VaadinFiddle

BETA

Jonni Nakari

VAADIN EXPERT

02/26/2018

vaadin

SHARING SAMPLES ONLINE

NO TOOLS AVAILABLE

vaadin}>

SHARING SAMPLES ONLINE

In JavaScript realm

Lot of such tools exist in the JavaScript world

- JSFiddle
- CodePen
- JS Bin
- etc.

They are so easy to implement some projects have even rolled their own sample playgrounds.

SHARING SAMPLES ONLINE

In Vaadin / Java realm

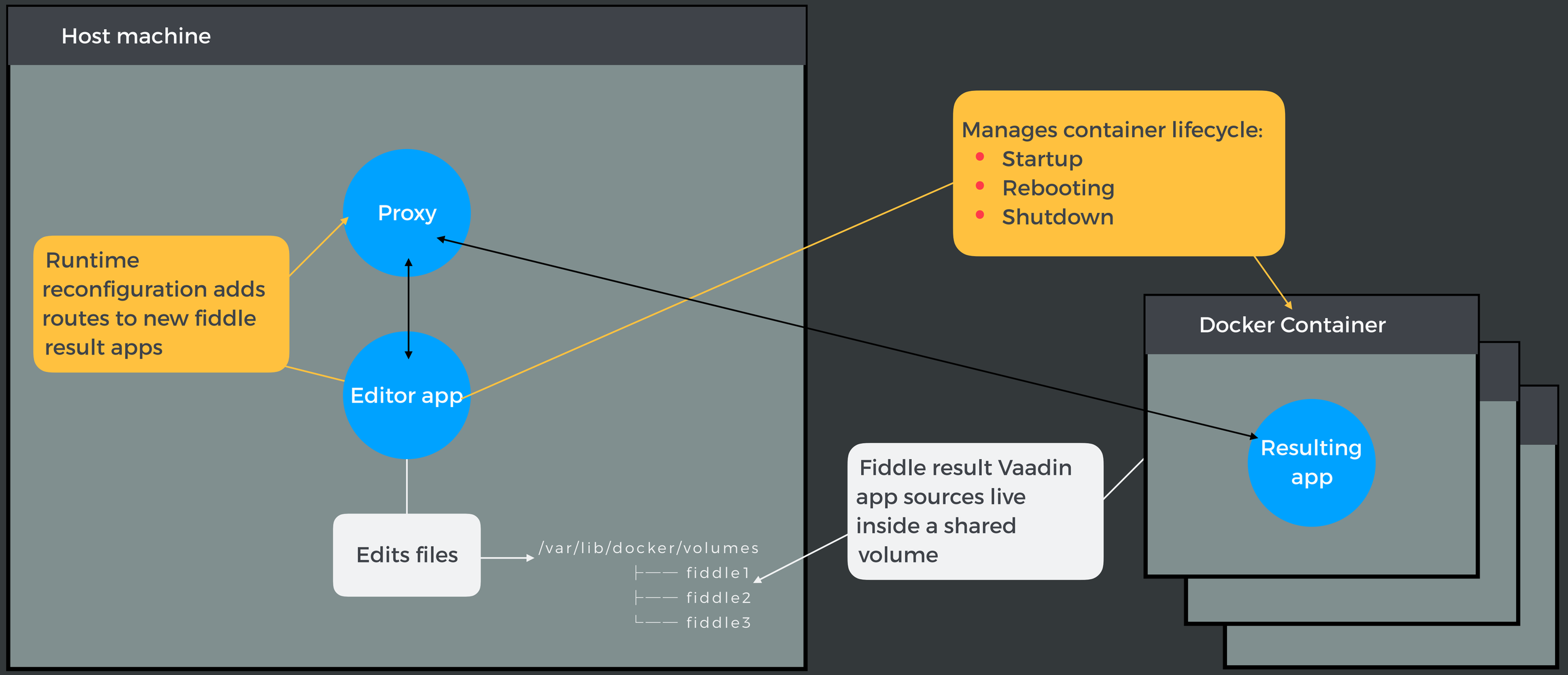
Special challenges as the server-side code needs to be edited. Giving access server-side code to random people on the internet can lead to:

- People ruining each other's samples
- Using server resources for DDOSing, mining bitcoins
- Sending spam email

SOLUTION I CAME UP WITH

NETWORK SANDBOXED DOCKER CONTAINERS TO RUN THE USER CODE

CURRENT PROTOTYPE



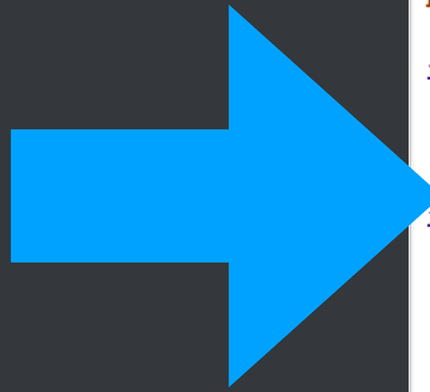
vaadin

DEMO TIME

vaadin}>

HOW WAS SLACK ABLE TO SHOW THE CODE + UI PREVIEW IMAGE?

Bootstrap page has Open Graph protocol meta tags



```
<!doctype html>
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=11">
  <style type="text/css">html, body {height:100%;margin:0;}</style>
  <link rel="shortcut icon" type="image/vnd.microsoft.icon"
href="./../../../../../../../../../VAADIN/themes/vaadin-fiddle/favicon.ico">
  <link rel="icon" type="image/vnd.microsoft.icon"
href="./../../../../../../../../../VAADIN/themes/vaadin-fiddle/favicon.ico">
  <meta property="og:url" content="https://vaadinfiddle.com/editor/container/[c
id]/src/main/java/org/vaadin/vaadinfiddle/MainView.java">
  <meta property="og:type" content="website">
  <meta property="og:title" content="MainView.java on VaadinFiddle">
  <meta property="og:image" content="https://vaadinfiddle.com/editor/preview-
image/[container id]_src_main_java_org_vaadin_vaadinfiddle_MainView.java.png">
  <meta property="og:image:width" content="1067">
  <meta property="og:image:height" content="473">
 </head>
 <body scroll="auto" class="v-generated-body">
  <div id="ROOT-2521314" class="v-app vaadin-fiddle fiddleui">
```

vaadin

Separate Servlet listening for preview image requests

```
@WebServlet("/preview-image/*")
public class PreviewImageServlet extends HttpServlet {
```

vaadin

Phantom JS opens a preview version of the code & resulting app and screenshots it

```
function sleepAndTry() {
  window.setTimeout(function () {
    var vaadinReady = page.evaluate(isVaadinReady);
    console.log("vaadin:");
    console.log(vaadinReady);
    if (vaadinReady) {
      // disable CSS scaling as it doesn't seem to work in PhantomJS
      page.evaluate(disableScaling);
      page.evaluate(hideLoadingIndicators);
      window.setTimeout(function () {
        page.render(output);
        phantom.exit();
      }, 20);
    }
    else {
      sleepAndTry();
    }
```

OK, GREAT

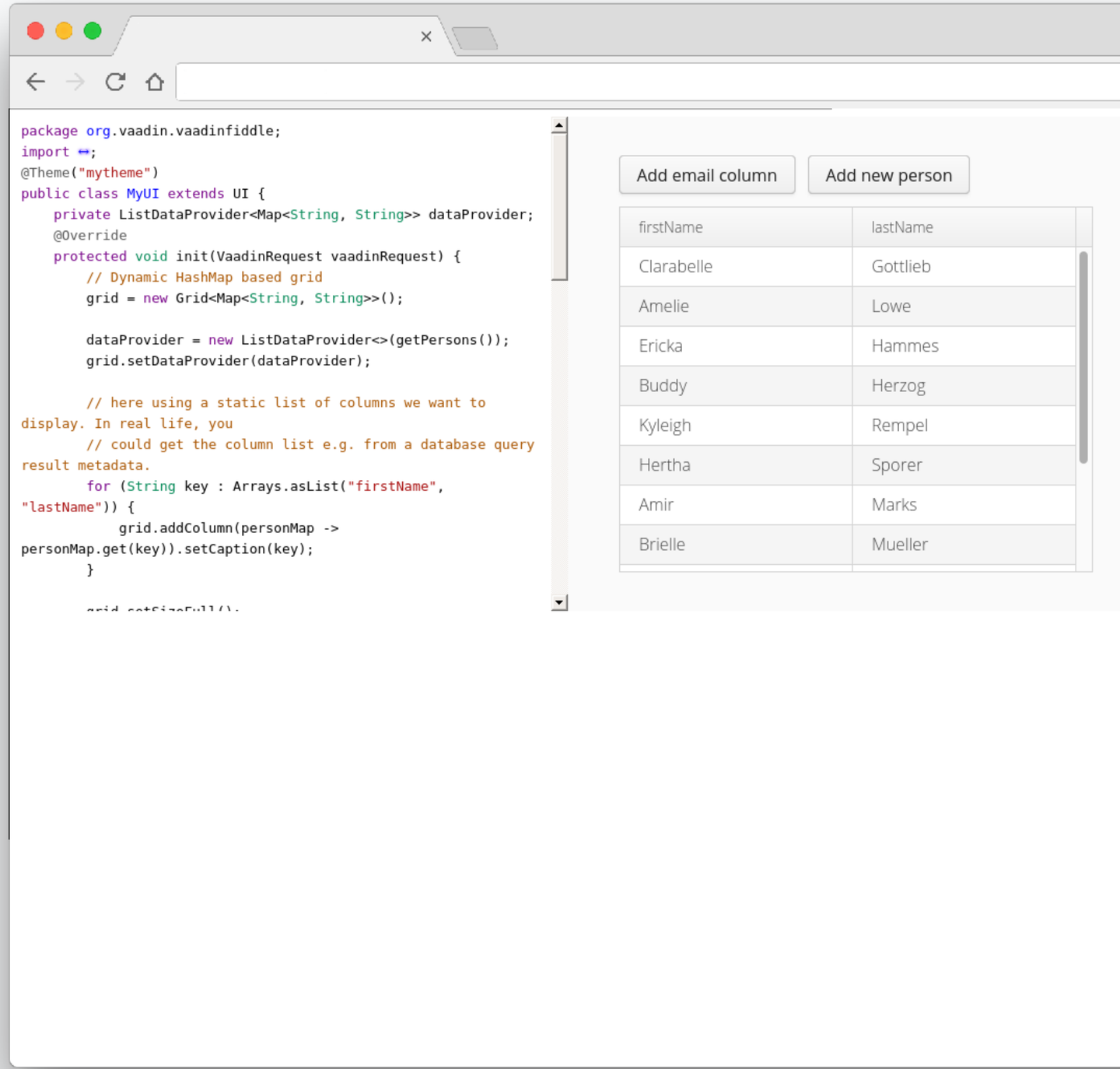
BUT WHAT CAN U DO WITH IT?

vaadin}>

Will be a great tool for

- Sharing examples
- Demoing Vaadin add-ons
- Collaborating with other Vaadin users e.g. by showing how to solve Vaadin forum users problems.

Demoing how to use DataProvider

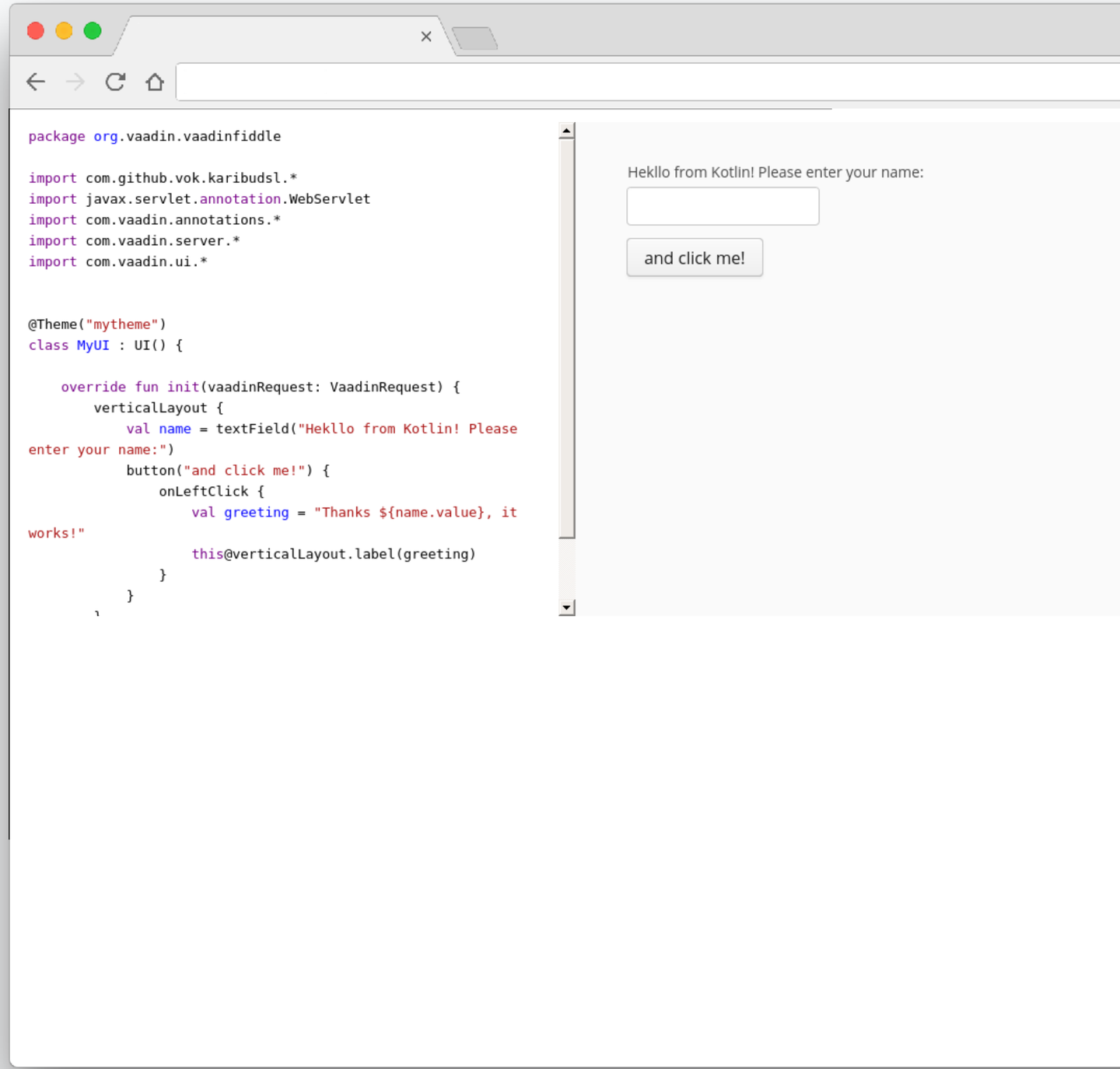


```
package org.vaadin.vaadinfiddle;
import ⇔;
@Theme("mytheme")
public class MyUI extends UI {
    private ListDataProvider<Map<String, String>> dataProvider;
    @Override
    protected void init(VaadinRequest vaadinRequest) {
        // Dynamic HashMap based grid
        grid = new Grid<Map<String, String>>();

        dataProvider = new ListDataProvider<>(getPersons());
        grid.setDataProvider(dataProvider);

        // here using a static list of columns we want to
display. In real life, you
        // could get the column list e.g. from a database query
result metadata.
        for (String key : Arrays.asList("firstName",
"lastName")) {
            grid.addColumn(personMap ->
personMap.get(key)).setCaption(key);
        }

        grid.setSizeFull();
```

Add email column | Add new person

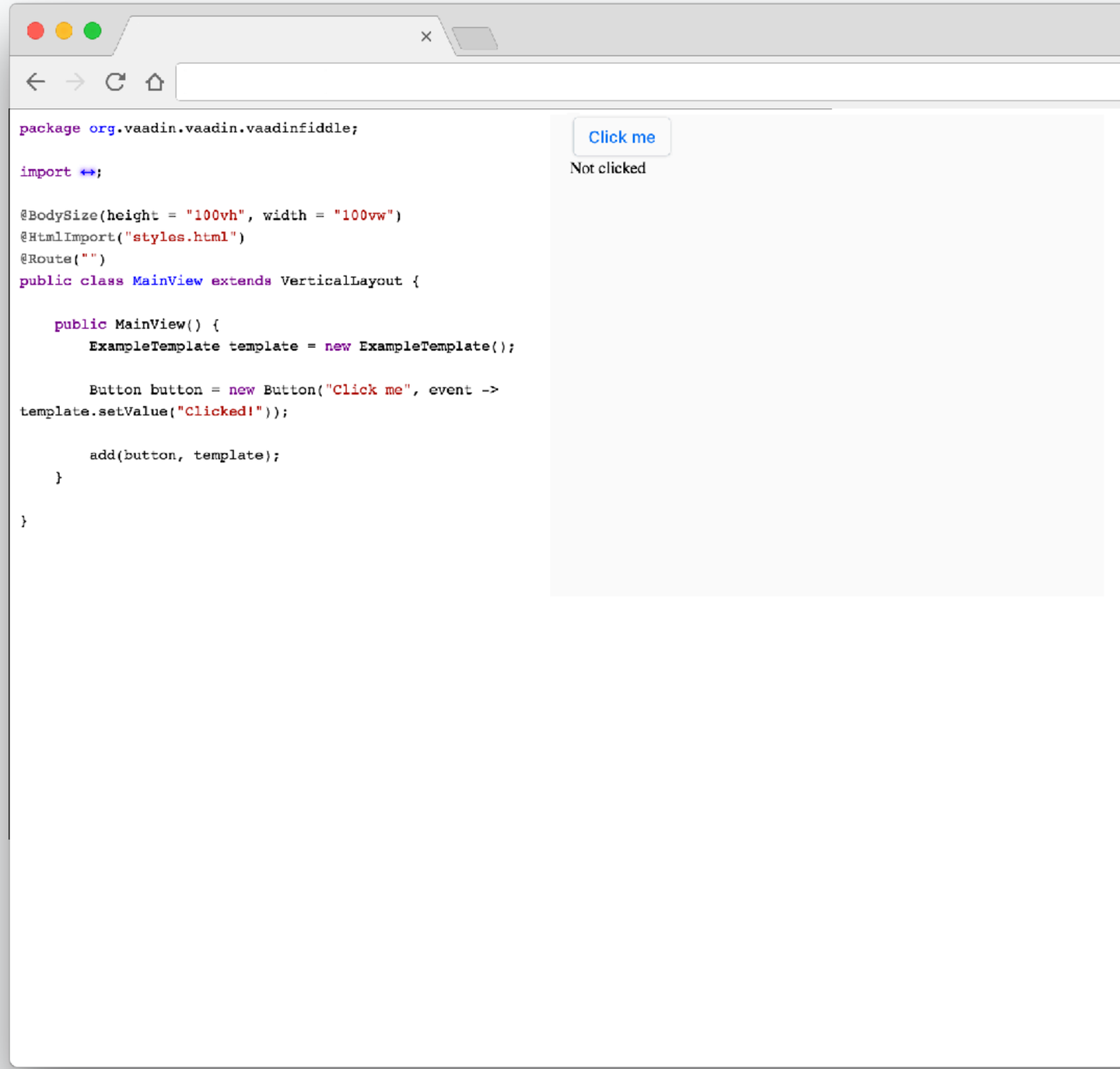
firstName	lastName
Clarabelle	Gottlieb
Amelie	Lowe
Ericka	Hammes
Buddy	Herzog
Kyleigh	Rempel
Hertha	Sporer
Amir	Marks
Brielle	Mueller

Vaadin 8 Kotlin demo



```
package org.vaadin.vaadinfiddle

import com.github.vok.karibudsl.*
import javax.servlet.annotation.WebServlet
import com.vaadin.annotations.*
import com.vaadin.server.*
import com.vaadin.ui.*


@Theme("mytheme")
class MyUI : UI() {

    override fun init(vaadinRequest: VaadinRequest) {
        verticalLayout {
            val name = textField("Hekllo from Kotlin! Please
enter your name:")
            button("and click me!") {
                onLeftClick {
                    val greeting = "Thanks ${name.value}, it
works!"
                    this@verticalLayout.label(greeting)
                }
            }
        }
```

Vaadin 10 demo



```
package org.vaadin.vaadin.vaadinfiddle;

import ↔;

@BodySize(height = "100vh", width = "100vw")
@HtmlImport("styles.html")
@Route("")
public class MainView extends VerticalLayout {

    public MainView() {
        ExampleTemplate template = new ExampleTemplate();

        Button button = new Button("Click me", event ->
template.setValue("Clicked!"));

        add(button, template);
    }

}
```

Click me

Not clicked

Roadmap and timeline

Hoping that we can launch the service for community in few months

3/2018	Make the system run on a cloud hosting provider's infra
NEXT	Minor UX improvements
In few months	Use for all Java code samples on vaadin.com

vaadin

Roadmap and timeline

Endless possibilities for improvements and new features

- Improve save -> result app load time
- Auto organize imports
- Provide context aware autofill & JavaDoc
- More stubs to start from (different Vaadin
- Versions, Spring Boot, Kotlin stub etc.)
- Export to GitHub
- Revisions
- User accounts
- ...

vaadin}>

THANKS!

vaadin}>