),  
    new Button("Click me")  
);
```



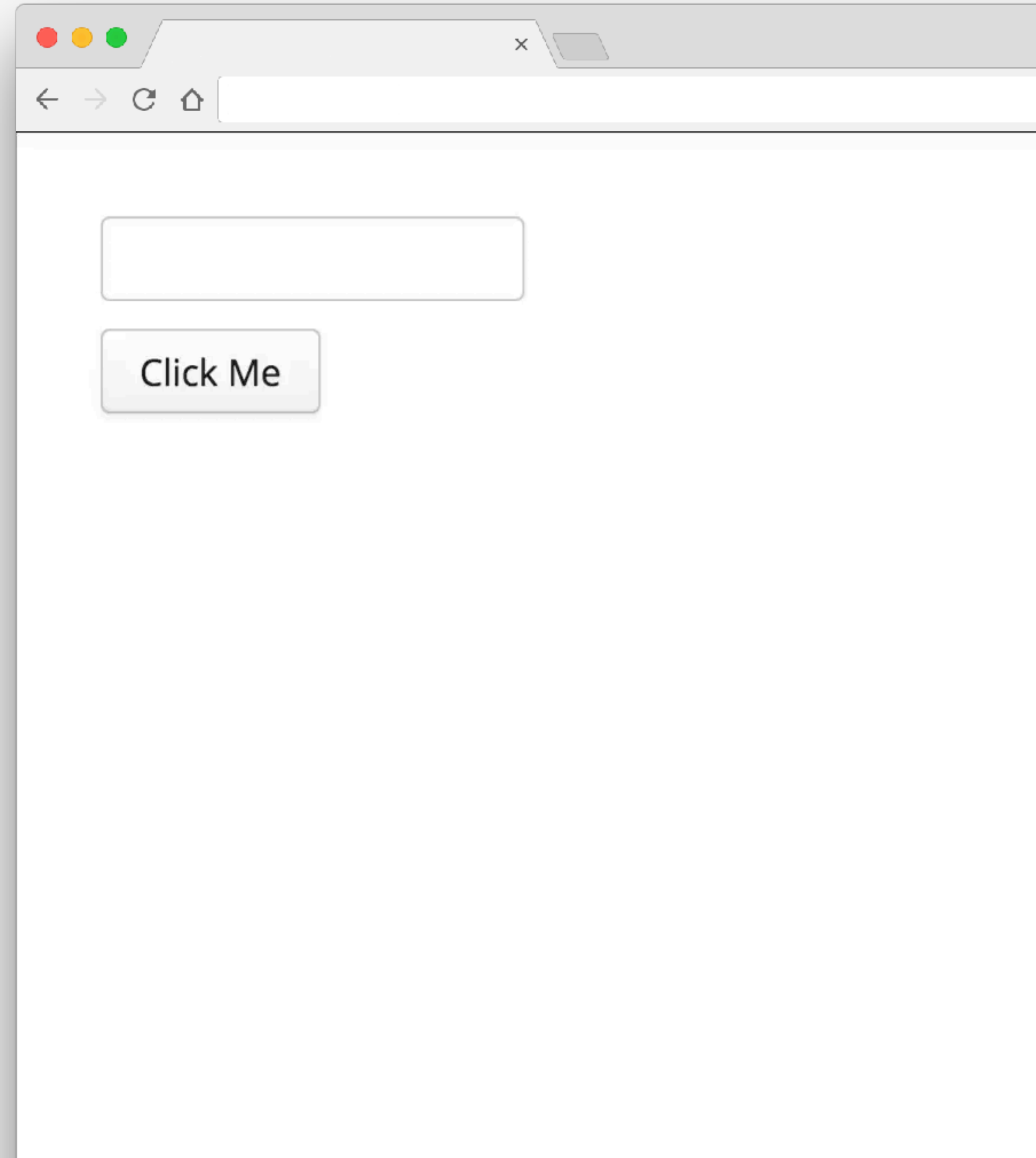
```
layout = new VerticalLayout();  
textField = new TextField();  
button = new Button("Click Me");
```



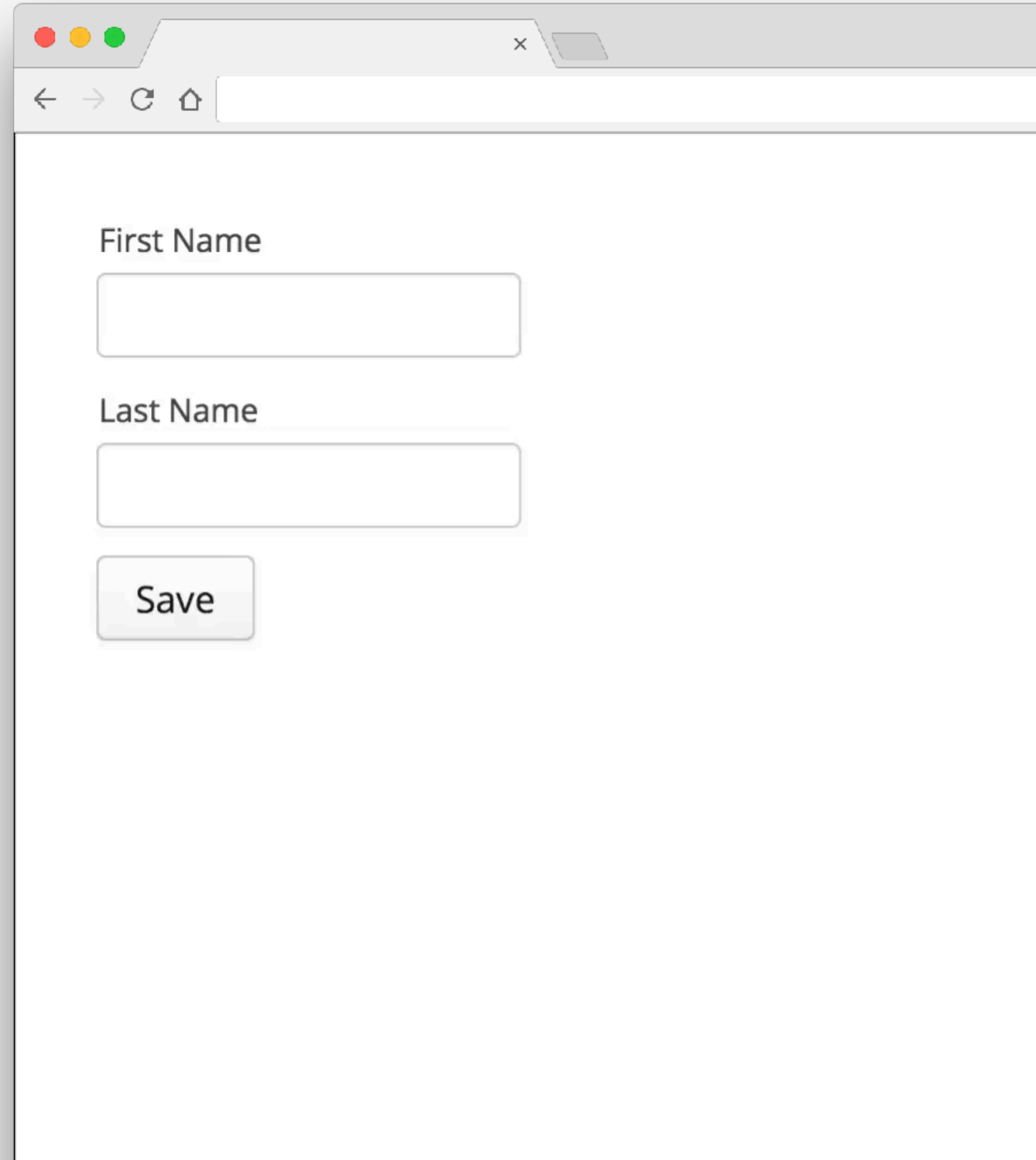

```
layout = new VerticalLayout();  
textField = new TextField();  
button = new Button("Click Me");  
  
button.addClickListener(click ->  
    layout.addComponent(  
        new Label(textField.getValue())  
    ));
```



```
layout = new VerticalLayout();  
textField = new TextField();  
button = new Button("Click Me");  
  
button.addClickListener(click ->  
    layout.addComponent(  
        new Label(textField.getValue())  
    ));
```



```
private TextField firstName;  
private TextField lastName;
```



A web browser window with a single tab. The address bar is empty. The page content includes two text input fields. The first field is labeled "First Name" and the second is labeled "Last Name". Below these fields is a button labeled "Save".

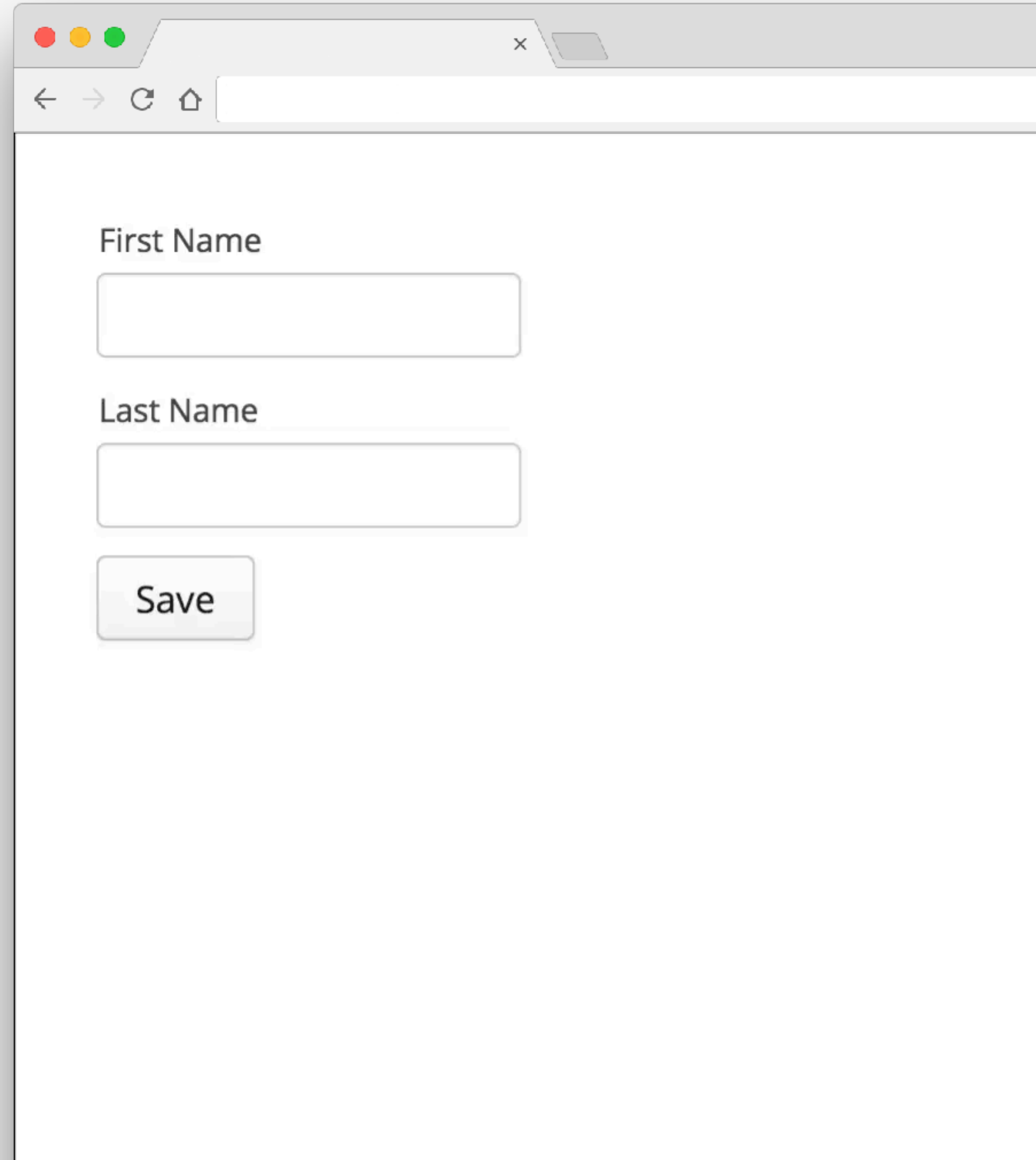
First Name

Last Name

Save

```
private TextField firstName;  
private TextField lastName;
```

```
person = new Person();
```



A web browser window with a single tab. The address bar is empty. The page content includes two text input fields. The first field is labeled "First Name" and the second is labeled "Last Name". Below these fields is a button labeled "Save".

First Name

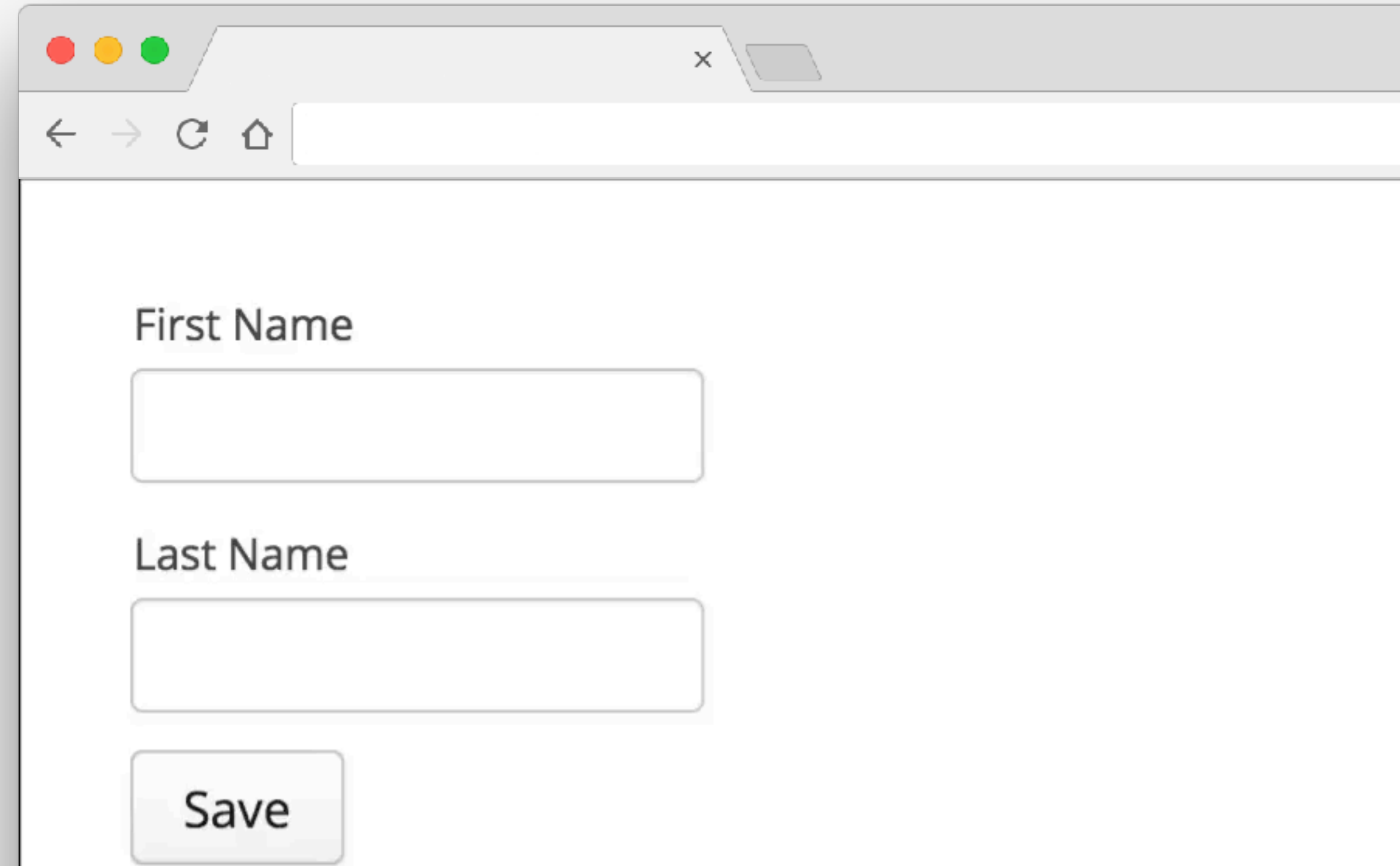
Last Name

Save

```
private TextField firstName;  
private TextField lastName;
```

```
person = new Person();
```

```
binder = new BeanBinder<>(  
    Person.class);
```



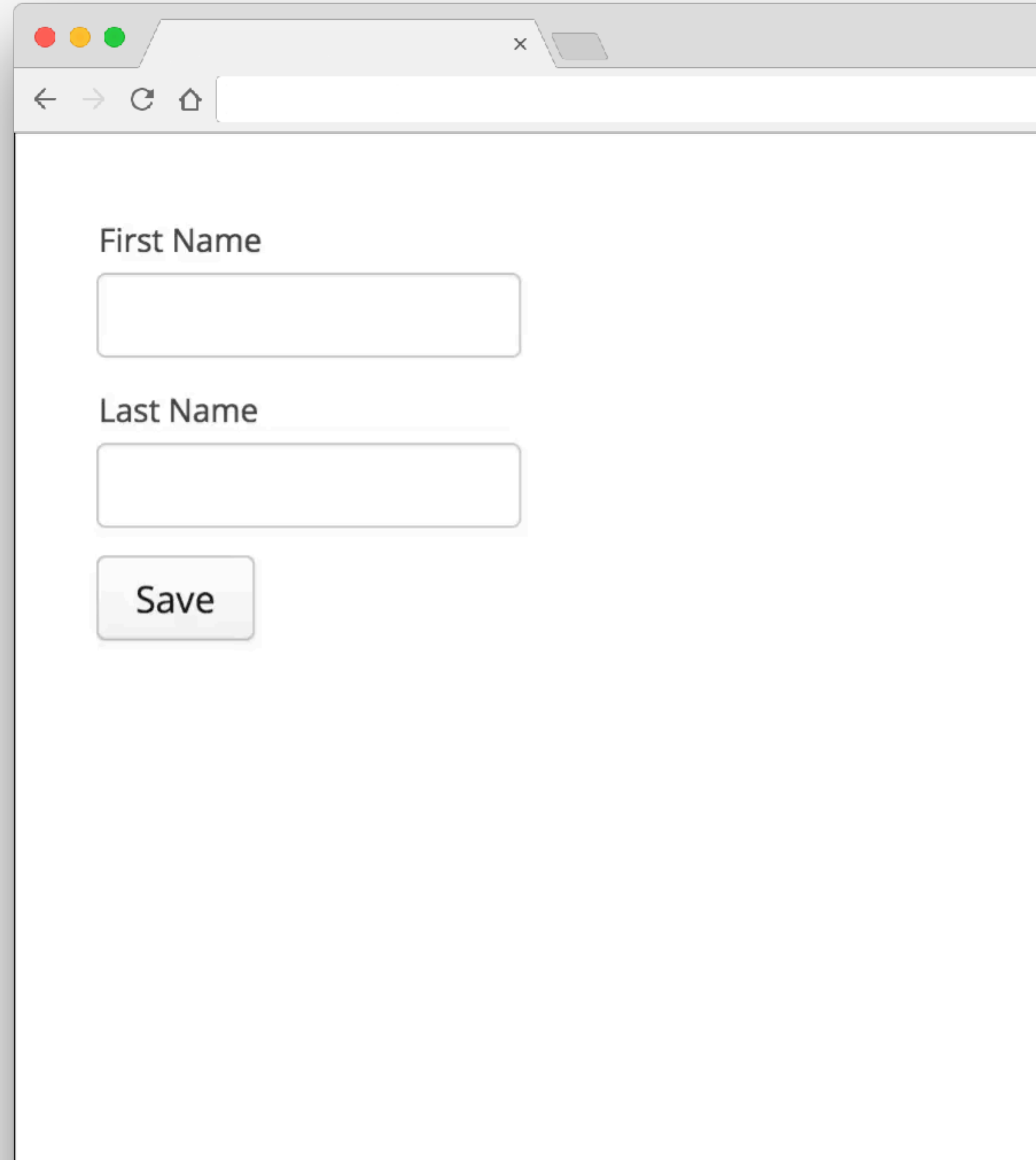
A web browser window with a single tab. The address bar is empty. The page content consists of two text input fields. The first field is labeled "First Name" and the second field is labeled "Last Name". Below the fields is a button labeled "Save".

```
private TextField firstName;  
private TextField lastName;
```

```
person = new Person();
```

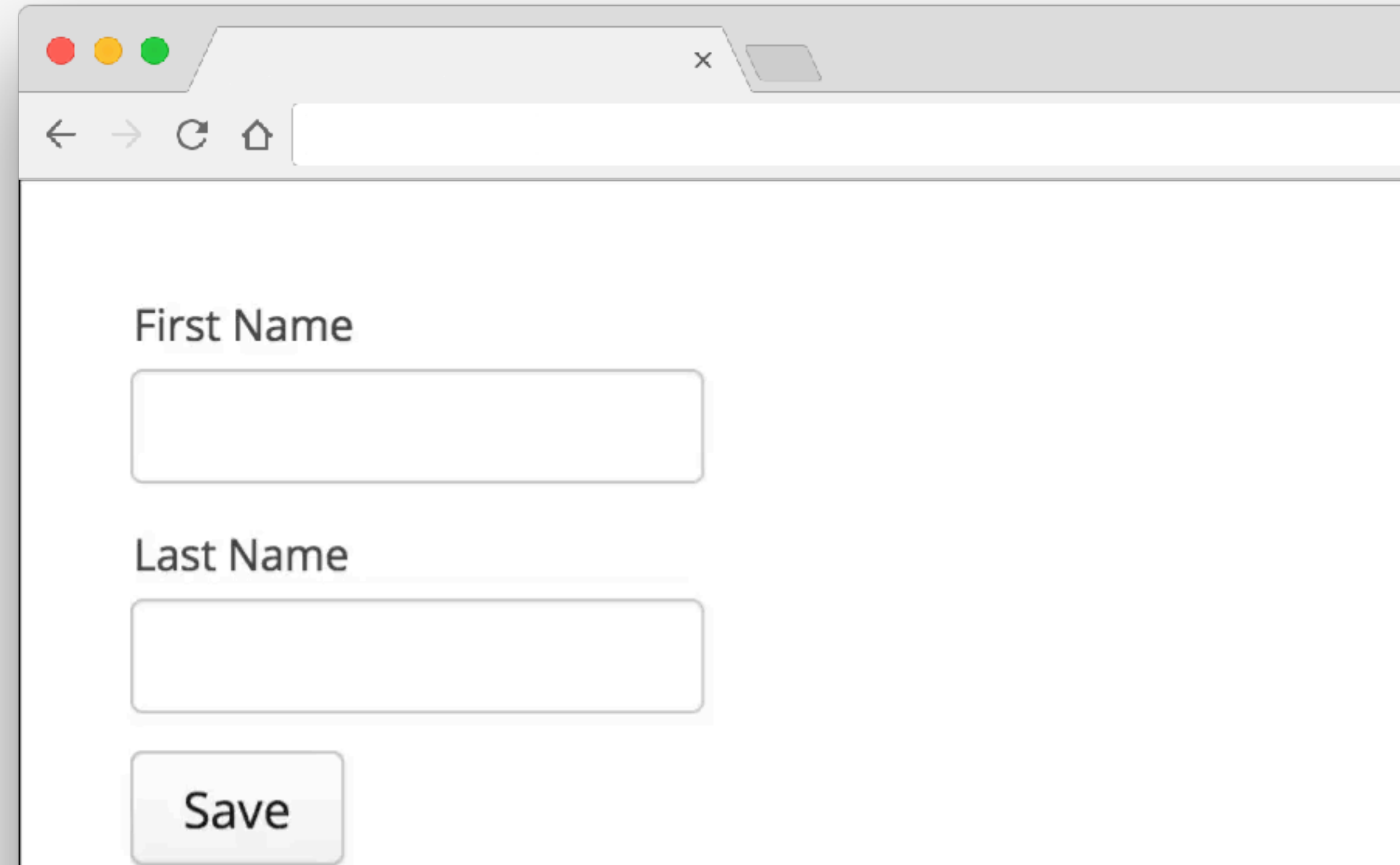
```
binder = new BeanBinder<>(  
    Person.class);
```

```
binder.bindInstanceFields(this);
```



A web browser window with a single tab. The address bar is empty. The page content includes two text input fields. The first field is labeled "First Name" and the second is labeled "Last Name". Below these fields is a button labeled "Save".

```
private TextField firstName;  
private TextField lastName;  
  
person = new Person();  
  
binder = new BeanBinder<>(  
    Person.class);  
  
binder.bindInstanceFields(this);  
  
save = new Button("Save", e -> {  
    // try-catch  
    binder.writeBean(person);  
  
    Notification.show(  
        person.toString());  
});
```

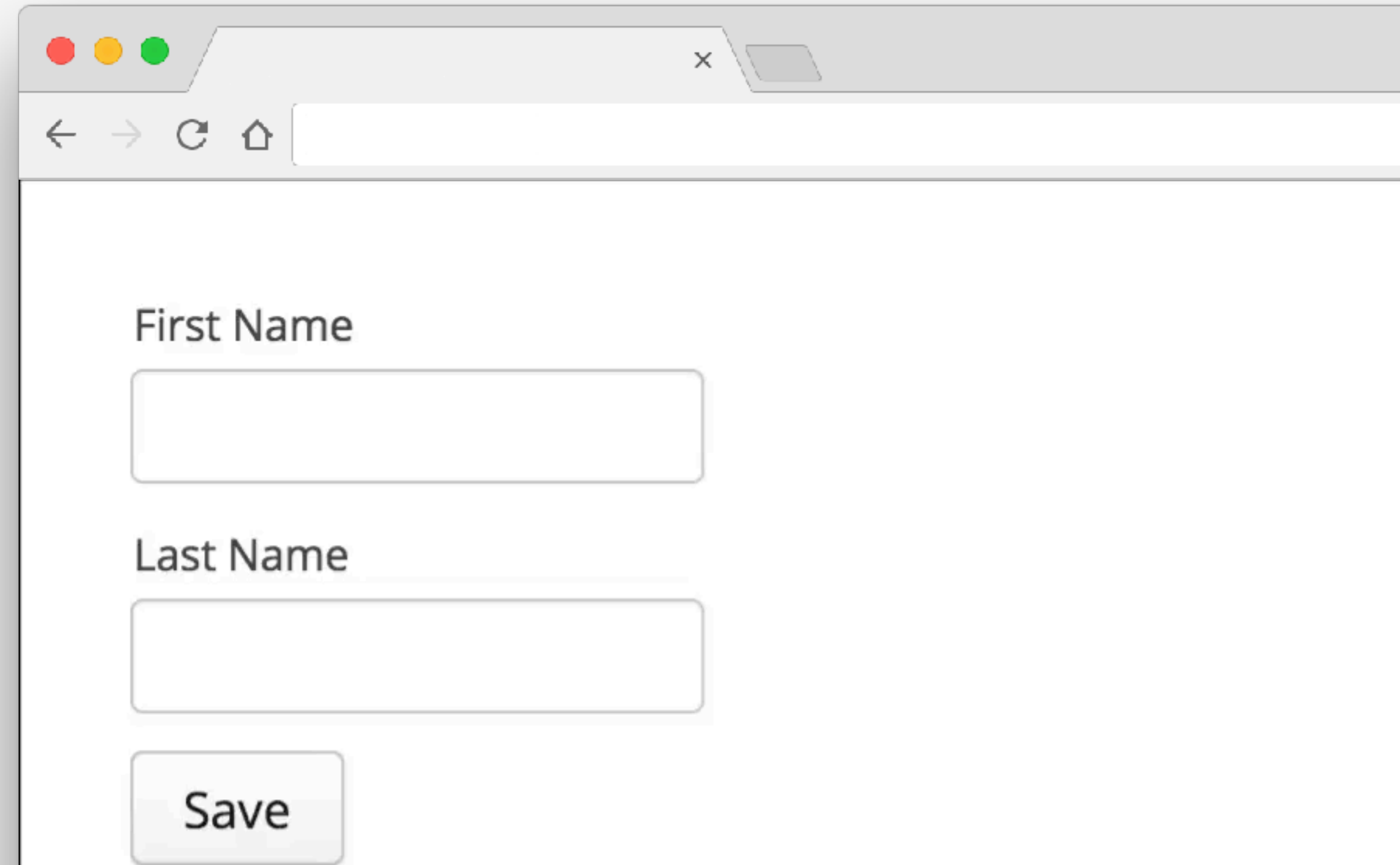


First Name

Last Name

Save

```
private TextField firstName;  
private TextField lastName;  
  
person = new Person();  
  
binder = new BeanBinder<>(  
    Person.class);  
  
binder.bindInstanceFields(this);  
  
save = new Button("Save", e -> {  
    // try-catch  
    binder.writeBean(person);  
  
    Notification.show(  
        person.toString());  
});
```

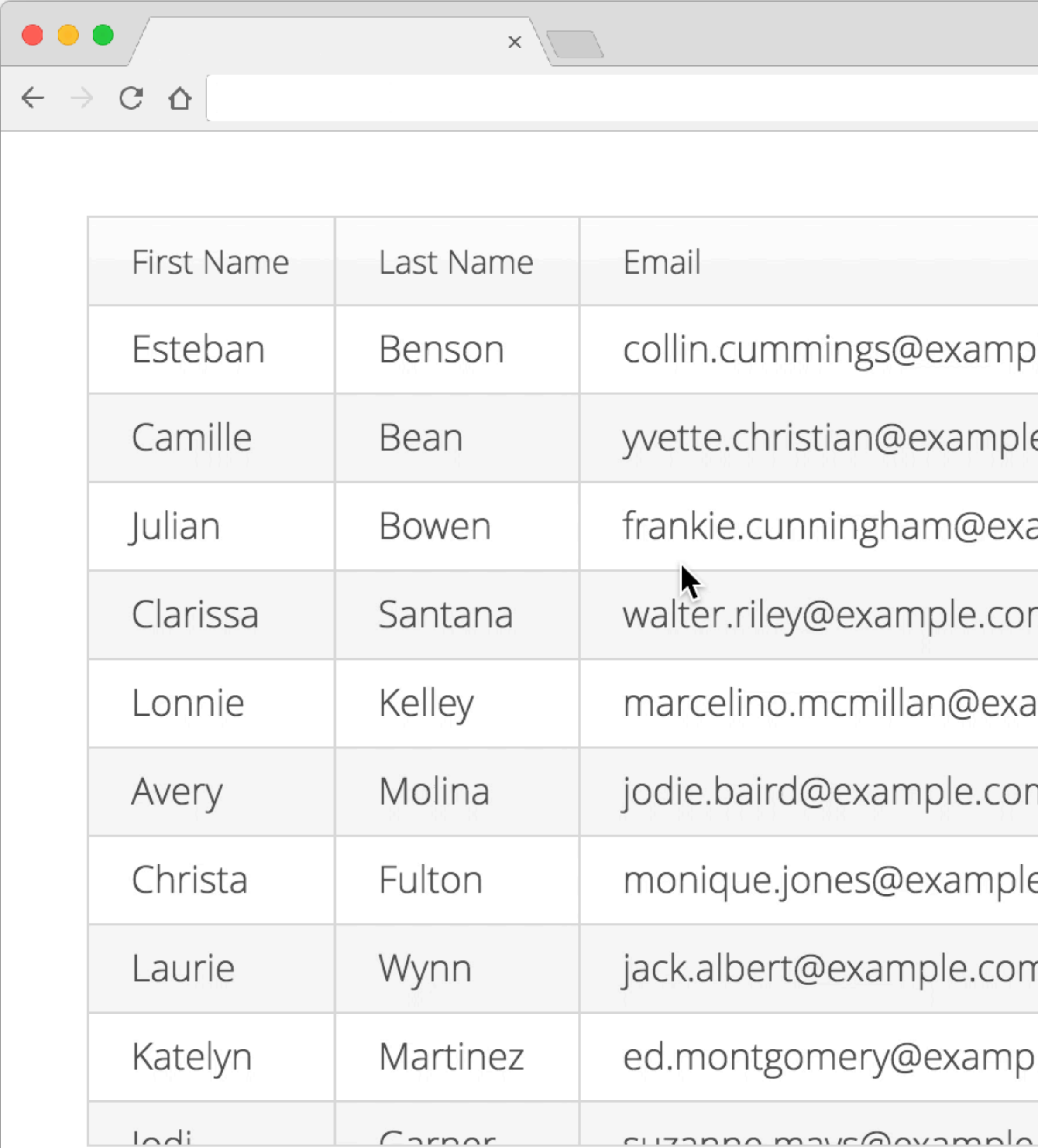


First Name

Last Name

Save

@Autowired
private PersonService service;



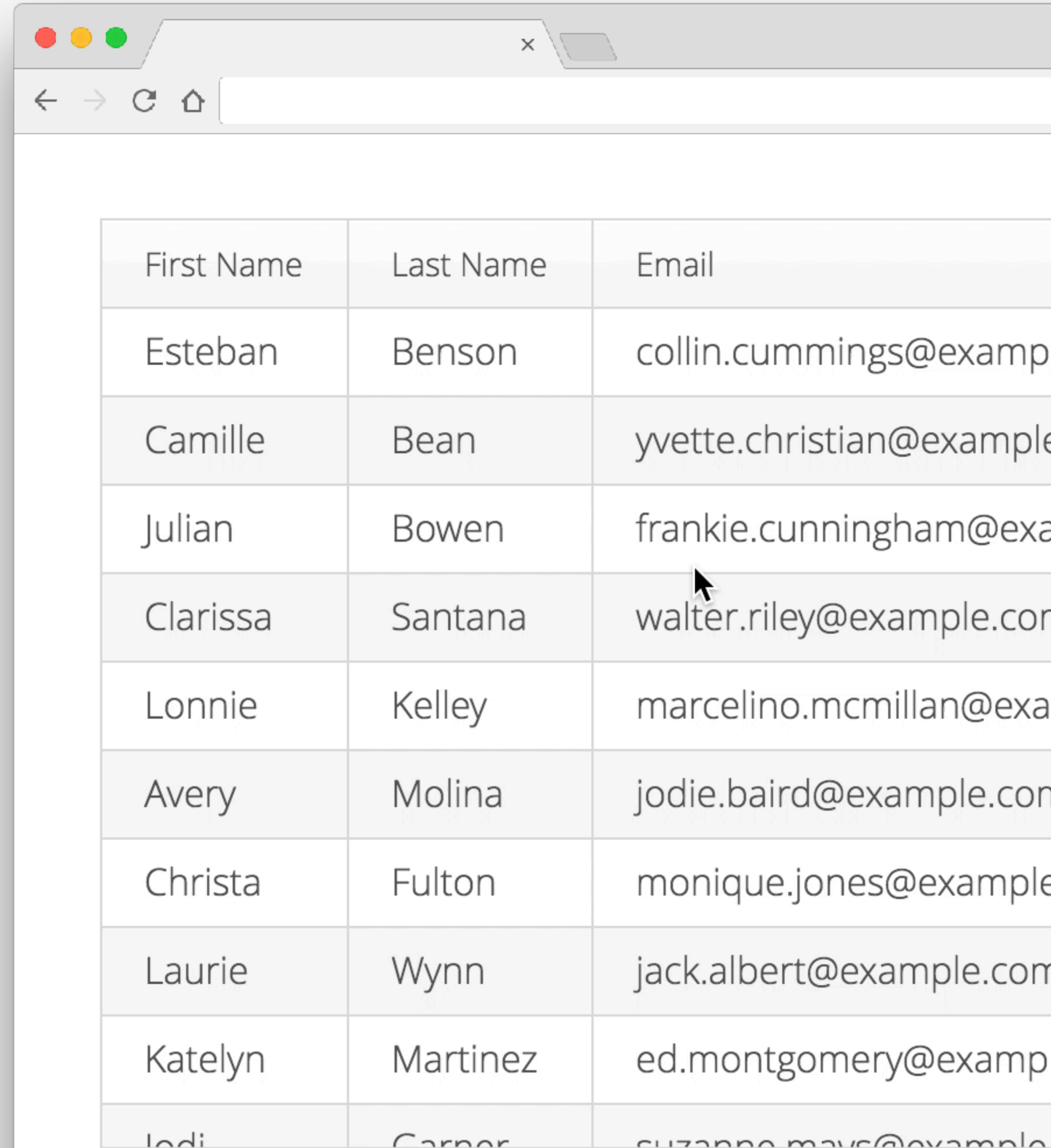
A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The browser has a single tab and a navigation bar with back, forward, and refresh buttons. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fifth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

@Autowired

private PersonService service;

Grid<Person> grid = new Grid<>();

A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fifth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

@Autowired

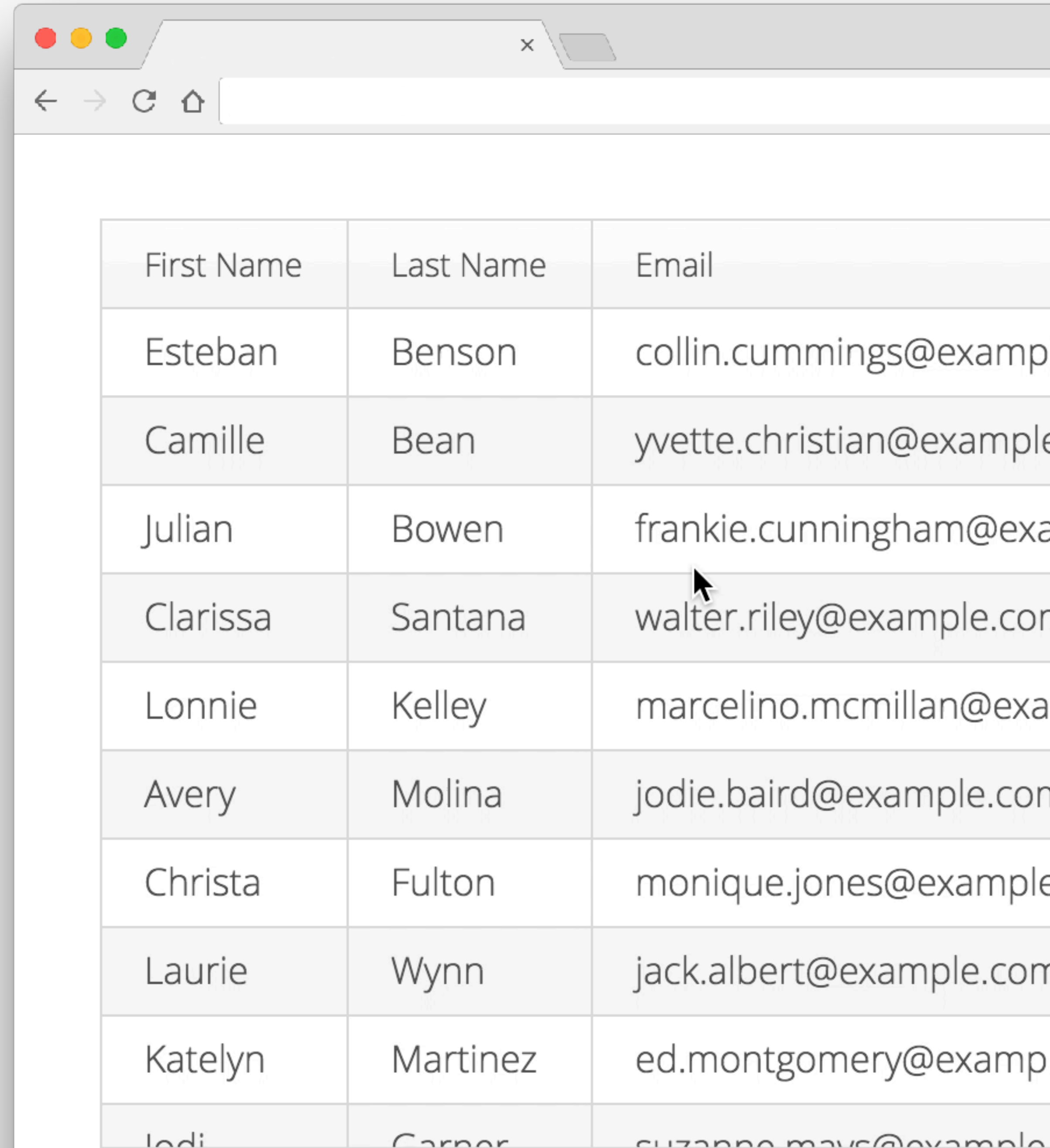
private PersonService service;

```
Grid<Person> grid = new Grid<>();
```

```
grid.addColumn("First name",  
               Person::getFirstName);
```

```
grid.addColumn("Last name",  
               Person::getLastName);
```

```
grid.addColumn("Email",  
               Person::getEmail);
```

A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fourth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

@Autowired

private PersonService service;

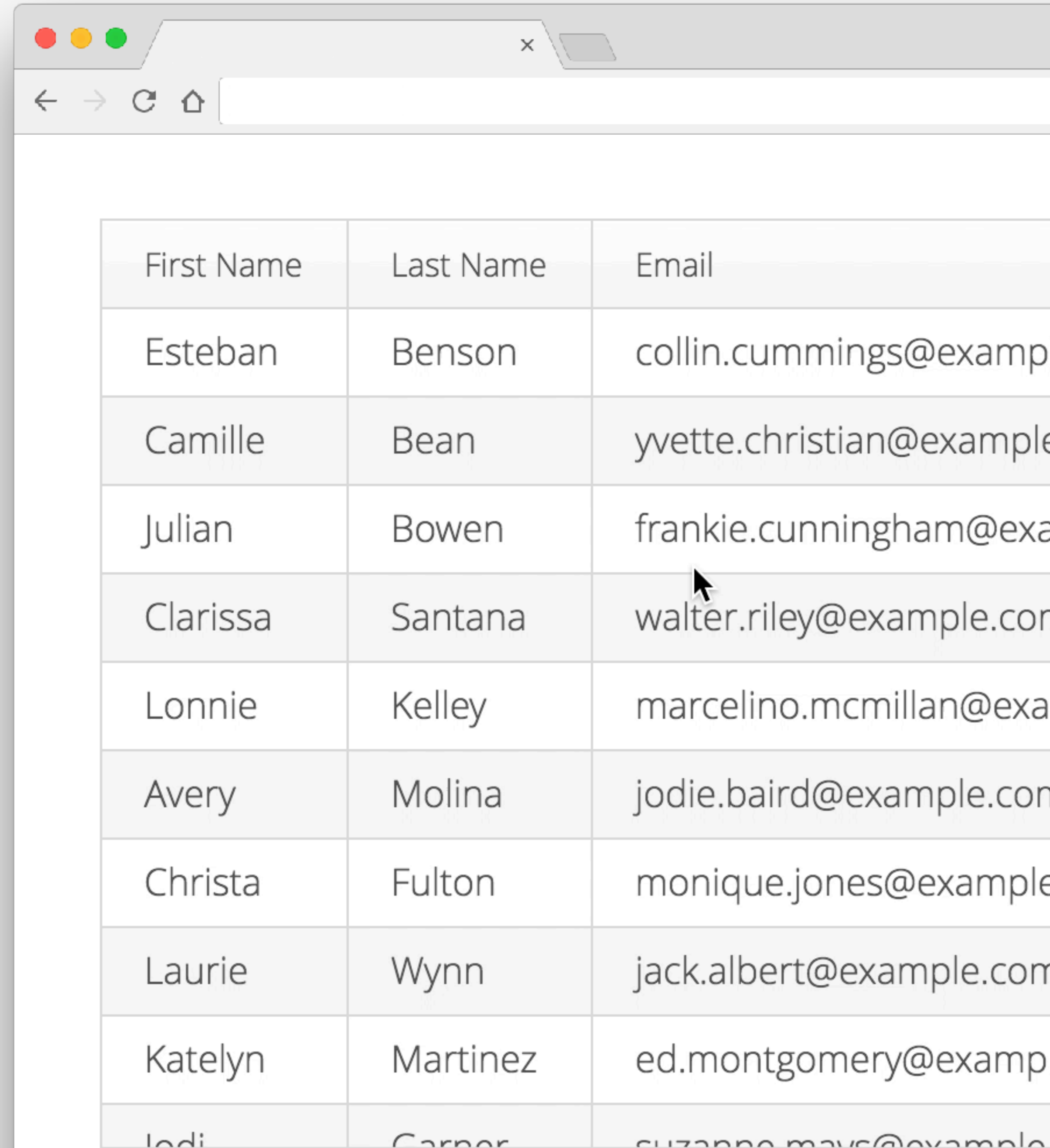
Grid<Person> grid = new Grid<>();

grid.addColumn("First name",
Person::getFirstName);

grid.addColumn("Last name",
Person::getLastName);

grid.addColumn("Email",
Person::getEmail);

grid.setItems(service.getPeople());

A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fourth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

@Autowired

private PersonService service;

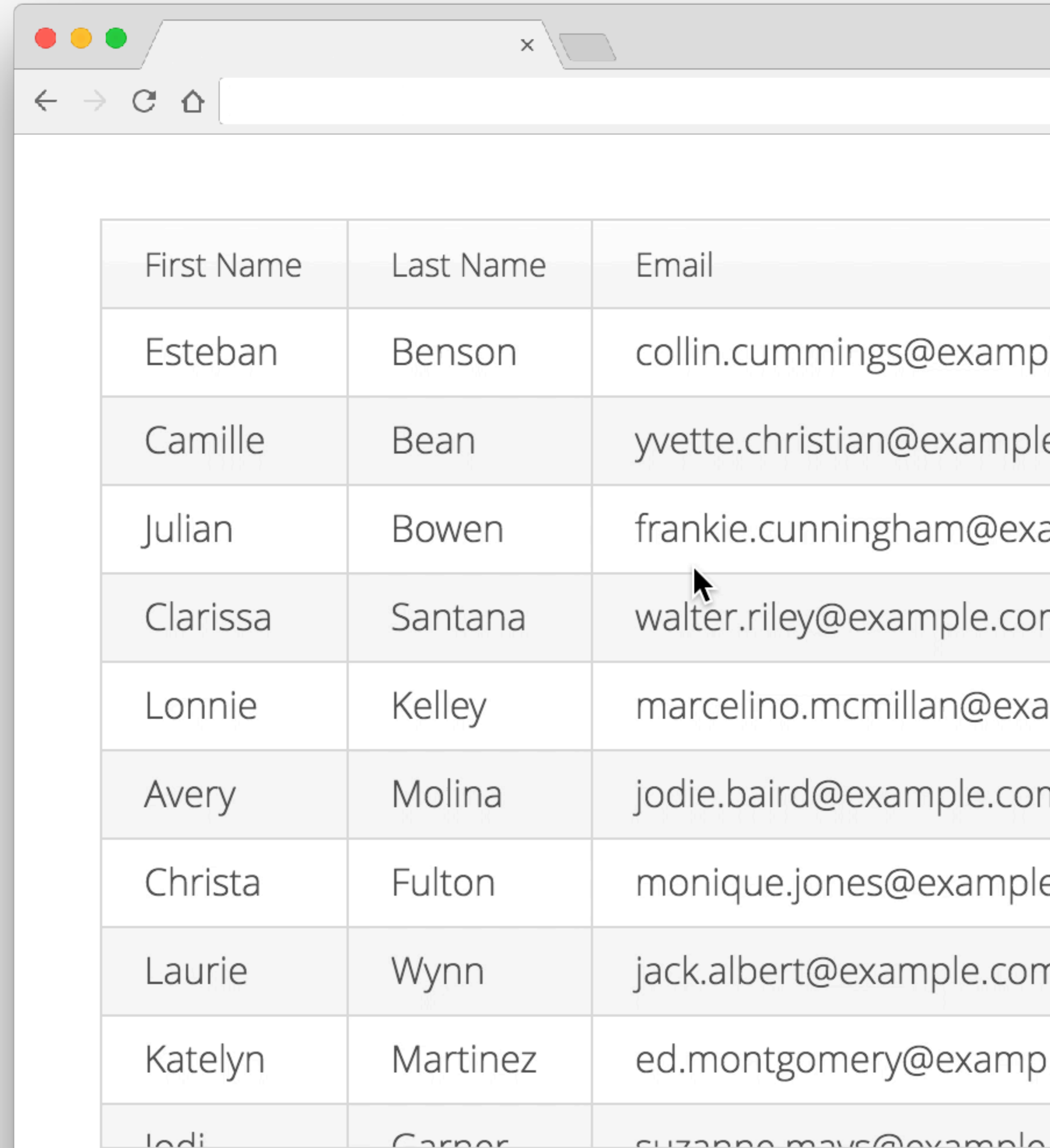
Grid<Person> grid = new Grid<>();

grid.addColumn("First name",
Person::getFirstName);

grid.addColumn("Last name",
Person::getLastName);

grid.addColumn("Email",
Person::getEmail);

grid.setItems(service.getPeople());



A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fourth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

```
@Autowired
```

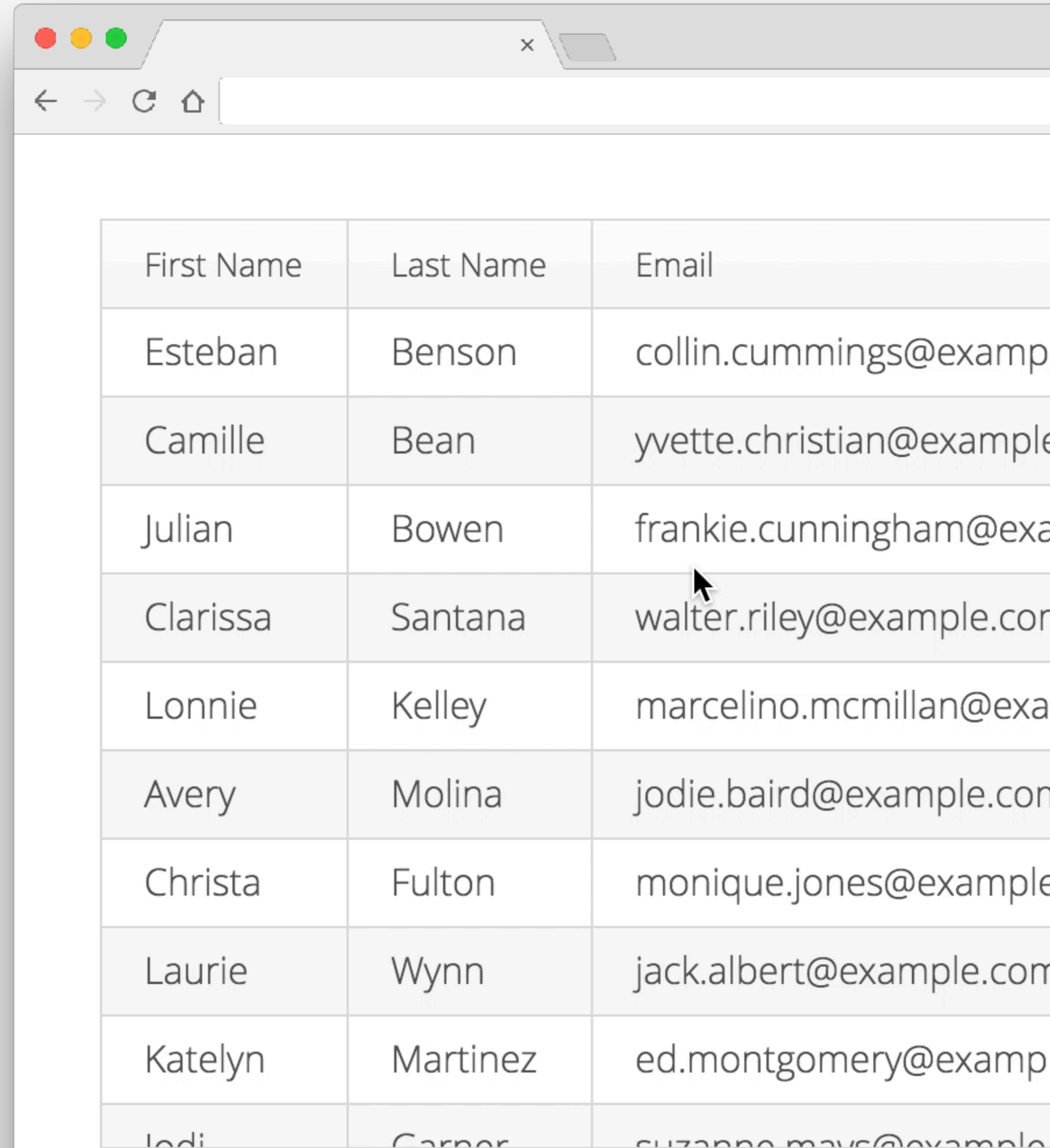
```
private PersonService service;
```

```
Grid<Person> grid = new Grid<>();
```

```
grid.addColumn("First name",  
               Person::getFirstName);
```

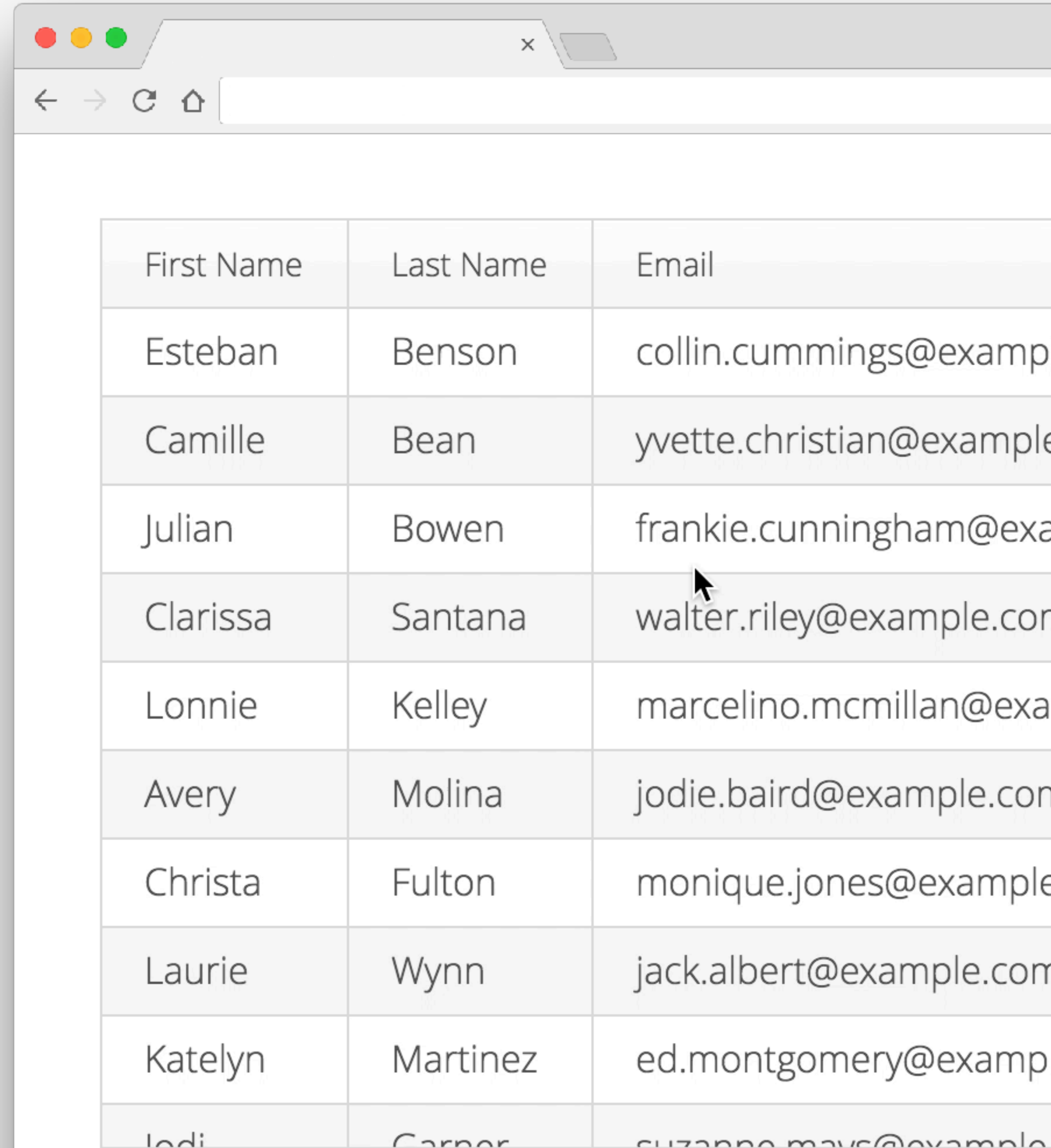
```
grid.addColumn("Last name",  
               Person::getLastName);
```

```
grid.addColumn("Email",  
               Person::getEmail);
```

A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fourth row.

First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

```
grid.addColumn("First name",  
               Person::getFirstName);  
grid.addColumn("Last name",  
               Person::getLastName);  
grid.addColumn("Email",  
               Person::getEmail);
```

A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fifth row.

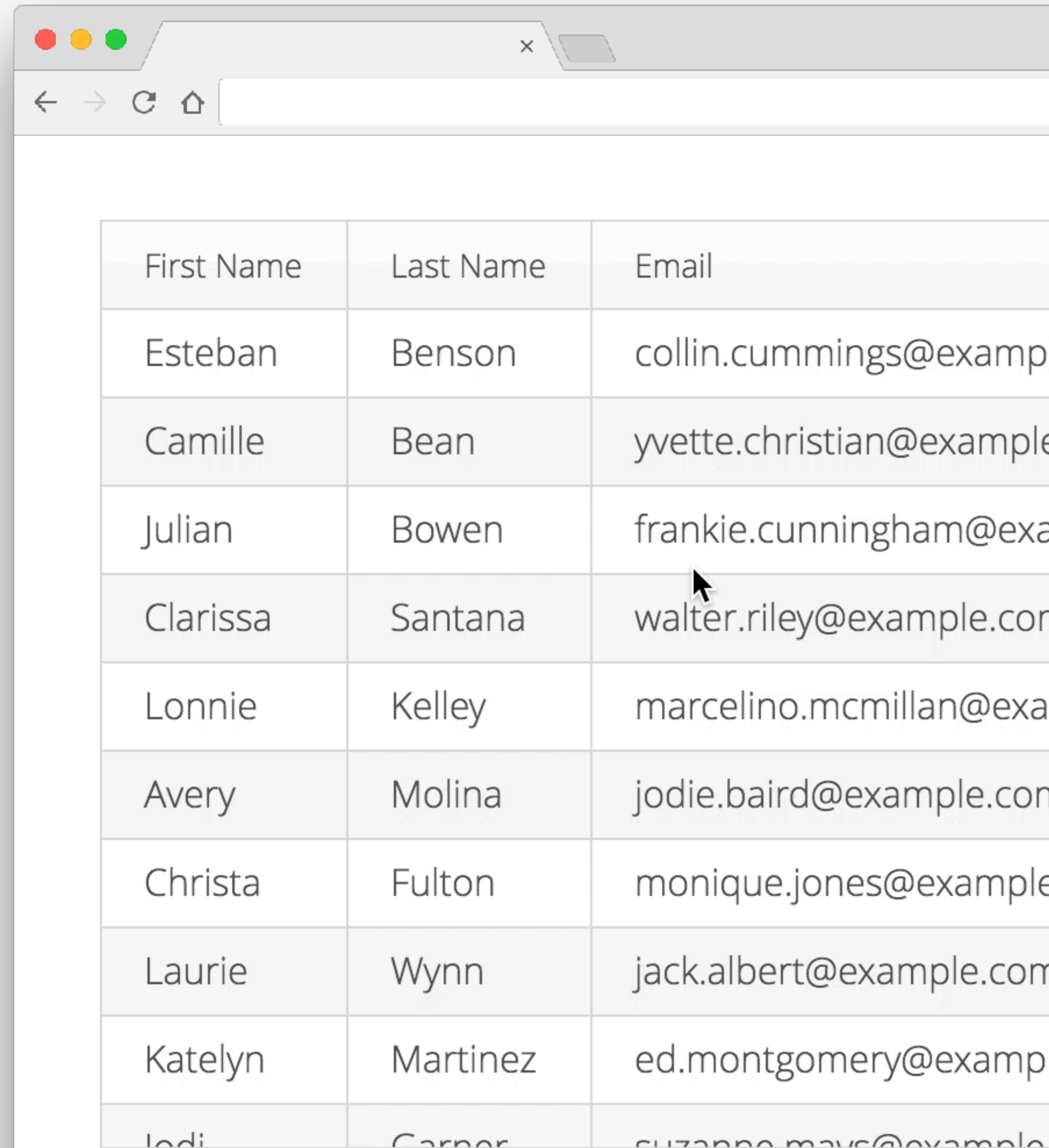
First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example


```

grid.addColumn("First name",
               Person::getFirstName);
grid.addColumn("Last name",
               Person::getLastName);
grid.addColumn("Email",
               Person::getEmail);

grid.setDataSource(
    new BackendDataSource<>(
        q -> service.getPeople(
            q.getOffset(),
            q.getLimit())
            .stream(),
        q -> service.count()));

```



A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains 12 rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fifth row.

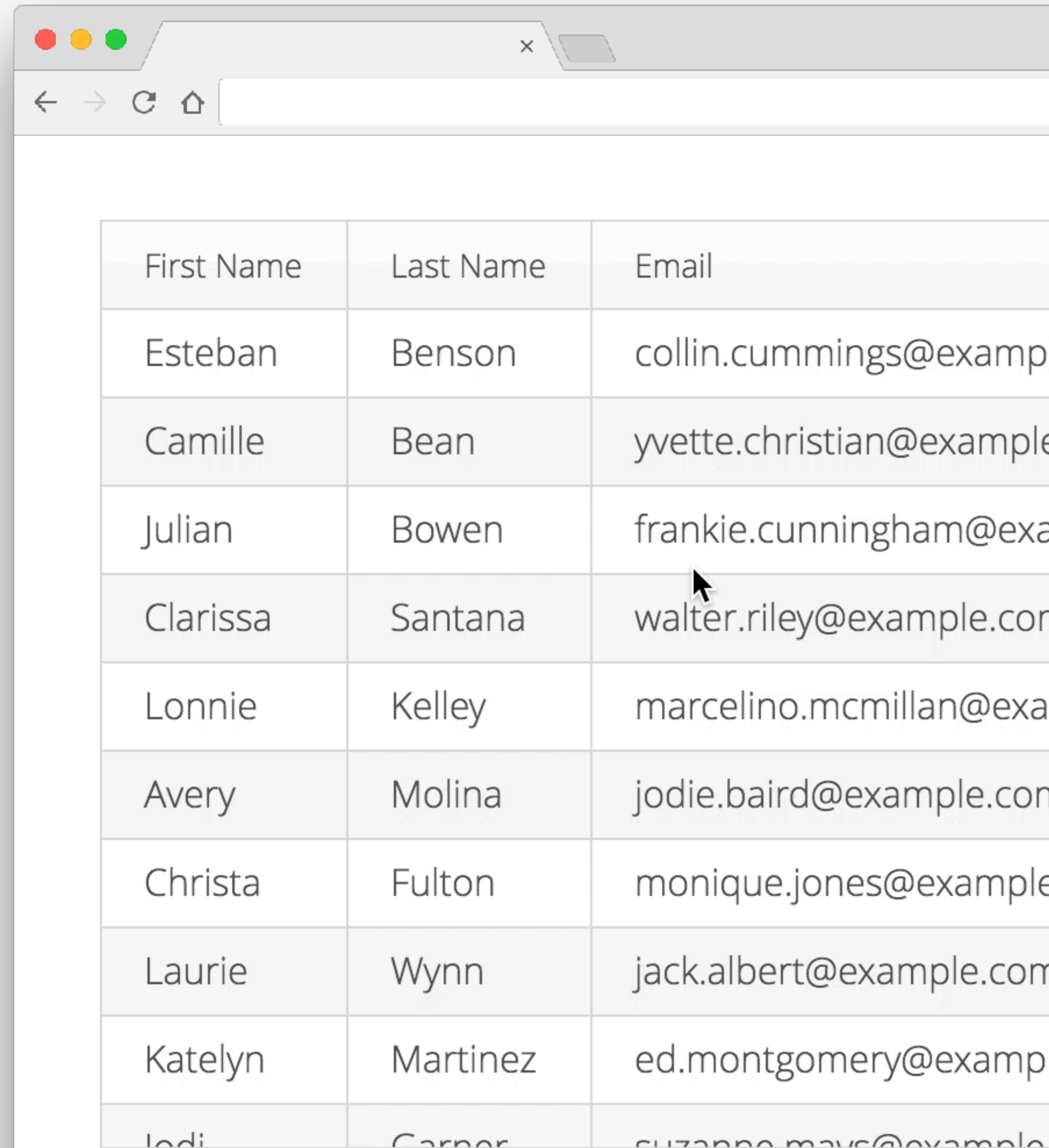
First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example


```

grid.addColumn("First name",
               Person::getFirstName);
grid.addColumn("Last name",
               Person::getLastName);
grid.addColumn("Email",
               Person::getEmail);

grid.setDataSource(
    new BackEndDataSource<>(
        q -> service.getPeople(
            q.getOffset(),
            q.getLimit())
            .stream(),
        q -> service.count()));

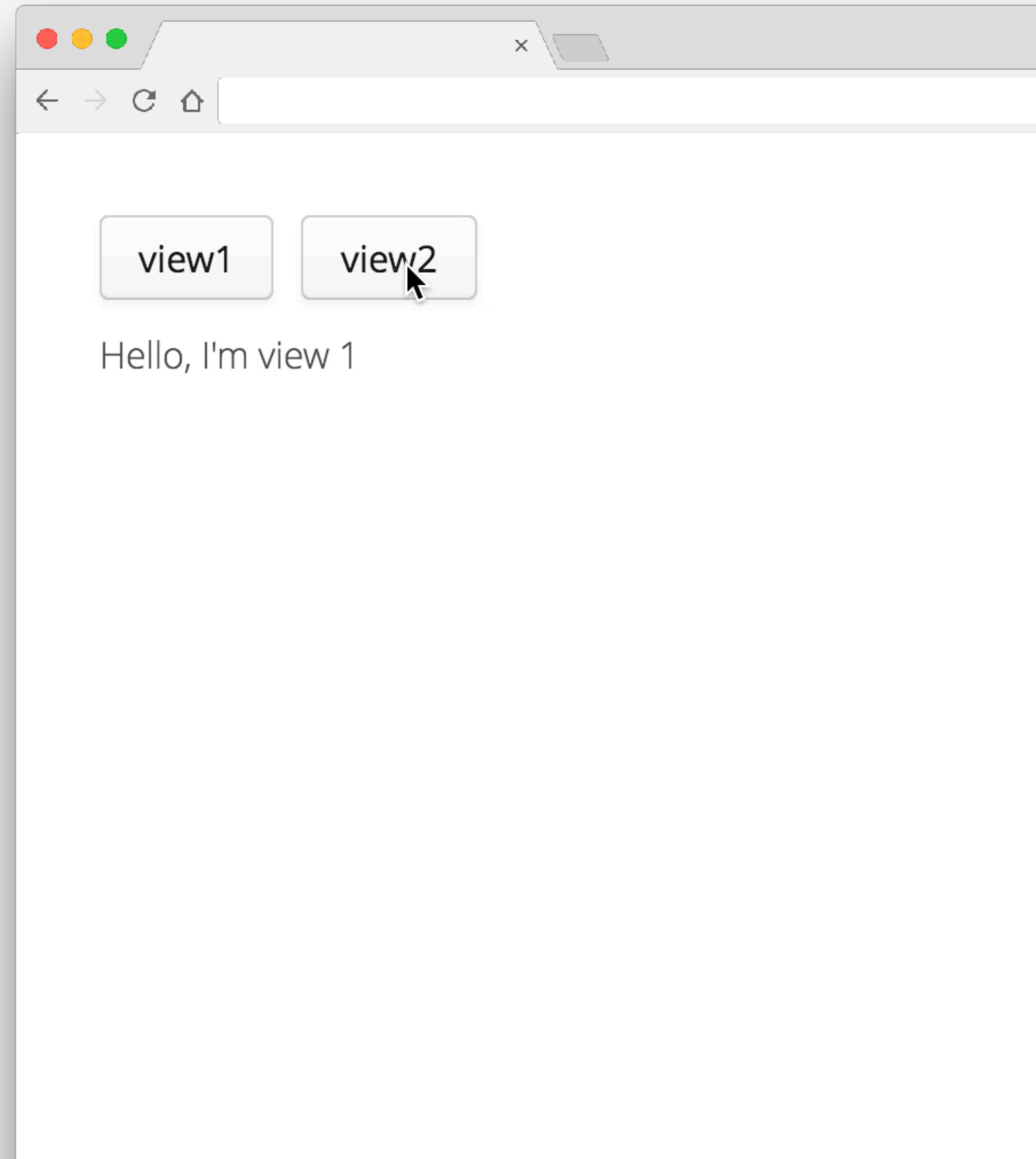
```



A screenshot of a web browser window showing a table with three columns: First Name, Last Name, and Email. The table contains ten rows of data. A mouse cursor is hovering over the email address 'walter.riley@example.com' in the fifth row.

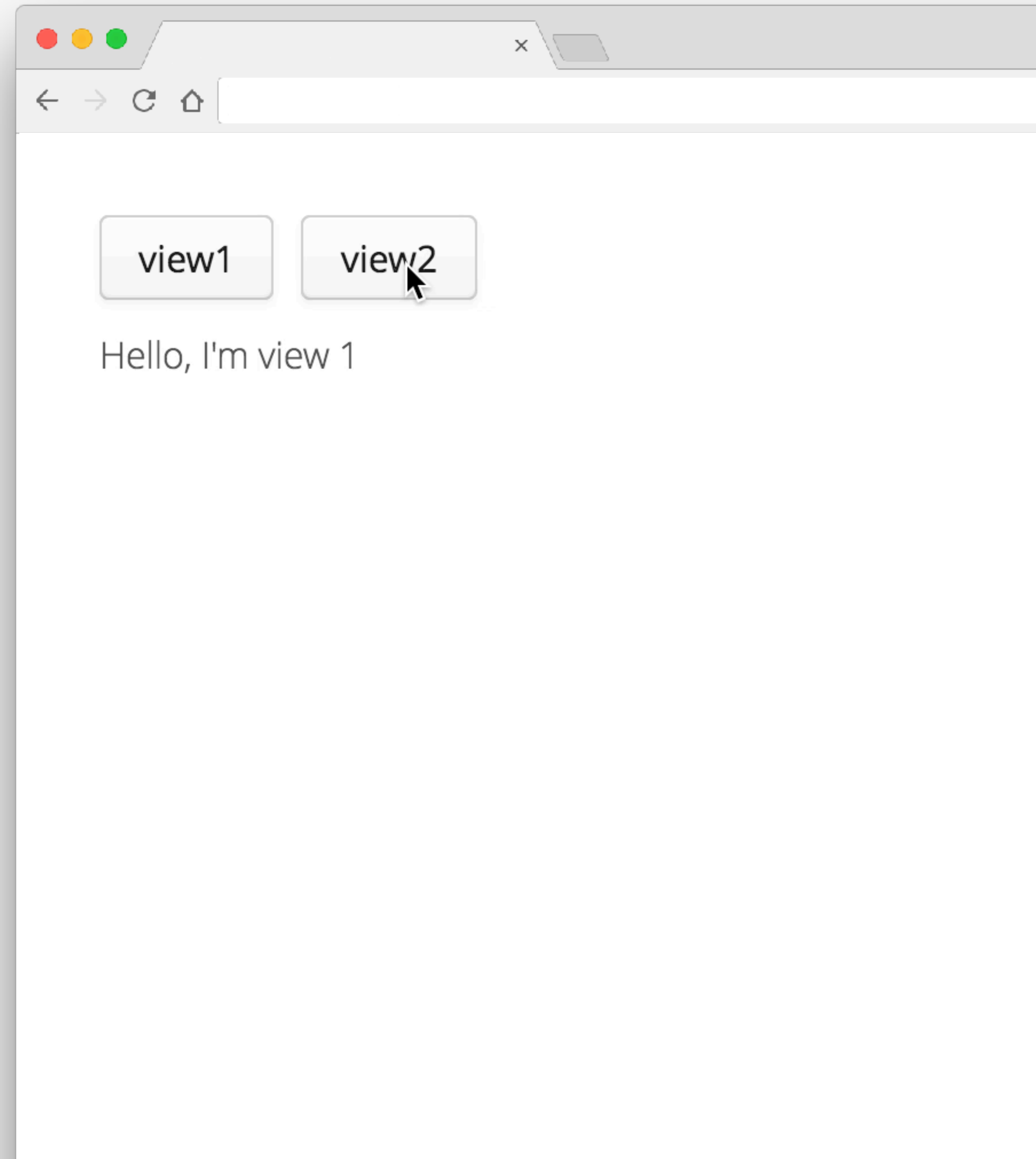
First Name	Last Name	Email
Esteban	Benson	collin.cummings@examp
Camille	Bean	yvette.christian@example
Julian	Bowen	frankie.cunningham@exa
Clarissa	Santana	walter.riley@example.com
Lonnie	Kelley	marcelino.mcmillan@exa
Avery	Molina	jodie.baird@example.com
Christa	Fulton	monique.jones@example
Laurie	Wynn	jack.albert@example.com
Katelyn	Martinez	ed.montgomery@examp
Jodi	Carper	suzanne.mays@example

```
navBar = new HorizontalLayout();
navBar.addComponent(
    new Button("view1",
        click -> nav.navigateTo("")));
navBar.addComponent(
    new Button("view2",
        click -> nav.navigateTo("view2")));
```



```
navBar = new HorizontalLayout();  
navBar.addComponent(  
    new Button("view1",  
        click -> nav.navigateTo(""))));  
navBar.addComponent(  
    new Button("view2",  
        click -> nav.navigateTo("view2")));
```

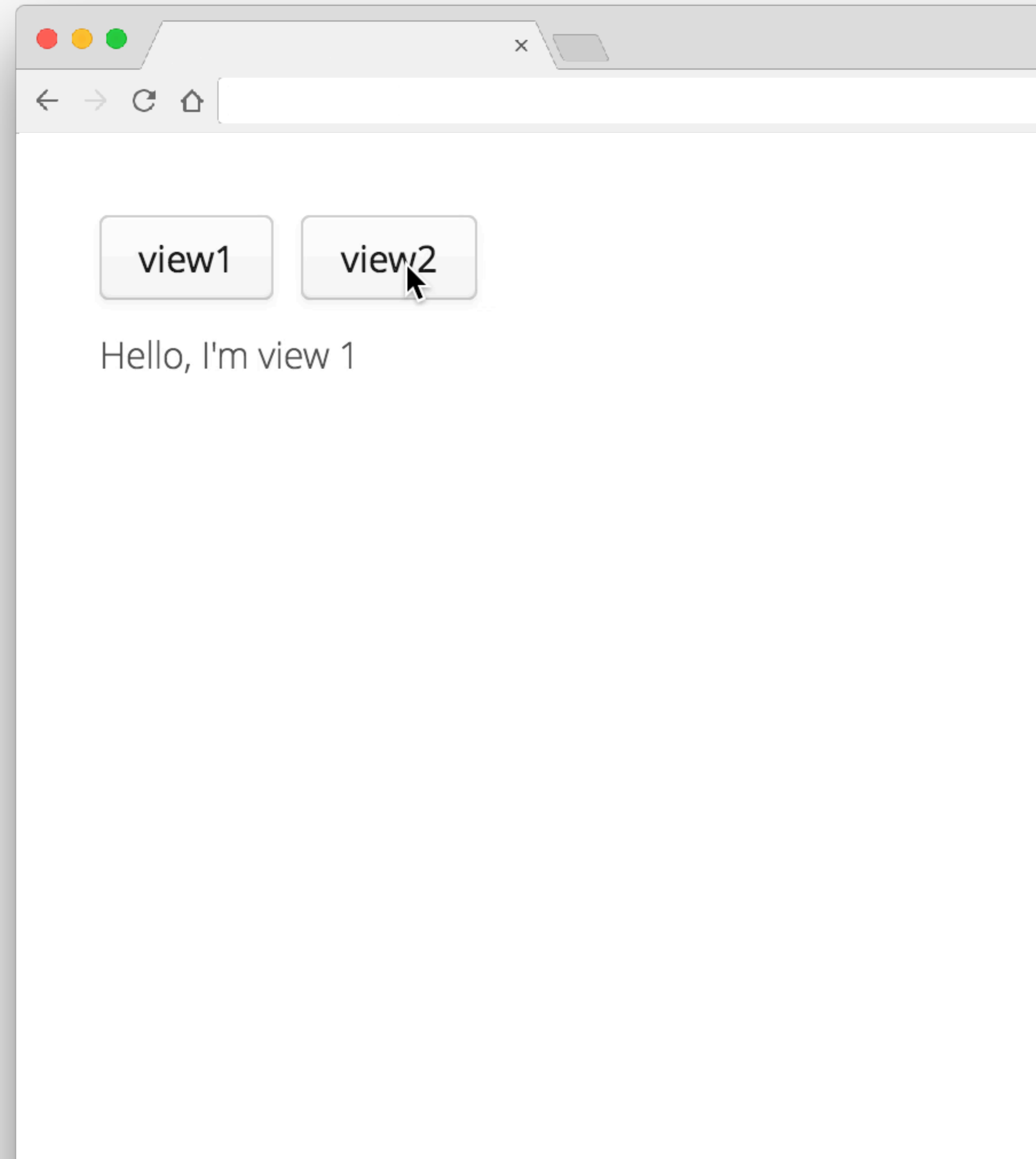
```
CssLayout navOutlet = new CssLayout();  
addComponents(navBar, navOutlet);
```



```
navBar = new HorizontalLayout();
navBar.addComponent(
    new Button("view1",
        click -> nav.navigateTo("")));
navBar.addComponent(
    new Button("view2",
        click -> nav.navigateTo("view2")));
```

```
CssLayout navOutlet = new CssLayout();
addComponents(navBar, navOutlet);
```

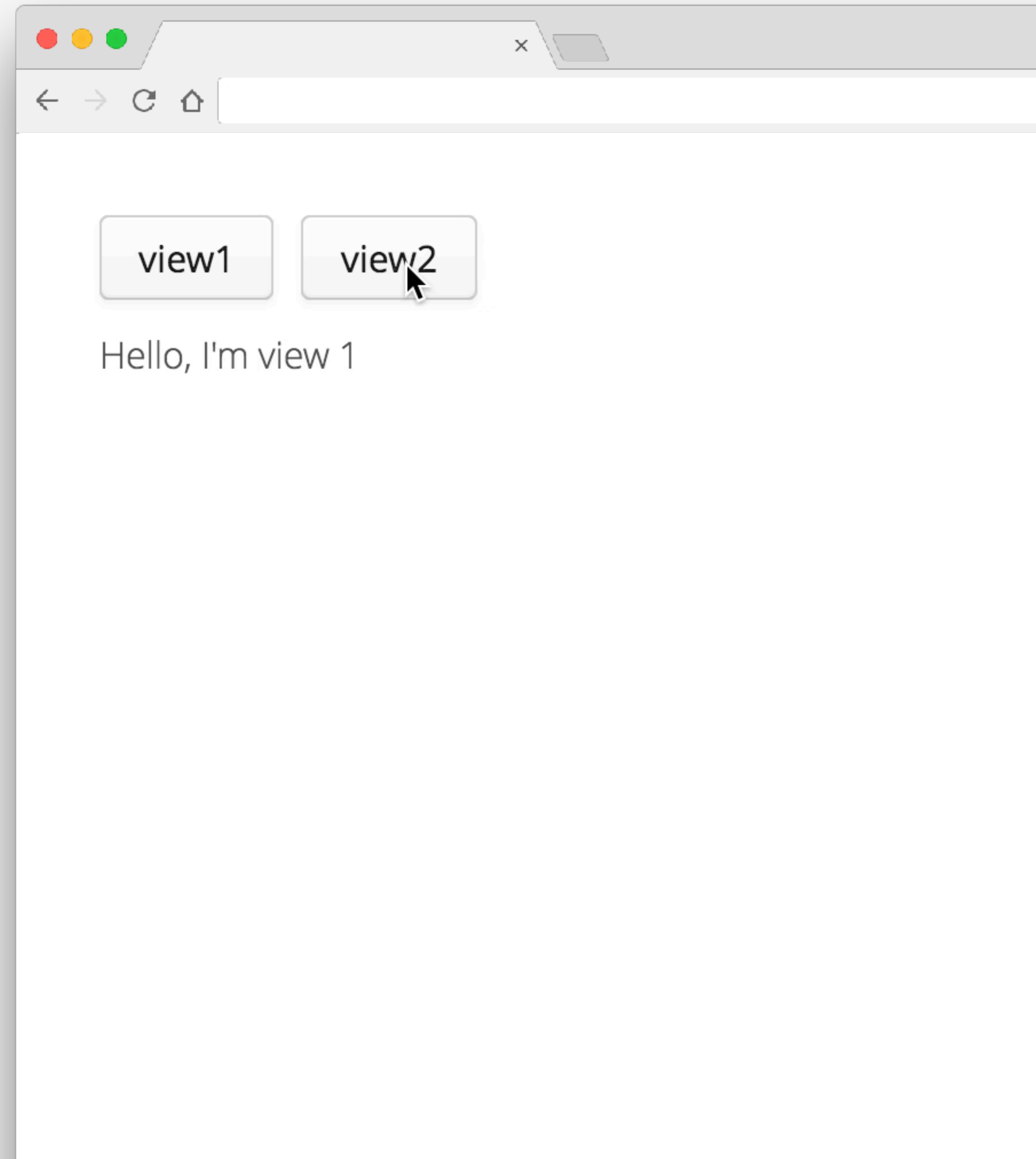
```
nav = new Navigator(this, navOutlet);
nav.addView("",
    new DemoView("Hello, I'm view 1"));
nav.addView("view2",
    new DemoView("Hello, I'm view 2"));
```



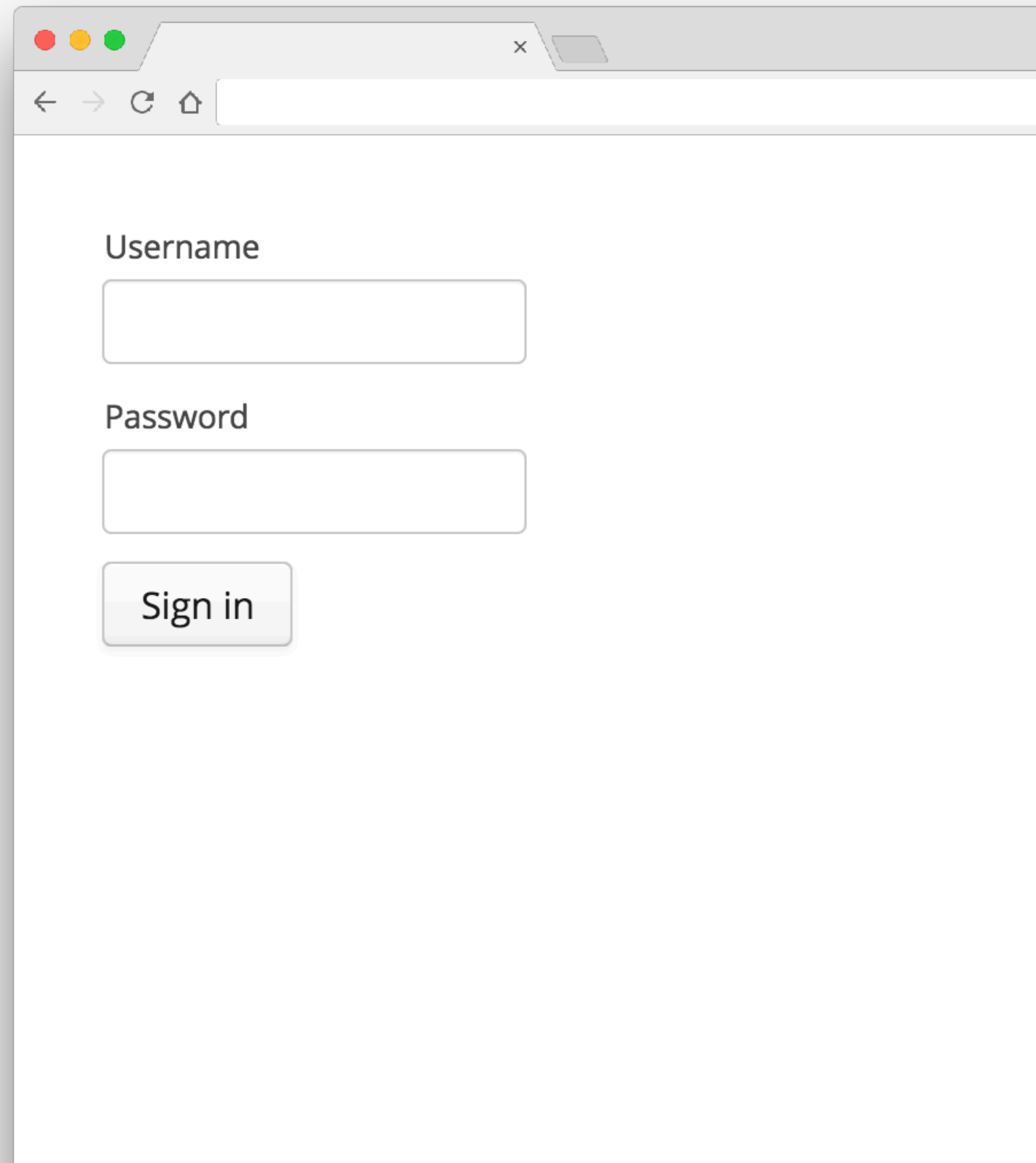
```
navBar = new HorizontalLayout();
navBar.addComponent(
    new Button("view1",
        click -> nav.navigateTo("")));
navBar.addComponent(
    new Button("view2",
        click -> nav.navigateTo("view2")));
```

```
CssLayout navOutlet = new CssLayout();
addComponents(navBar, navOutlet);
```

```
nav = new Navigator(this, navOutlet);
nav.addView("",
    new DemoView("Hello, I'm view 1"));
nav.addView("view2",
    new DemoView("Hello, I'm view 2"));
```



```
$v-background-color:  
  hsl(210, 0%, 100%);
```



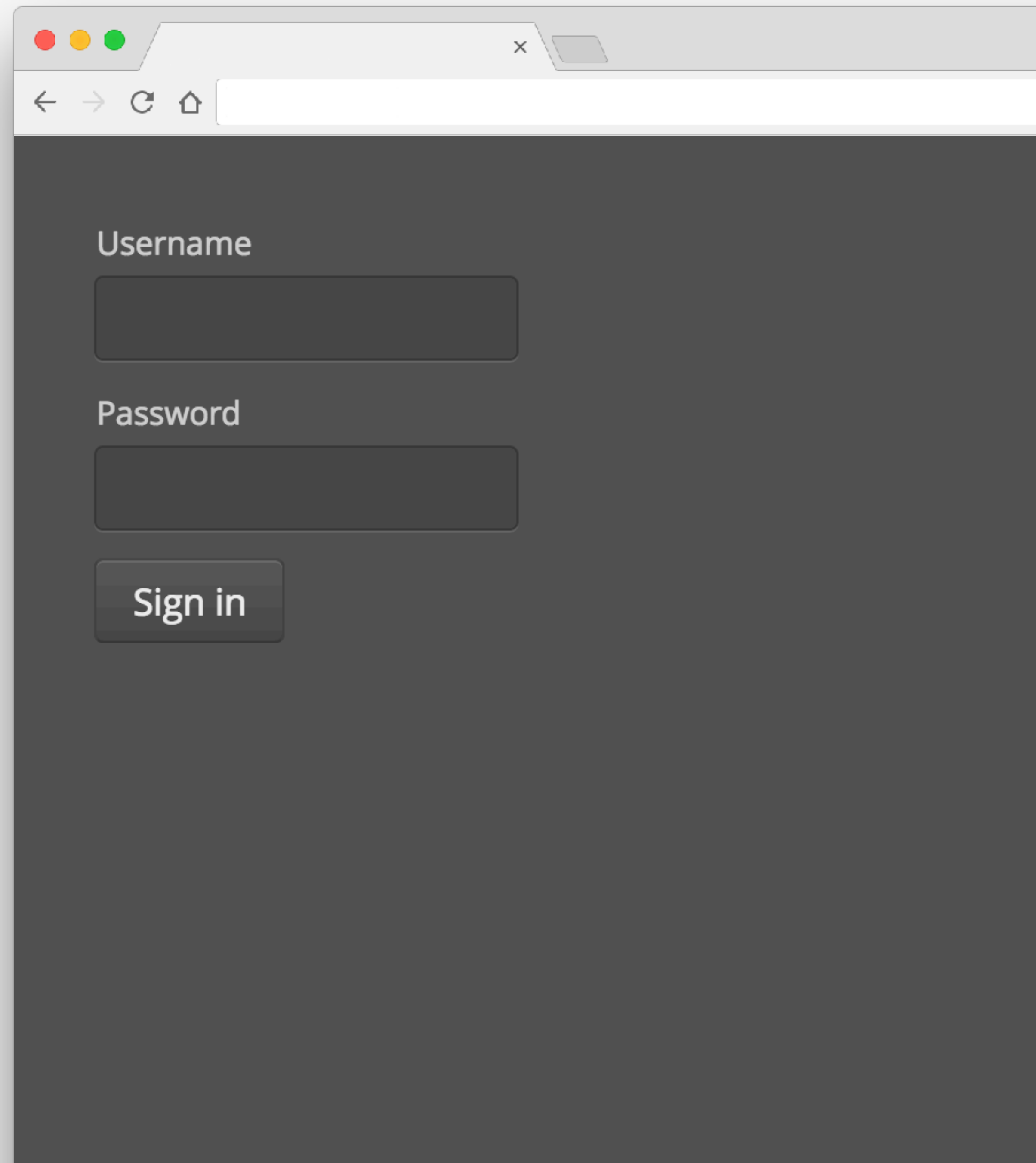
A browser window with a single tab. The address bar is empty. The page content includes a 'Username' label above a text input field, a 'Password' label above another text input field, and a 'Sign in' button below them.

Username

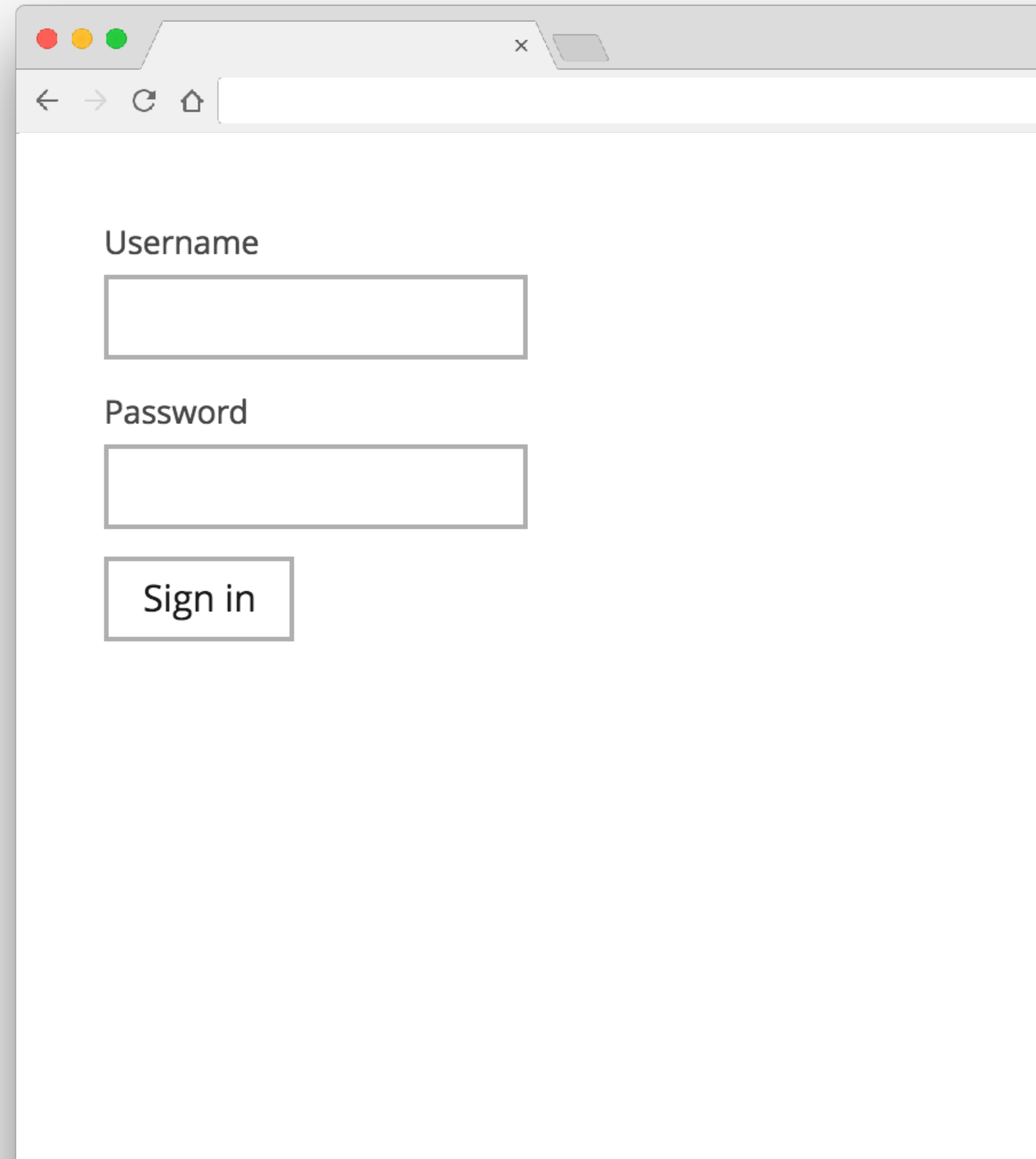
Password

Sign in

```
$v-background-color:  
  hsl(210, 0%, 33%);
```



```
$v-background-color:  
  hsl(210, 0%, 100%);  
$v-border: 2px solid v-shade;  
$v-border-radius: 0;  
$v-bevel: false;  
$v-gradient: false;  
$v-shadow: false;
```



TODO



- ☒ Prepare presentation for Meetup
- ☒ Go to meetup
- ☐ Have presentation at Meetup

Delete selected

SPRING INITIALIZR

bootstrap your application now

Generate a Maven Project ▾ with Spring Boot 1.4.2 ▾

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Vaadin ✕

JPA ✕

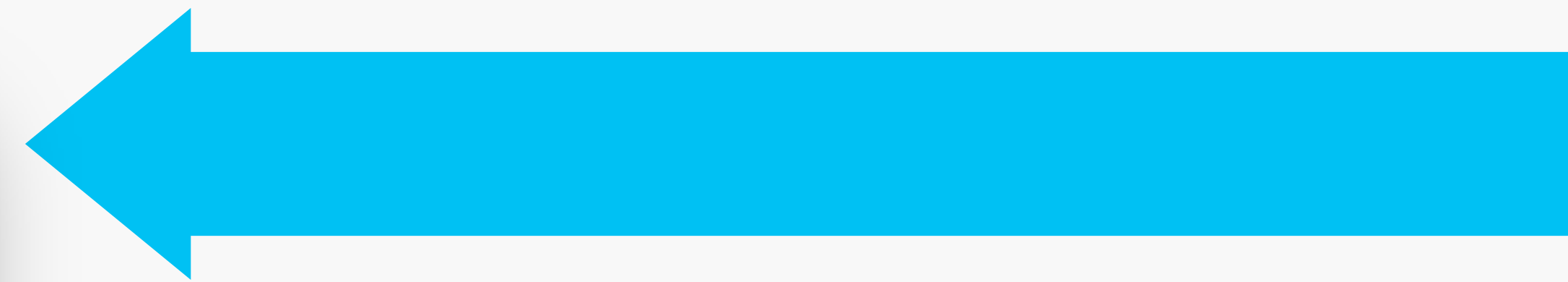
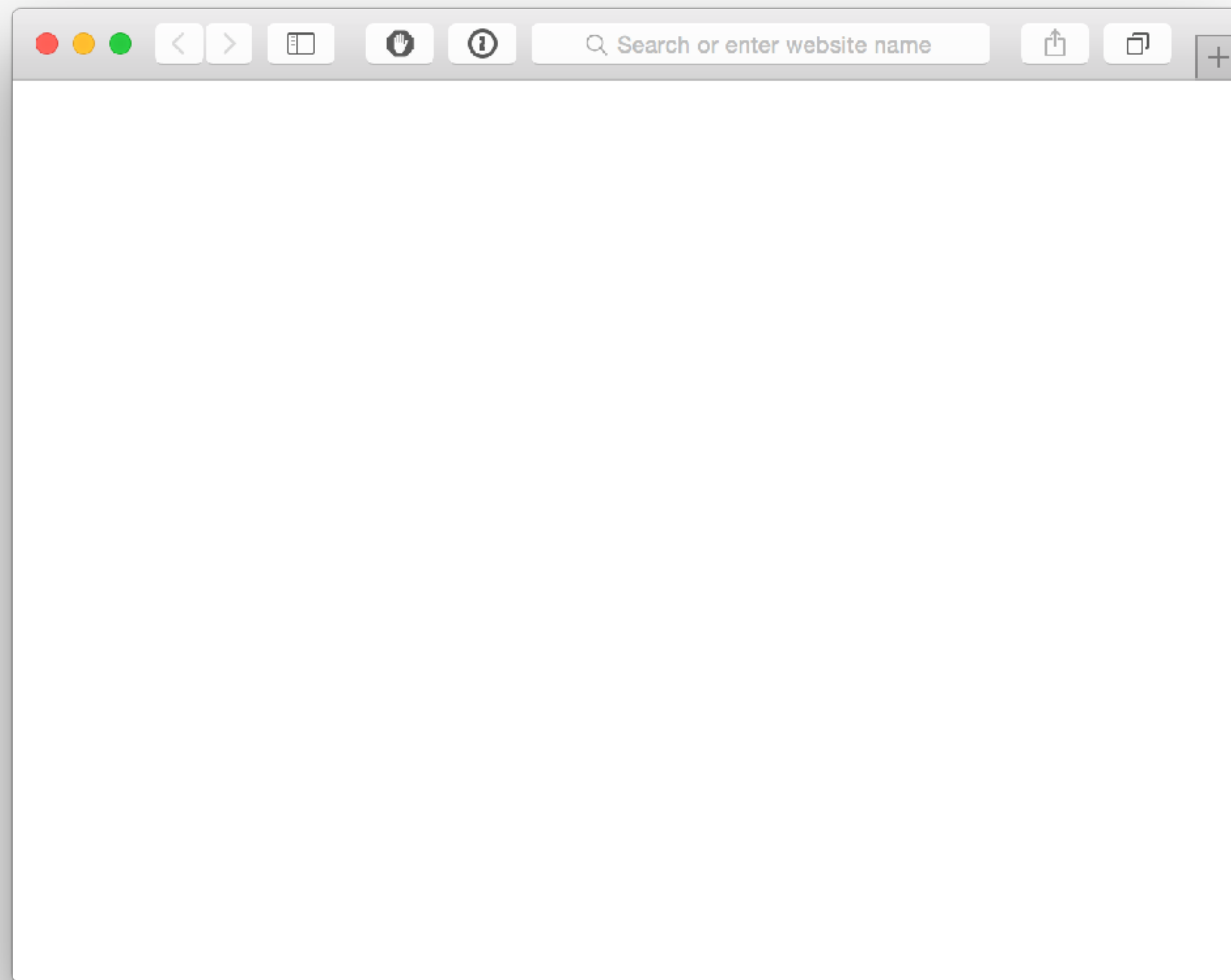
H2 ✕

Generate Project ⌘ + ↵

Don't know what to look for? Want more options? [Switch to the full version.](#)

 github.com/vaadin-marcus/spring-boot-todo

```
TextField name = new TextField();  
Button button = new Button("Greet");  
layout.addComponent(name, button);  
  
button.addClickListener(click -> {  
    Notification.show("Hi, " + name.getValue());  
});
```

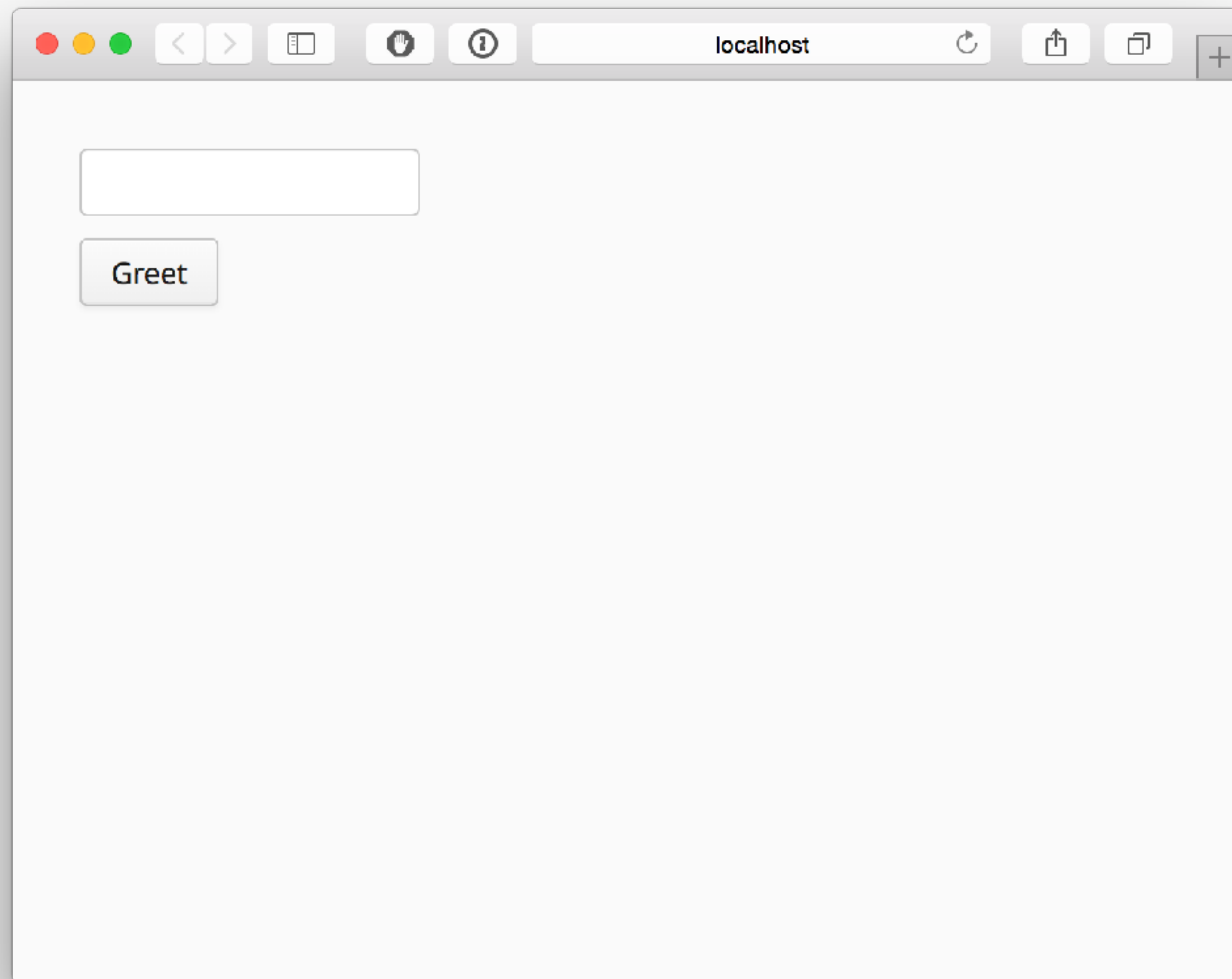


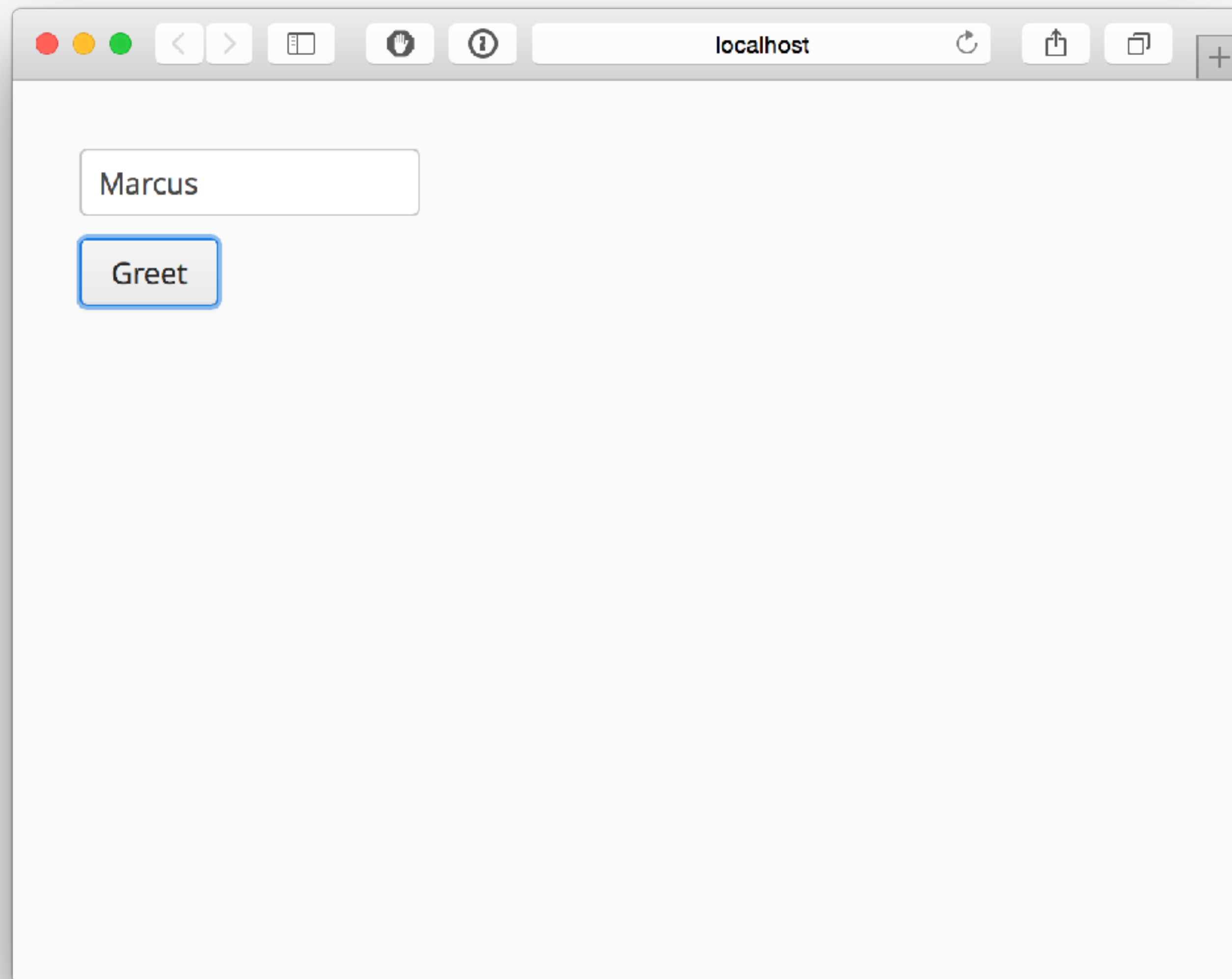
Initial HTML

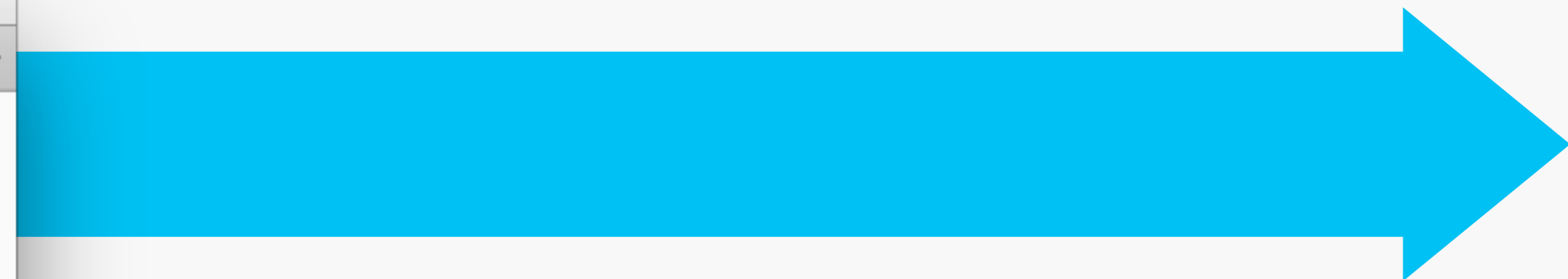
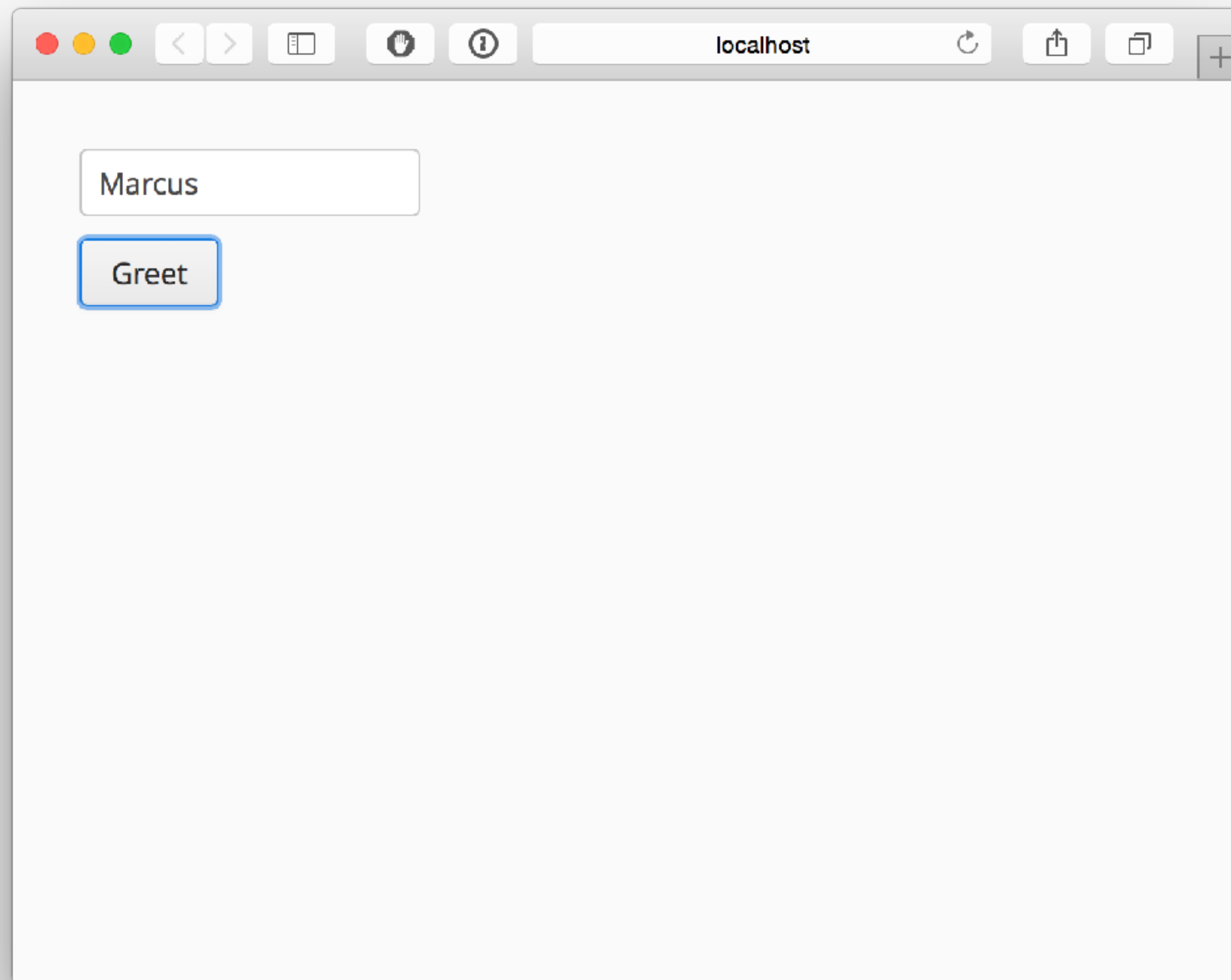
CSS

JavaScript

~300k







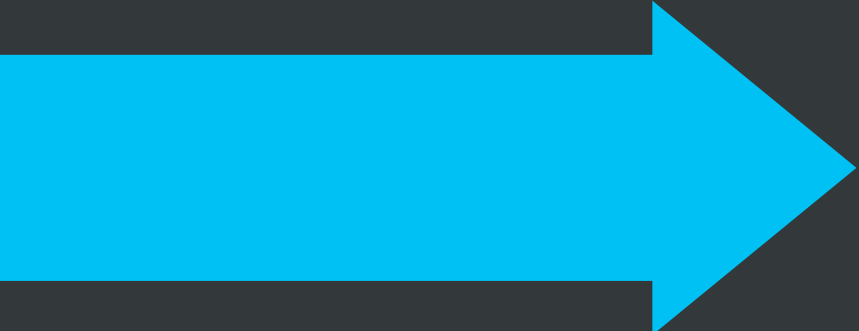
```
{name: {value: 'Marcus'},  
button: {event: 'clicked'}}
```

261 bytes

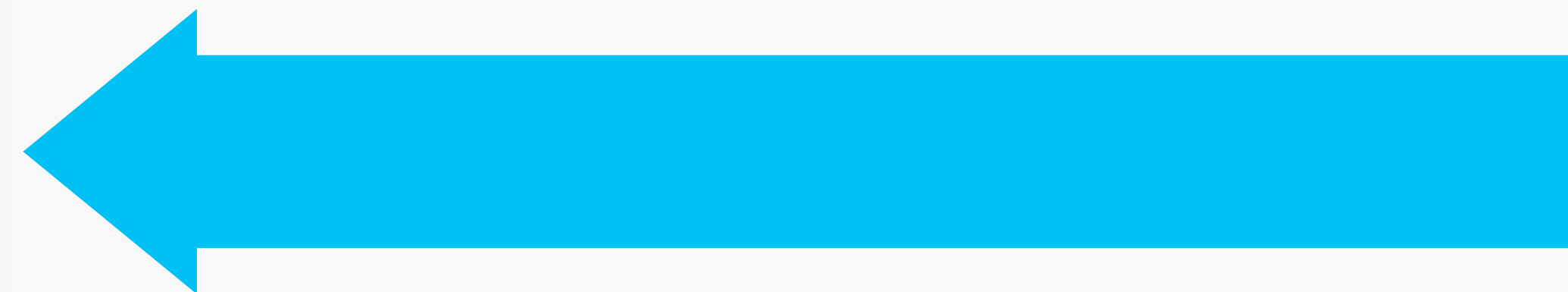
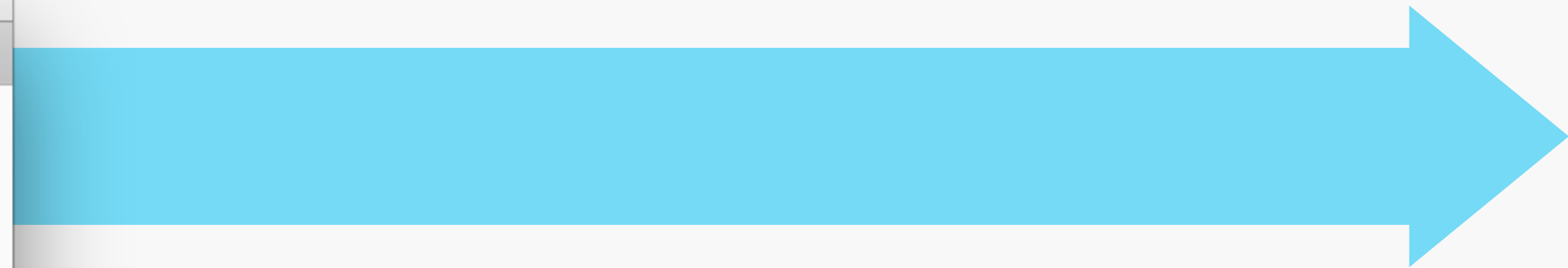
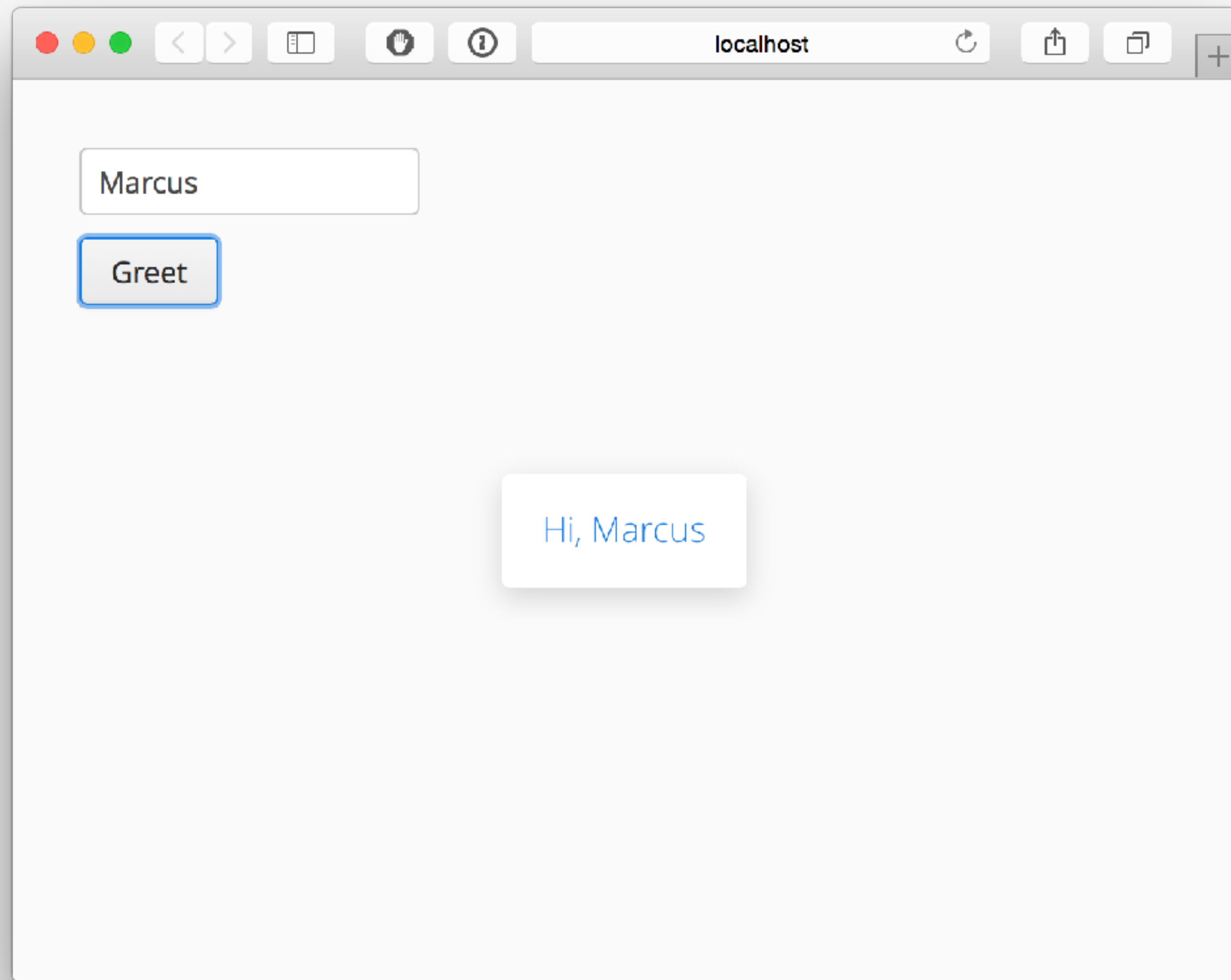
```
TextField name = new TextField();  
Button button = new Button("Greet");  
layout.addComponent(name, button);
```

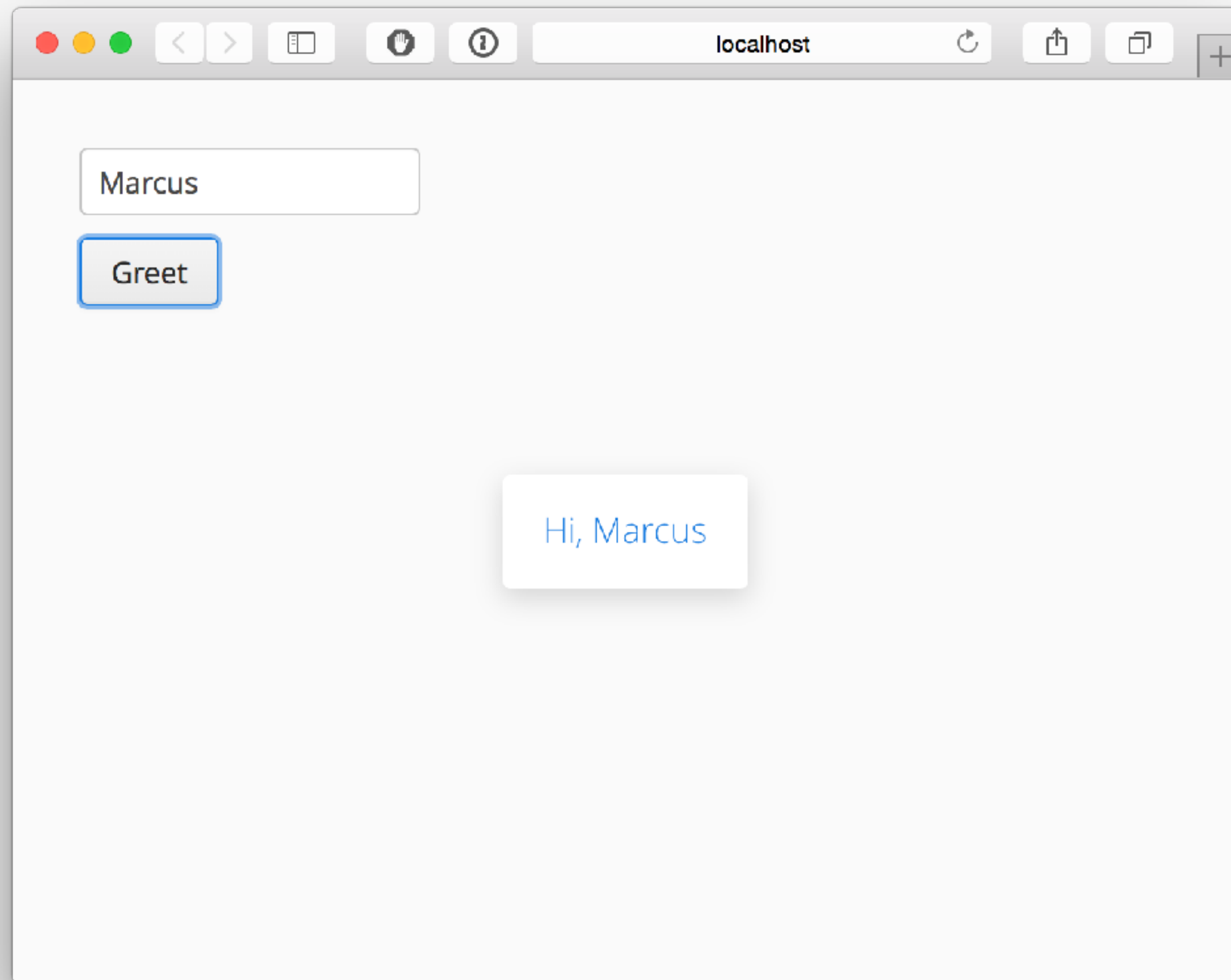
```
button.addClickListener(click -> {  
    Notification.show("Hi, " + name.getValue());  
});
```

```
TextField name = new TextField();  
Button button = new Button("Greet");  
layout.addComponents(name, button);
```



```
button.addClickListener(click -> {  
    Notification.show("Hi, " + name.getValue());  
});
```





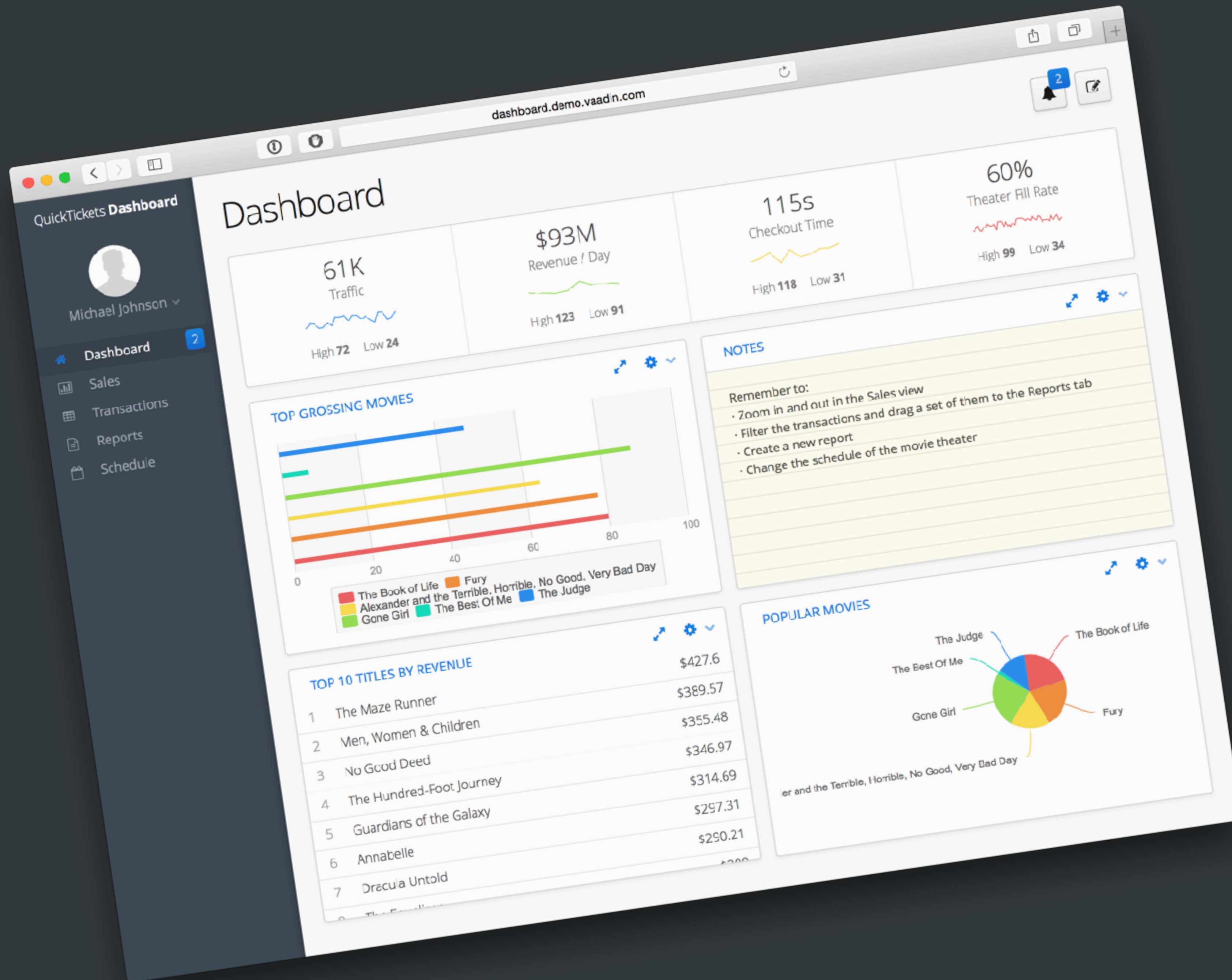
```
{name: {value: 'Marcus'},  
button: {event: 'clicked'}}
```

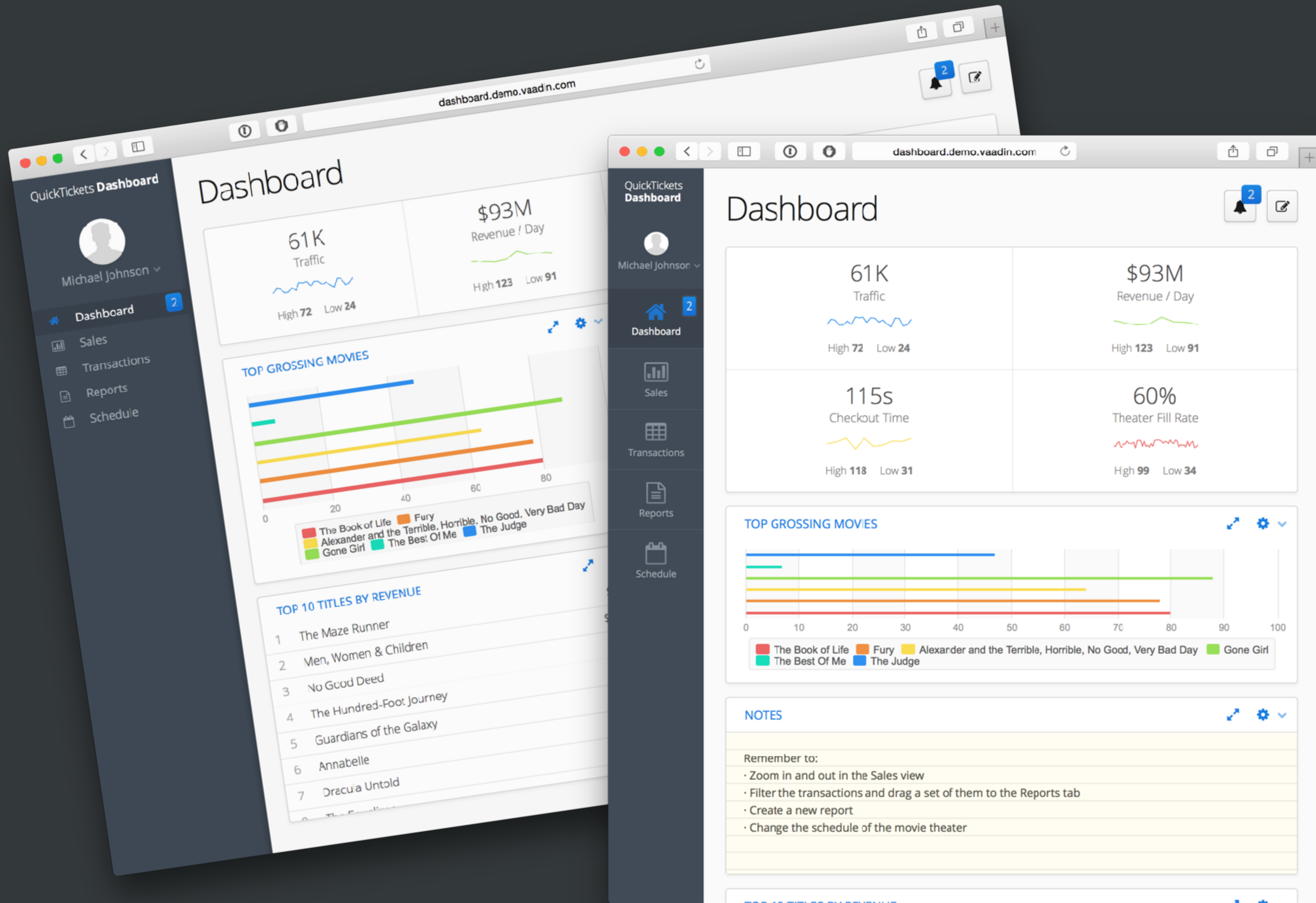
261 bytes

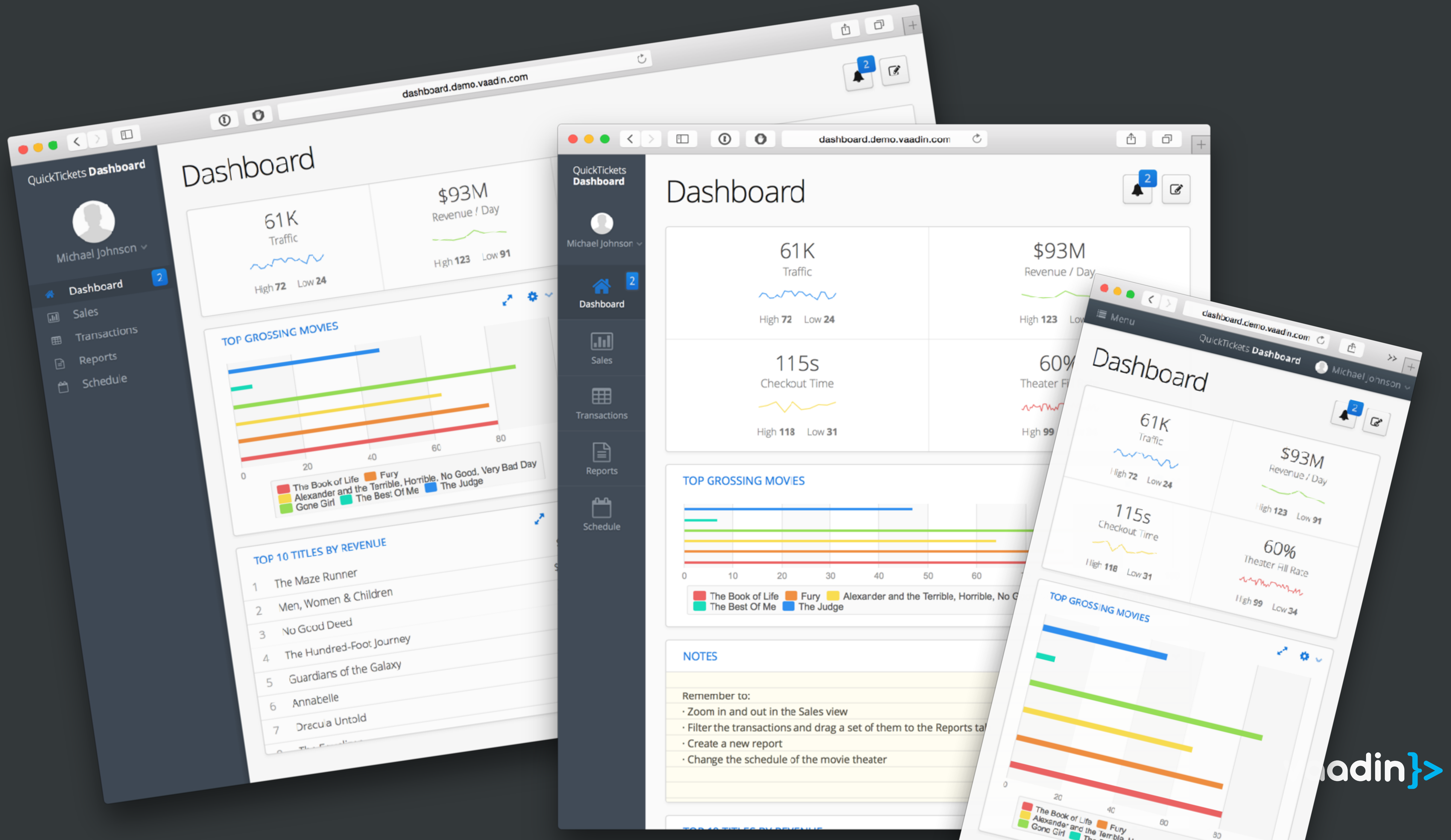


```
{new: [  
  {notification: 'Hi, Marcus'}  
]}
```

267 bytes








```
.v-ui[width-range~="0-800px"]{  
    /* Styles for small devices */  
}  
.v-ui[width-range~="801px-"]{  
    /* Styles for large devices */  
}
```

```
.v-ui[width-range~="0-800px"]{  
    /* Styles for small devices */  
}  
.v-ui[width-range~="801px-"]{  
    /* Styles for large devices */  
}
```

```
getPage().addBrowserWindowResizeListener( newSize -> {  
    if (newSize.getWidth() > LARGE_DEVICE_WIDTH){  
        setupLargeLayout();  
    } else {  
        setupSmallLayout();  
    }  
});
```

WEBSOCKETS

```
public class VaadinUI extends UI {  
    ...  
}
```

WEBSOCKETS

```
@Push
public class VaadinUI extends UI {
    ...
}
```

WEBSOCKETS

```
@Push
public class VaadinUI extends UI {
    ...
}

ui.access(() -> Notification.show("Alert!"));
```





 STRONG SECURITY

♥ THE JVM

 EXTENDABLE



Framework Elements Community Services Pro Tools Company



Marcus Hellberg



Demo Get started Tutorial Docs Add-ons Comparison

Add-ons

Expand your possibilities
with add-on components



Vaadin Spreadsheet

Let users create, edit, link and share Excel documents online.

Search

Top lists

My Add-ons



Feedback


☒ UI Components ☒ Data Components ☒ Themes ☒ Tools ☒ Miscellaneous

All Vaadin versions


A-Z Top Downloads Highest Rated **Most Recent**

Search results (643 add-ons found)

UI Components

 **cdi-helpers**
by [Matti Tahvonen](#)
Various helpers for apps using Vaadin CDI
★★★★★

UI Components

 **Crud UI Add-on**
by [Alejandro Duarte](#)
Generate CRUD UIs for your entities/beans/POJOs at runtime
★★★★★

UI Components

 **g-leaflet-rotatedmarker**
by [Octavian Marcu](#)
g-leaflet-rotatedmarker
★★★★★
30
0





Vaadin Spreadsheet

[Home page](#) · [JavaDoc](#) · [Manual](#)

Basic functionality

Edit imported Excel file with formatting, basic formulas, and a chart. Updates dynamically when values are edited.

Collaborative features

Freeze panes, protected cells and add comments

Formatting

Style your spreadsheet

Grouping

Use the Excel feature for grouping rows and columns

Report mode

Use the read only mode of spreadsheet

Simple invoice

Use the spreadsheet for invoices

Upload Excel files

Upload a .xlsx or .xls file

[Demo](#)

[Java Source](#)

G5

A

B

C

D

E

F

G

H

I

J

K

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

SIMPLE MONTHLY BUDGET

MONTHLY INCOME

Item	Amount
Ash	\$1,000.00
Jamie	\$1,000.00
Dividends	\$250.00

MONTHLY EXPENSES

Item	Amount
Rent/mortgage	\$800.00
Electric	\$120.00
Gas	\$50.00
Cell phone	\$45.00
Groceries	\$500.00
Car payment	\$273.00
Auto expenses	\$120.00
Student loans	\$50.00
Credit cards	\$100.00
Auto Insurance	\$78.00
Personal care	\$50.00
Entertainment	\$100.00
Miscellaneous	\$50.00

PERCENTAGE OF INCOME SPENT

104%

SUMMARY

Total Monthly Income	Total Monthly Expenses	Balance
\$2,250	\$2,336	-\$86

2500

2000

1500

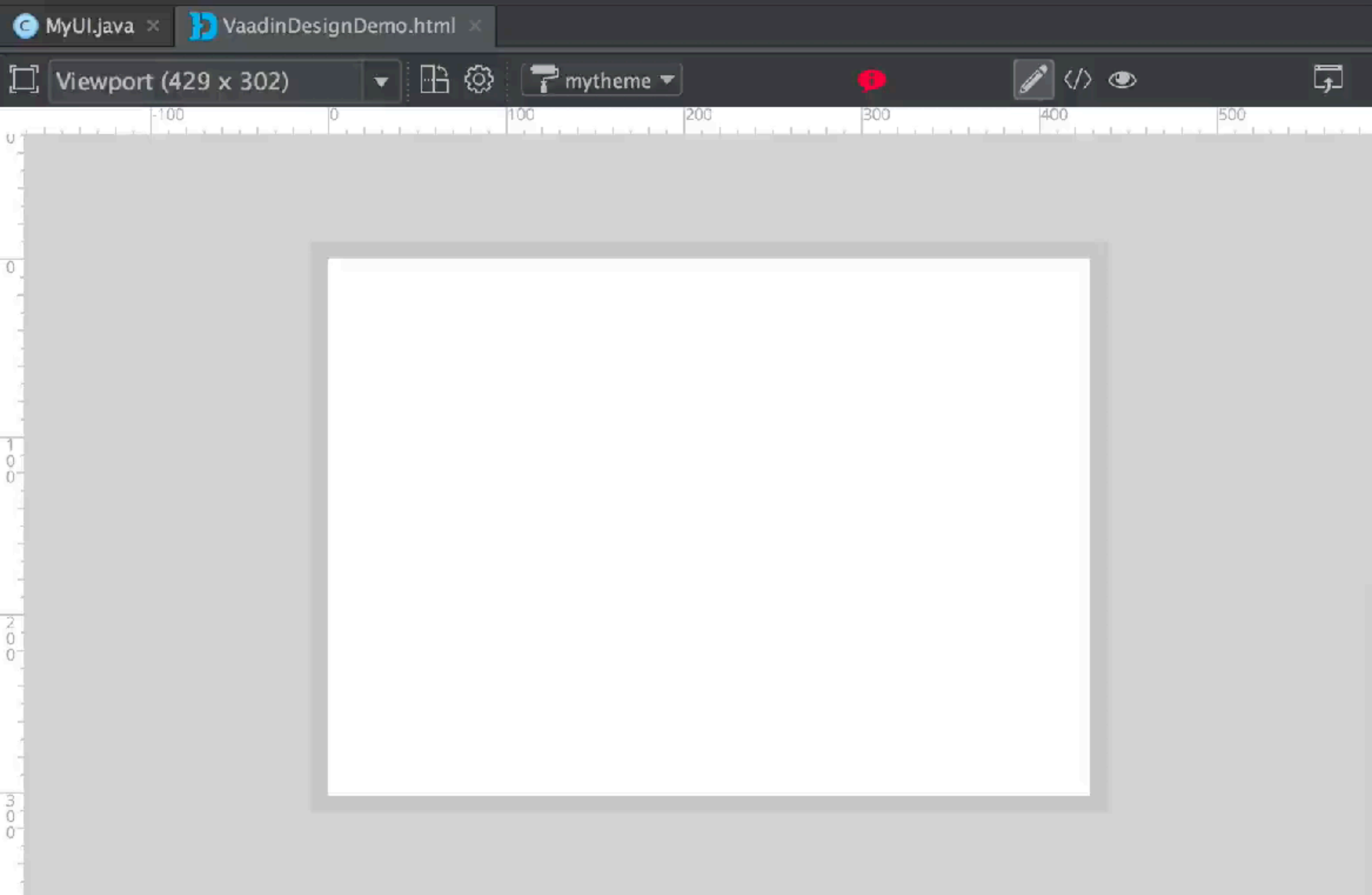
1000

500

0

Total Monthly Income

Total Monthly Expenses



```
13  
14 // Affects the color of some component elements, e.g Button, Panel title, etc  
15 $v-background-color: hsl(210, 0%, 100%);  
16 // Affects the color of content areas, e.g Panel and Window content, TextField input etc  
17 //$v-app-background-color: $v-background-color;  
18  
19 // $v-border: 2px solid v-shade;
```

Vaadin Palette

Layouts

Components

Outline

Name	Type
VaadinDesignDemo	Design
VerticalLayout	VerticalLayout

Name	Value
Nothing to show	

 JR Rebel Console



We'd be happy to help you.





THANK YOU

 @MARCUSHELLBERG