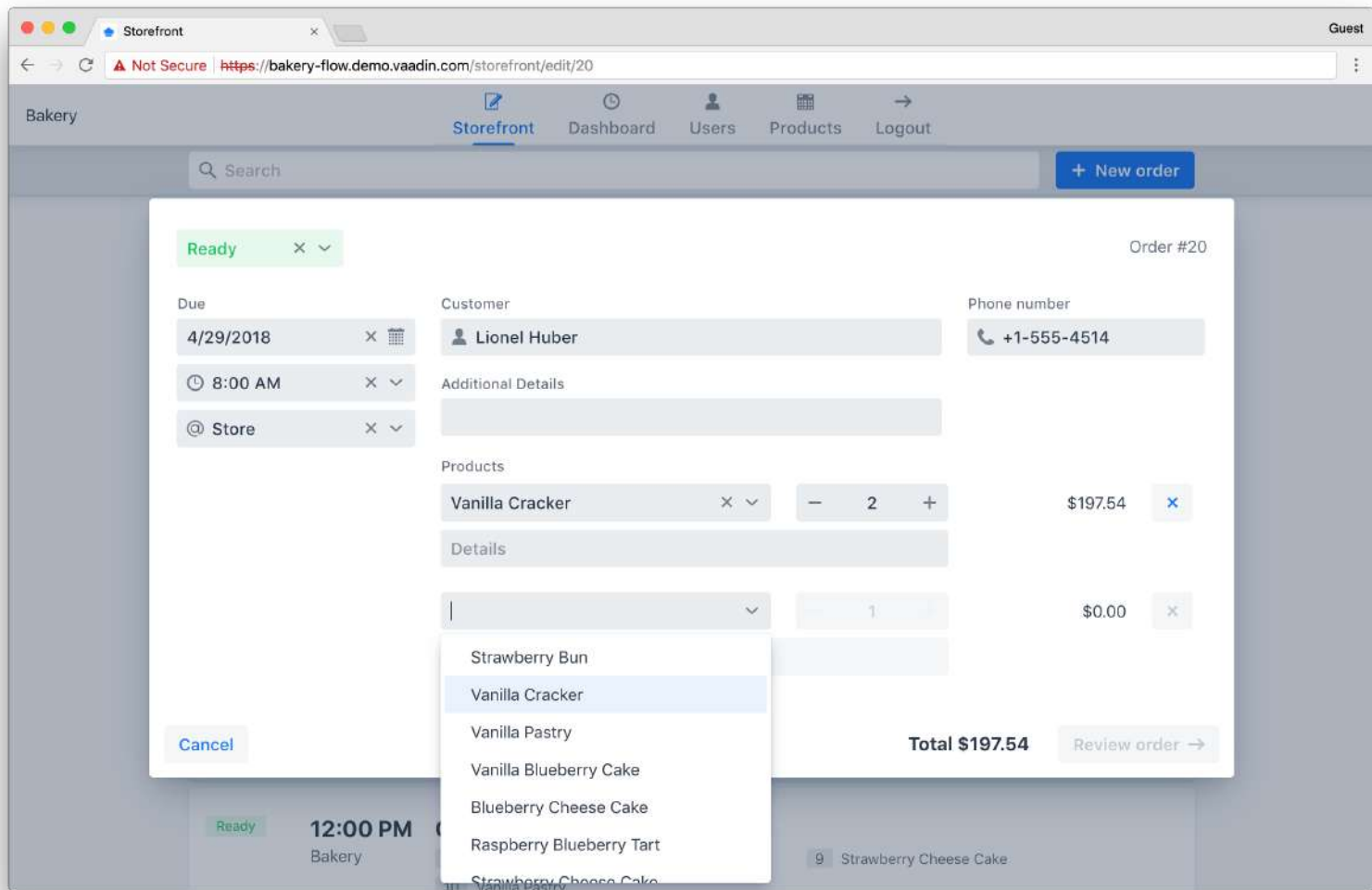


使用 Vaadin 搭配 Kotlin 快速开发 Web 应用

Maggie

November 27, 2022



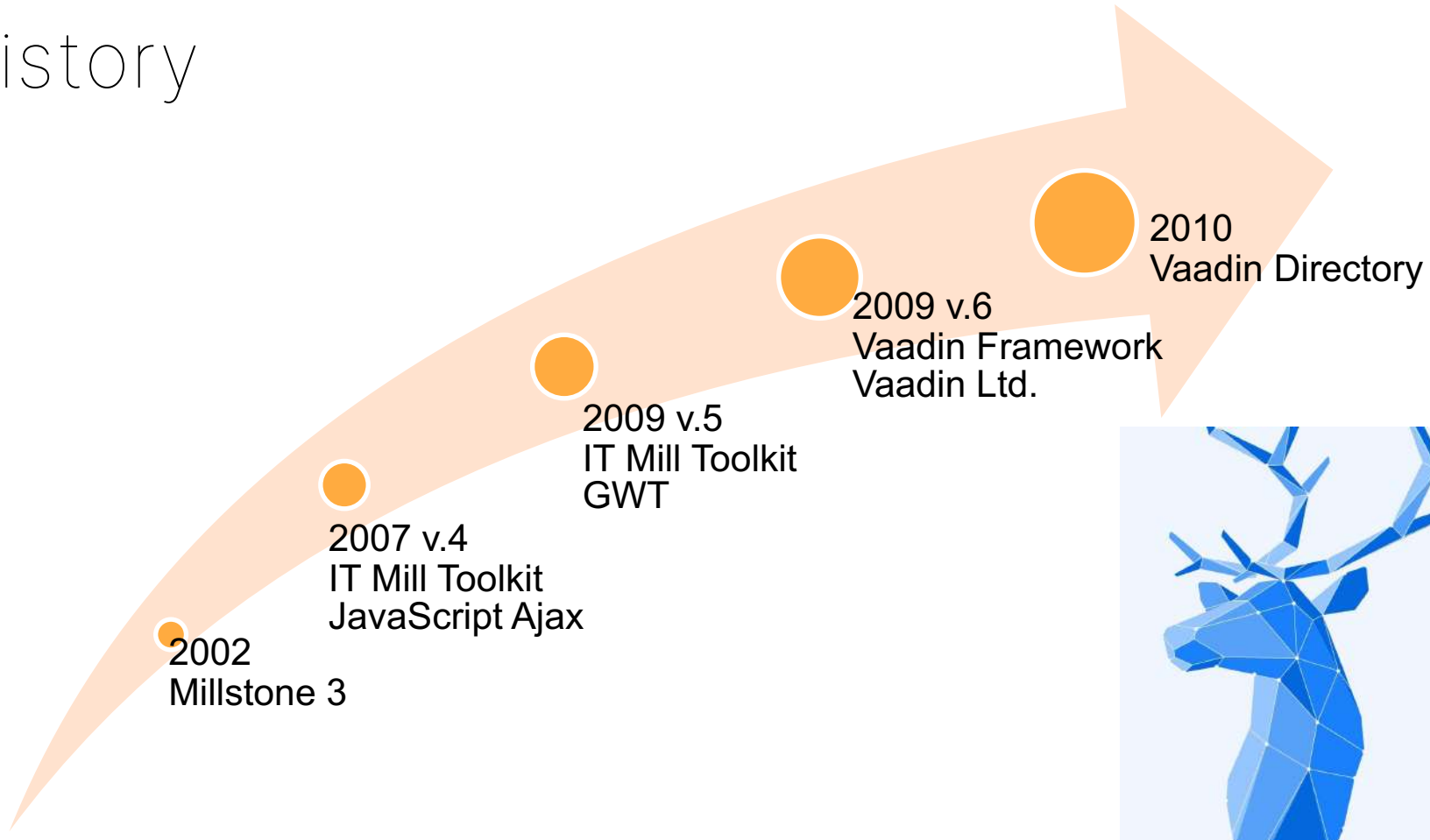
Agenda

- Introduction Vaadin
- Route, RouteLayout
- Login
- Data Binding
- CRUD
- Custom Component
- Rest API
- CRUD Pro

What is Vaadin?



History



Vaadin releases version

- 23 LTS – 2022/03
- 22 – 2021/12
- 21 – 2021/09
- 20 – 2021/06
- 19 – 2021/05
- 18 – 2020/12
- 17 – 2020/09
- 16 – 2020/01
- 15 – 2020/05
- 14 LTS – 2019/05
- 14.9.1 - 2022/10
- 13 – 2019/05
- 12 – 2018/12
- 11 – 2018/09
- 10 LTS – 2018/06
- 8 LTS – 2017/2
- 7 LTS – 2013/2
- 6 LTS – 2009/5

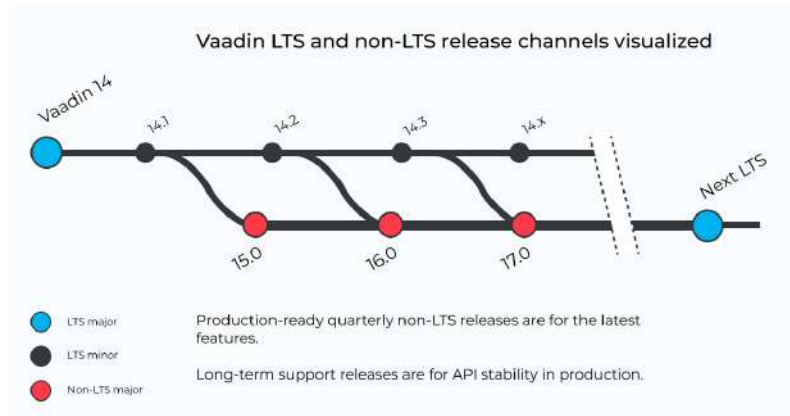
LTS and non-LTS release channels

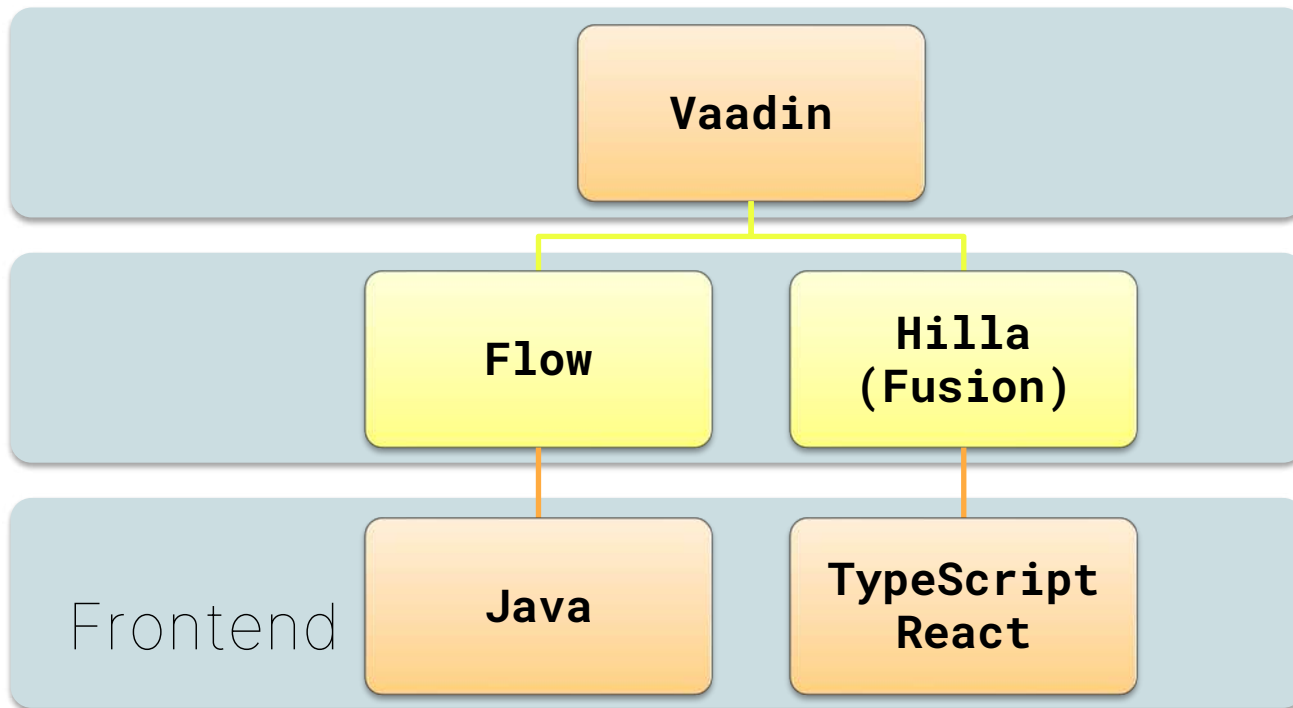
Non-LTS

- You get all the latest features.
- To receive continued updates and support you need to update your application dependencies every three months.
- You might need to migrate code because of API-incompatible changes in new major versions.

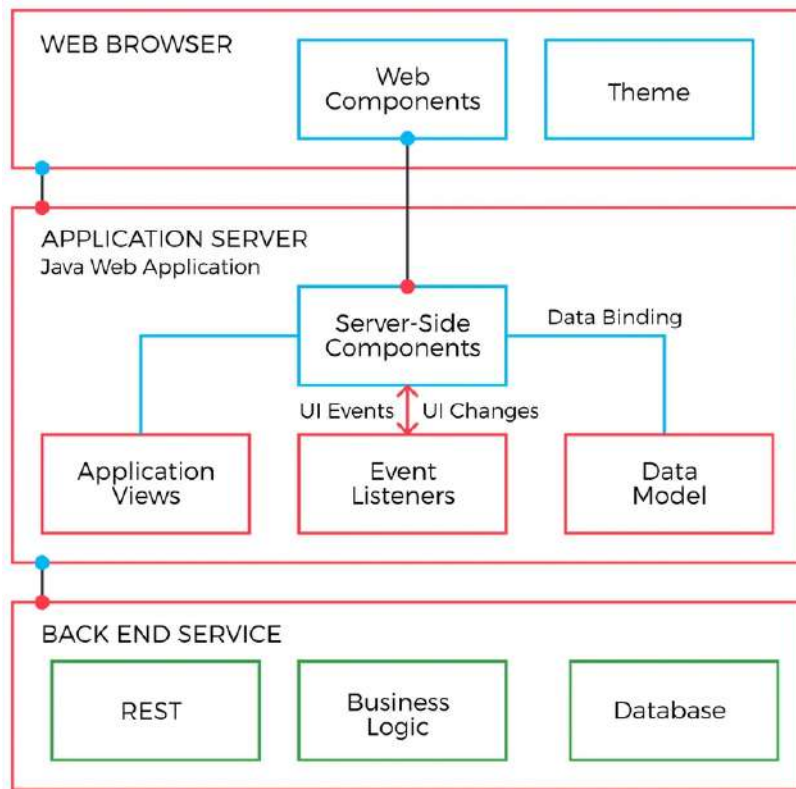
LTS

- You get long-term API stability.
- Some of the newest features are missing



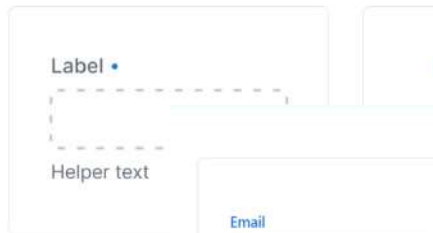


How does Vaadin work?



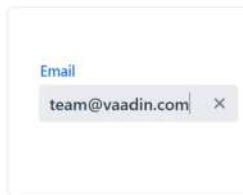
Components

Form Inputs



Input Fields

Input field components features, such as label, and required states, and



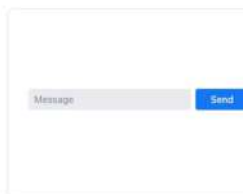
Email Field

Email Field, an extension of Text Field, only accepts email addresses as input.



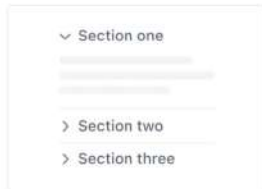
Custom Field

Custom Field is a component that combines multiple components as



Message Input

Message input allows users to author and send messages.



Accordion

Accordion is a vertically stacked set of expandable panels.



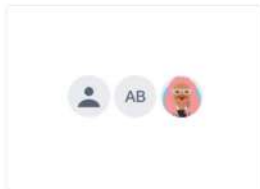
Button

The Button component allows users to perform actions.



Multi-Select Combo Box

Multi-Select Combo Box allows the user to choose one or more values from a filterable list of options presented in an overlay.



Avatar

Avatar is a graphical representation of an object or entity, for example a person or an organization.



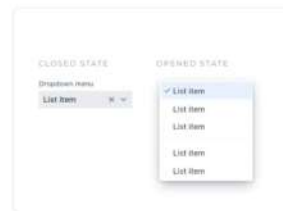
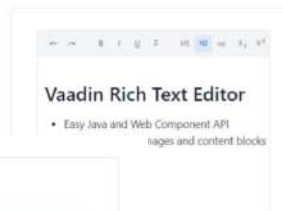
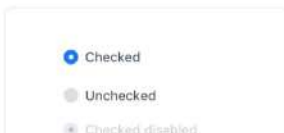
Charts

Charts is a feature-rich interactive library with a vast number of chart types from basic plotting to complex



Password Field

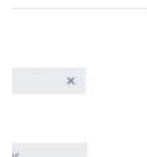
Password Field is an input component for passwords.



Select

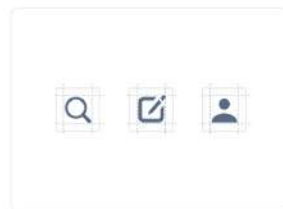
Select allows users to choose a single value from a list of options presented in an overlay.

Input field for entering



Grid Pro

Grid Pro is an extension of the Grid component that provides inline editing with full keyboard navigation.



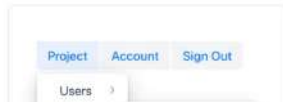
Icons

Over 600 built-in icons for business applications.



Map

Map is a component for displaying geographical maps from various sources.



Add-on Directory

The screenshot shows the Vaadin Add-on Directory website. The header includes the Vaadin logo and navigation links: Platform, Solutions, Company, Developers, Resources, Pricing, Contact us, Get started, and Login. A search bar on the left shows '1991 components'. The main content area is titled 'Add-on Directory' with the subtitle 'Extend Vaadin with open source components.' It features a grid of add-ons. The first add-on is 'TwinColSelect' by Tatu Lund, which is a 'UI' component. It has a 5-star rating and was updated 8 hours ago. The second add-on is 'SO Components' by Syam Pillai, which is a 'Miscellaneous' component. It has a 5-star rating and was updated 10 hours ago. The third add-on is 'Breeze theme for Vaadin' by Team Partio, which is a 'Themes' component. It has a 5-star rating and was updated 7 days ago. The fourth add-on is 'CKEditorVaadin' by Ryan Pang, which is a 'UI' component. It has a 5-star rating and was updated 7 days ago. The right sidebar shows details for 'TwinColSelect', including its version (2.2.2), release date (24 November 2020), and framework support (Vaadin platform 23+). It also includes a section for 'Assorted examples for different use cases' with code snippets.

Directory | Vaadin

https://vaadin.com/directory/

Platform Solutions Company Developers Resources Pricing Contact us Get started Login

SEARCH

1991 components

FRAMEWORKS

Vaadin platform

Polymer 3

GWT 2

TAGS

UI (1023) Data (237)

Themes (109) Tools (159)

Miscellaneous (267)

Web Components (879)

Sponsored (107)

FOLLOW US

Follow us on Twitter to get Directory feeds.

A new Vaadin Directory search is available! Try it now →

Add-on Directory

Extend Vaadin with open source components.

Help Log in to publish

Last updated Highest rated A-Z

UI

TwinColSelect
by Tatu Lund
TwinColSelect for Flow (Vaadin 14 and 23)
★★★★★
Updated 8 hours ago

Miscellaneous UI

SO Components
by Syam Pillai
A set of Vaadin Flow Components and Abstractions
★★★★★
Updated 10 hours ago

Themes

Breeze theme for Vaadin
by Team Partio
Breeze is built on top of Lumo and heavily inspired by visual style of Tailwind CSS
★★★★★
Updated 7 days ago

UI Tools Data

CKEditorVaadin
by Ryan Pang
CKEditor 5 and Vaadin, rich editor including multiple functionalities
★★★★★
Updated 7 days ago

TwinColSelect

by Tatu Lund

★★★★★ 8 8 hours ago

Click to rate

TwinColSelect for Flow (Vaadin 14 and 23)

Overview Discussions (34) Code Samples Links API

Assorted examples for different use cases

```
// Create a new TwinColSelect
TwinColSelect<String> select = new TwinColSelect<>();

select.setItems("One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten");

// Set comparator for sorting
dp.setSortComparator((a, b) -> a.compareTo(b));

// Use selection listener
select.addSelectionListener(event -> {
    log.removeAll();
    log.addComponentAsFirst(new Span("Value changed"));
    event.getValue().forEach(item -> log.addComponentAsFirst(new Span(item + "
```

COMPONENT VERSION

Version 2.2.2

[Link to this version](#)

Beta

Released 24 November 2020

[Apache License 2.0](#)

FRAMEWORK SUPPORT

Vaadin platform 23+

BROWSER COMPATIBILITY



INSTALL WITH

Maven Zip

```
<dependency>
<groupId>org.vaadin.tatu
<artifactId>twincolselect
<version>2.2.2</version>
</dependency>
```

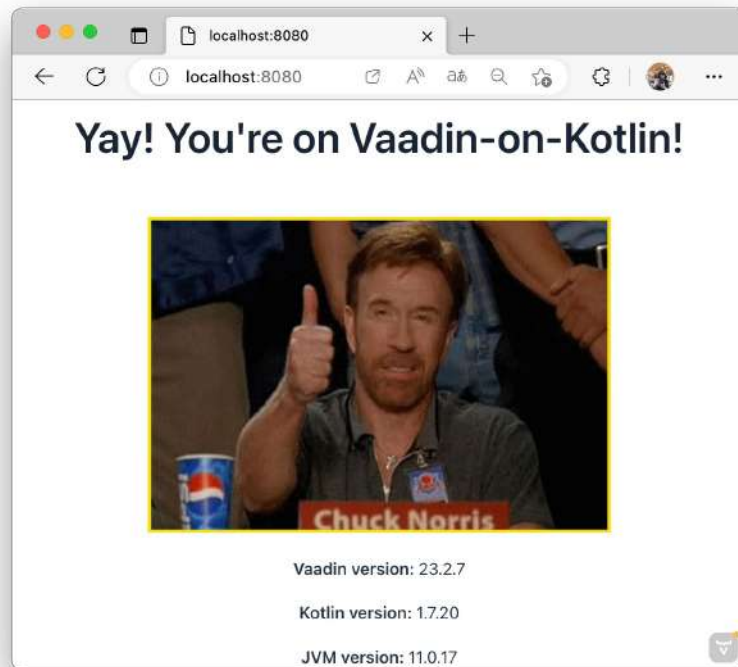
```
<repository>
<id>vaadin-addons</id>
<url>https://maven.vaadin
</repository>
```

Vaadin on Kotlin

- Based on Vaadin 23 LTS.
- Vaadin is a single-page web component framework.
- Vaadin Boot

Getting started with Vaadin-on-Kotlin

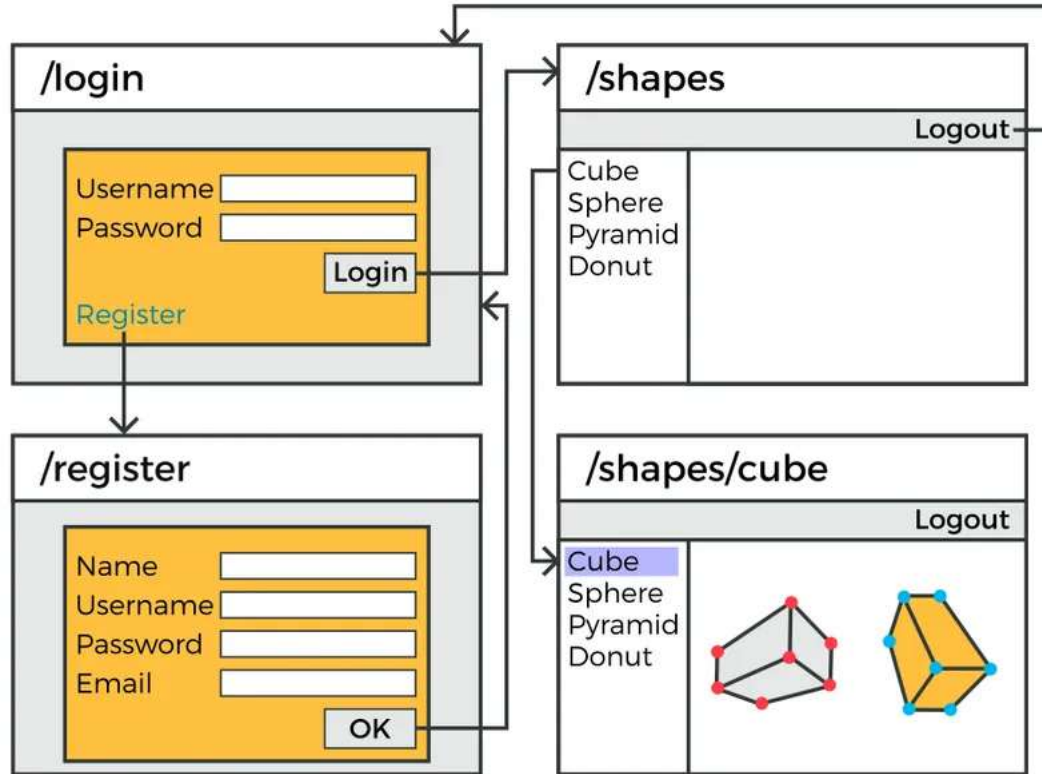
- Java JDK 11
 - Gradle
 - IntelliJ IDEA
1. `git clone https://github.com/mvysny/vok-helloworld-app`
 2. `./gradlew`
 3. `./gradlew clean run`





Route
RouterLayout

@Route



Router

```
@Route("/")
```

```
class WelcomeView: KComposite() {
```

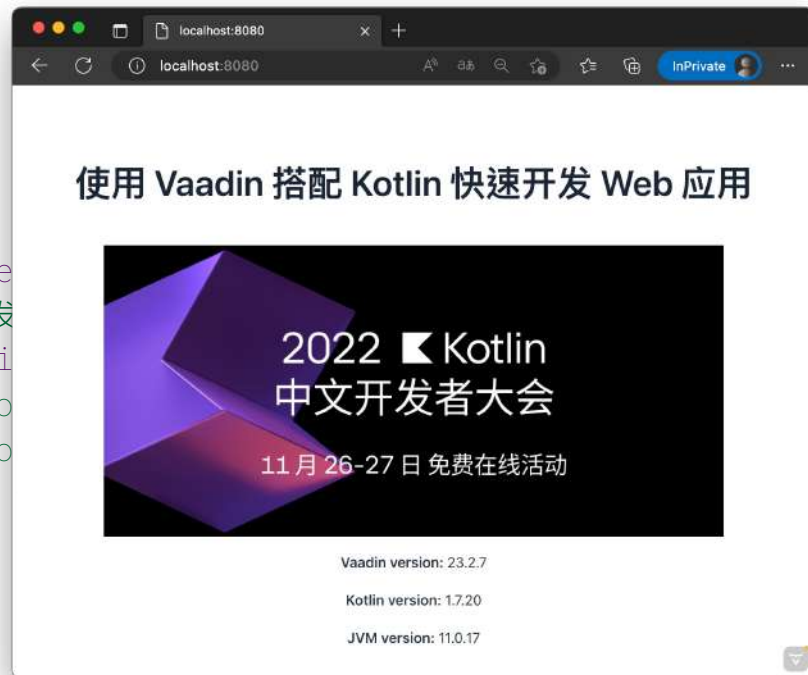
```
    private val root = ui {
```

```
        verticalLayout {
```

```
            setSizeFull(); content { align(ce  
            h1("使用 Vaadin 搭配 Kotlin 快速开发 Web 应用  
            image("images/kotlin2022.jpg"){wi  
            div { html("<strong>Vaadin versio  
            div { html("<strong>Kotlin versio  
            div { html("<strong>JVM version:
```

```
        }
```

```
    }
```



RouterLayout

```
class MainLayout: KComposite(), RouterLayout {  
    val root = ui{  
        appLayout {  
            addToNavbar(  
                DrawerToggle(),  
                Image("images/jetbrains_icon.png", "2022 Kotlin 中文开发者大会")  
                    .also{ it.height = "60px" }, Label("2022 Kotlin 中文开发者大会"))  
            val tabs = Tabs(  
                Tab(VaadinIcon.HOME.create(), RouterLink("Home", WelcomeView::class.java)),  
            )  
            tabs.orientation = Tabs.Orientation.VERTICAL  
            addToDrawer(tabs)  
        }  
    }  
}
```

Router

```
@Route("", layout = MainLayout::class)
```

```
class WelcomeView: KComposite() {
```

```
    private val root = ui {
```

```
        verticalLayout {
```

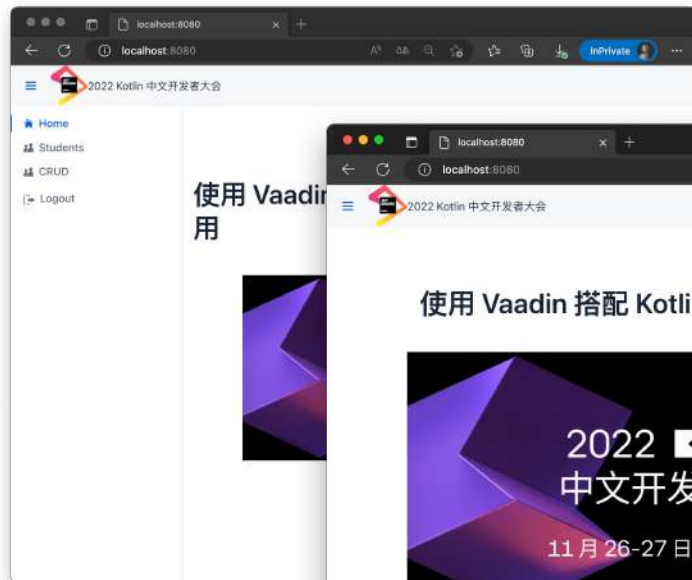
```
            :
```

```
            :
```

```
        }
```

```
    }
```

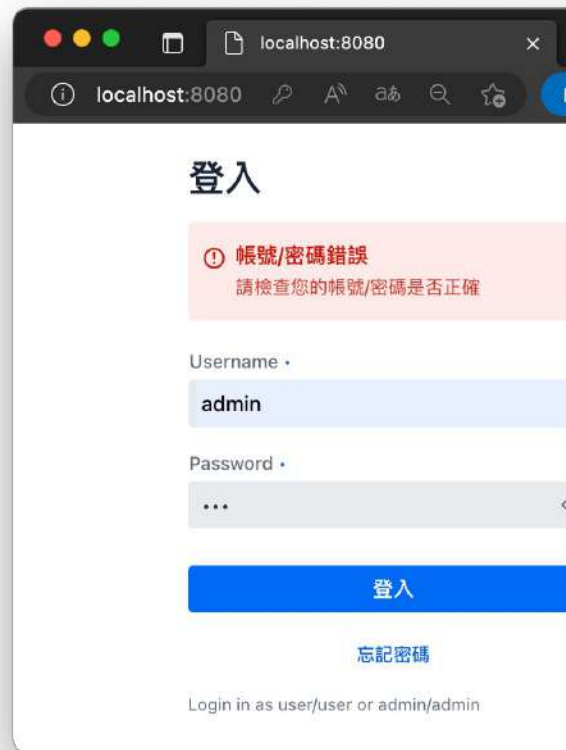
```
}
```



LoginForm / BeforeEnterObserver

```
@Route("login")
class LoginView:KComposite(), BeforeEnterObserver {
    val root = ui {
        verticalLayout {
            val i18n = loginI18n {...}
            loginForm(i18n) {
                addLoginListener {...}
            }
        }
    }

    override fun beforeEnter(event: BeforeEnterEvent?) {
        if (Session.loginService.isLoggedIn)
            event?.rerouteTo("")
    }
}
```



```

@Route("login")
class LoginView:KComposite(), BeforeEnterObserver {
    private lateinit var form: LoginForm
    val root = ui {
        verticalLayout {
            setSizeFull(); content { align(center, middle) }; isMargin = false; isSpacing = true
            val i18n = loginI18n {
                header.title = "歡迎來到 KOTLIN 世界"
                additionalInformation = "Login in as user/user or admin/admin"
                with(form){
                    title = "登入"
                    submit = "登入"
                    forgotPassword = "忘記密碼"
                }
                with(errorMessage){
                    title = "帳號/密碼錯誤"
                    message = "請檢查您的帳號/密碼是否正確"
                }
            }
            form = loginForm(i18n) {
                addLoginListener { e->
                    if (!Session.loginService.login(e.username, e.password)){
                        isError = true
                    }
                }
            }
        }
    }

    override fun beforeEnter(event: BeforeEnterEvent?) {
        if (Session.loginService.isLoggedIn)
            event?.rerouteTo("")
    }
}

```

```
class LoginService : Serializable {
    var currentUser: User? = null
        private set
    val isLoggedIn get() = currentUser != null

    fun login(username: String, password: String): Boolean {
        if ((username == "user" && password == "user") || (username == "admin" && password == "admin")) {
            currentUser = User(username)
            UI.getCurrent().page.reload()
            return true
        } else return false
    }

    fun logout() {
        Session.current.close()
        UI.getCurrent().navigate("")
        UI.getCurrent().page.reload()
    }
}

val Session.loginService get() = getOrPut(LoginService::class) { LoginService() }

data class User(val name: String) : Serializable
```

RouterLayout

```
class MainLayout : KComposite(), RouterLayout, BeforeEnterObserver {
    val tabs = Tabs(
        Tab(VaadinIcon.HOME.create(), RouterLink("Home", WelcomeView::class.java)),
    )
    var tabLogin = Tab(VaadinIcon.SIGN_IN.create(), RouterLink("Login", LoginView::class.java))
    var tabLogout = Tab(VaadinIcon.SIGN_OUT.create(), Label().p("Logout").also {
        it.addClickListener {Session.loginService.logout()}
    })
    val root = ui {appLayout {...}}

    override fun beforeEnter(event: BeforeEnterEvent?) {
        if (event?.navigationTarget != LoginView::class.java && !Session.isLoggedIn) {
            event?.rerouteTo(LoginView::class.java)
        } else {
            tabs.add(tabLogout)
        }
    }
}
```

Data Binding

- Create Table
- Grid
- CRUD

1. Create Table

create TABLE student

(

id **bigint auto_increment PRIMARY KEY,**

name **VARCHAR(200) NOT NULL,**

birthday **DATE,**

created **TIMESTAMP,**

email **VARCHAR(60),**

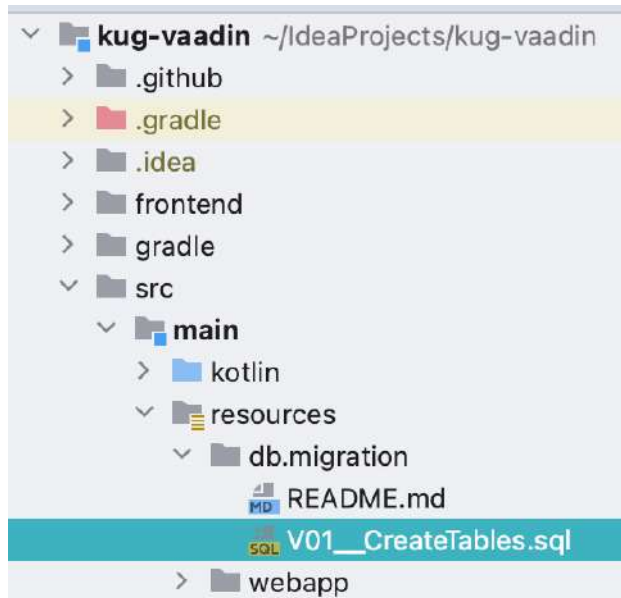
gender **VARCHAR(20) NOT NULL,**

height **DOUBLE NOT NULL,**

weight **DOUBLE NOT NULL,**

student_id **VARCHAR(20)**

);



2. KEntity, DAO

```
data class Student(  
    override var id: Long? = null,  
    @field:NotNull @field:Length(min = 2, max = 10, message = "...")  
    var name: String? = null,  
    :  
) : KEntity<Long> {  
    override fun save(validate: Boolean) {  
        val isNew = (id == null)  
        super.save(validate)  
        if (isNew) {  
            :  
        }  
    }  
}  
companion object : Dao<Student, Long>(Student::class.java)  
}
```

3.Display Students with Grid

```
@Route("students", layout = MainLayout::class)
class AllStudentsView : KComposite(){
    private val root = ui {
        verticalLayout {
            h1("學生資料")
            grid<Student> {
                isExpand = true
                setDataLoader(Student.dataLoader)
                addColumnFor(Student::id).setHeader("序號")
                addColumnFor(Student::student_id).setHeader("學號")
                :
            }
            grid.columns.forEach {it.setAutoWidth(true)}
        }
    }
}
```

序號	學號	姓名	性別	生日
1	KUG20220001	曲惠如	Male	2002-05-03
2	KUG20220002	朱佩珊	Custom	2002-01-20
3	KUG20220003	柳宜芳	Custom	2002-03-30
4	KUG20220004	侯智偉	Male	2002-03-14
5	KUG20220005	祝怡君	Female	2002-09-11
6	KUG20220006	杜佩玲	Male	2002-05-17
7	KUG20220007	袁雅惠	Custom	2002-09-14
8	KUG20220008	房欣怡	Female	2002-05-29

CRUD

```
@Route("create-student", layout = MainLayout::class)
class CreateStudentView : KComposite() {
    private val binder = beanValidationBinder<Student>()
    private val root = ui {
        verticalLayout {
            h1("新增學生資料")
            textField("姓名") {bind(binder).bind(Student::name)}
            datePicker("生日") {...}
            comboBox<Gender>("性別") {...}
            numberField("身高") {...}
            :
            button("Save") {onLeftClick {
                val student = Student()
                if (binder.validate().isOk && binder.writeBeanIfValid(student)){
                    student.save()
                }
            }}
        }
    }
}
```



CRUD

```
@Route("edit-student", layout = MainLayout::class)
class EditStudentView : KComposite(), HasUrlParameter<Long> {
    private val binder = beanValidationBinder<Student>()
    private var student: Student? = null
    private val root = ui {
        verticalLayout {
            :
        }
    }
    override fun setParameter(event: BeforeEvent?, studentId: Long?) {
        student = Student.getById(studentId!!)
        binder.readBean(student)
    }
}
```

CRUD

```
@Route("edit-student", layout = MainLayout::class)
class EditStudentView : KComposite(), HasUrlParameter<Long> {
    private val binder = beanValidationBinder<Student>()
    private var student: Student? = null
    private val root = ui {
        verticalLayout {
            :
            button("儲存"){
                onLeftClick {
                    if (binder.validate().isOk && binder.writeBeanIfValid(student))
                        student!!.save()
                }
            }
        }
    }
}
```

CRUD

```
@Route("students", layout = MainLayout::class)
```

```
class AllStudentsView : KComposite(), AfterNavigationObserver {
```

```
    private lateinit var grid: Grid<Student>
```

```
    private val root = ui {
```

```
        verticalLayout {
```

```
            :
```

```
            grid = grid {
```

```
                :
```

```
                addColumn(NativeButtonRenderer("Show", { navigateTo(StudentView::class, it.id) }))
```

```
                addColumn(NativeButtonRenderer("Edit", { navigateTo(EditStudentView::class, it.id) })
```

```
                addColumn(NativeButtonRenderer("Delete") {
```

```
                    confirmDialog {
```

```
                        it.delete()
```

```
                        this.refresh()
```

```
                    }
```

```
                })
```

```
            }
```

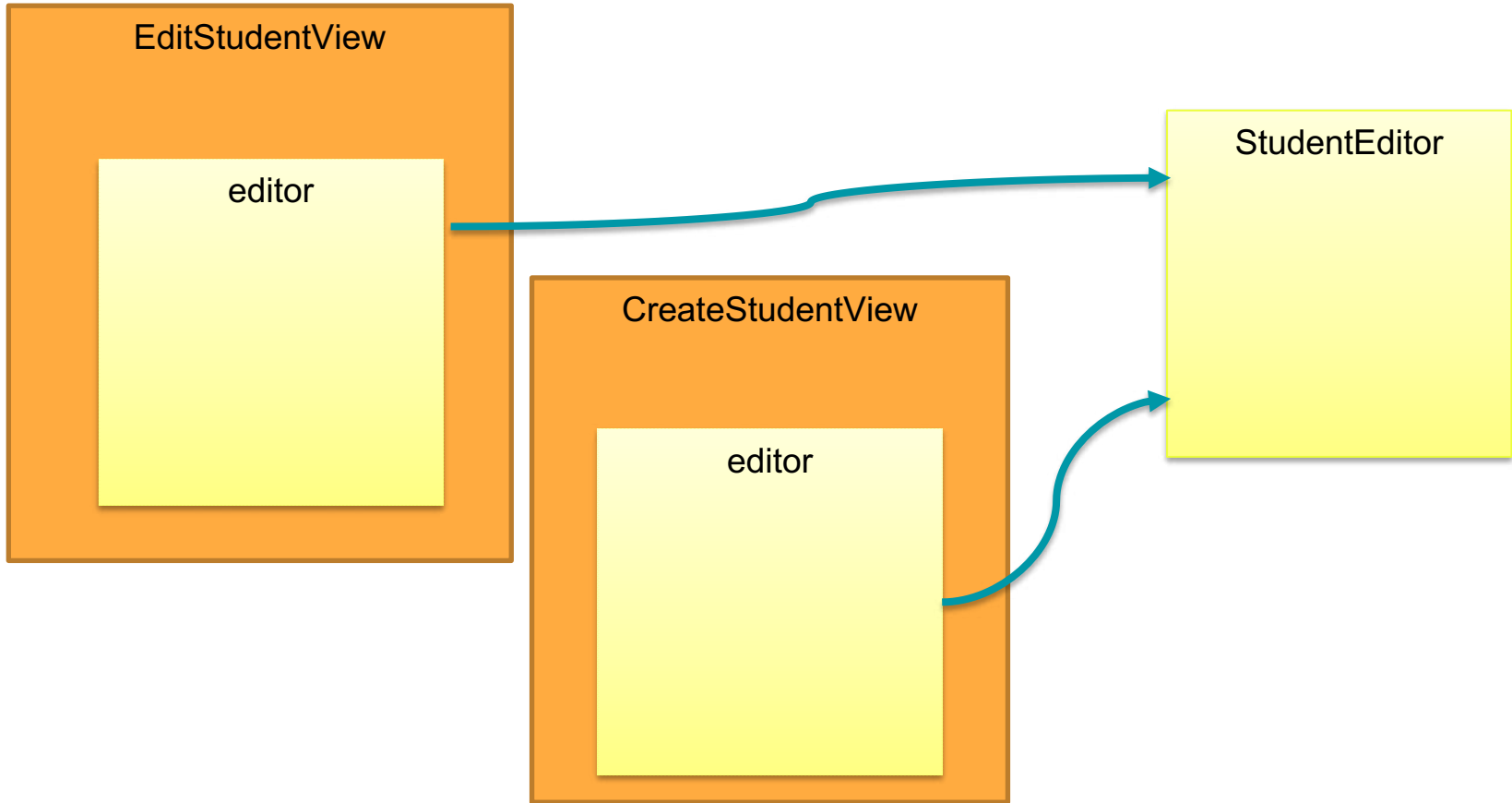
```
        }
```

序號	學號	姓名	性別	生日	身高	體重			
1	KUG20220001	穆佳穎	Male	2002-06-26	164.2	62.4	Show	Edit	Delete
2	KUG20220002	楊怡伶	Female	2002-08-21	179.2	65.6	Show	Edit	Delete
3	KUG20220003	李馨儀	Male	2002-05-11	169.5	67.2	Show	Edit	Delete
4	KUG20220004	湯靜宜	Male	2002-08-29	169.2	49.4	Show	Edit	Delete

Custom Component



Reuse ?



Build an Editor - 1

```
class StudentEditor : KComposite() {  
    private val binder = beanValidationBinder<Student>()  
    var student: Student? = null  
    set(value) {  
        field = value  
        if (value != null) binder.readBean(value)  
    }  
  
    private val root = ui {  
        verticalLayout {  
            (next page)  
        }  
    }  
}  
  
fun HasComponents.studentEditor(block: StudentEditor.() -> Unit = {})  
    = init(StudentEditor(), block)
```

Build an Editor - 2

```
textField("姓名 : ") {bind(binder).bind(Student::name)}
comboBox<Gender>("性別 : ") {
    setItems(*Gender.values())
    bind(binder).bind(Student::gender)
}
datePicker("生日 : ") {bind(binder).bind(Student::birthday)}
:

button("儲存") {
    onLeftClick {
        val student = student!!
        if (binder.validate().isOk && binder.writeBeanIfValid(student)) {
            student.save()
            navigateTo(StudentView::class, student.id)
        }
    }
}
```

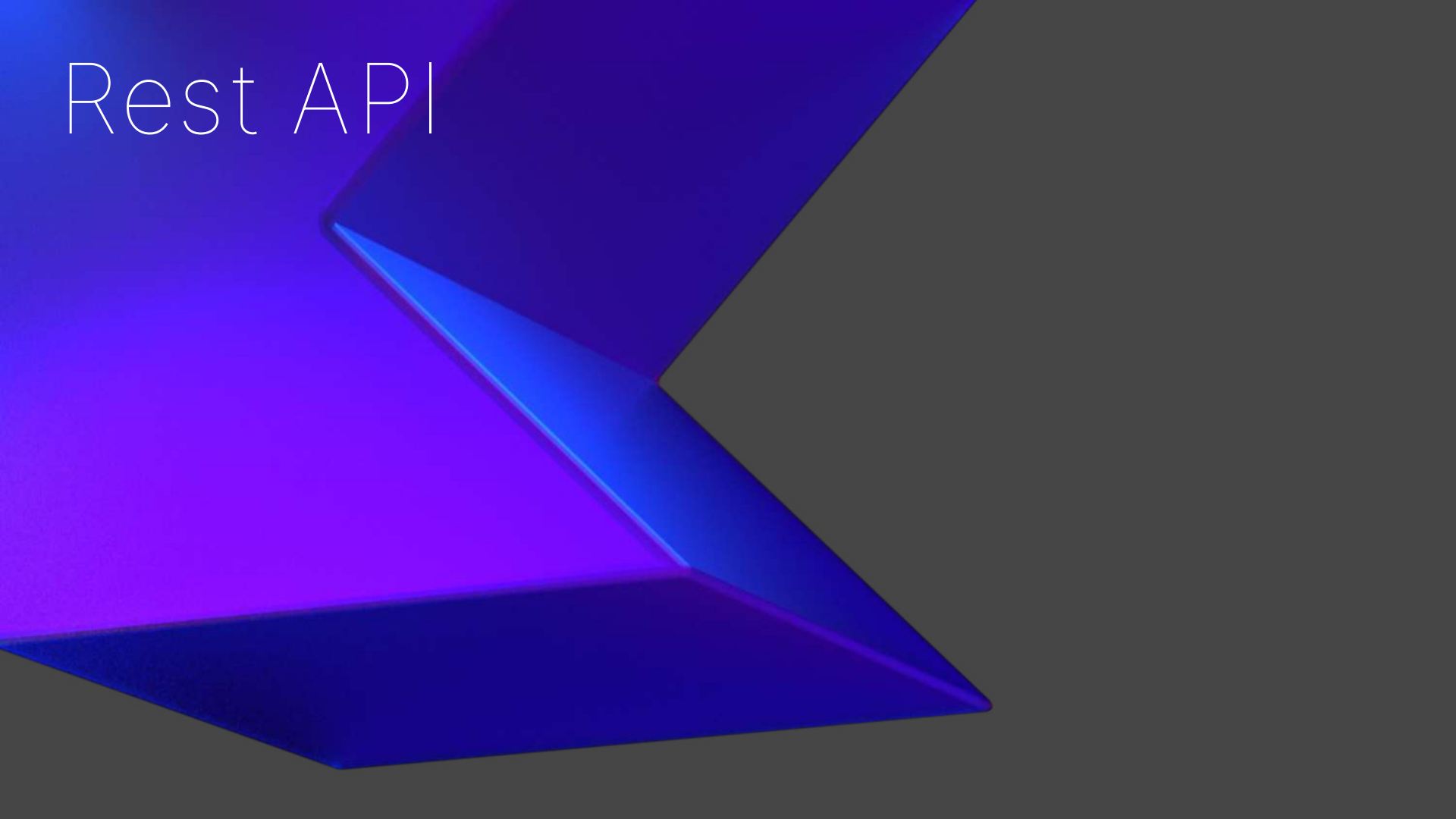
Refactor EditStudentView

```
@Route("edit-student", layout = MainLayout::class)
class EditStudentView : KComposite(), HasUrlParameter<Long> {
    private lateinit var editor: StudentEditor
    private val root = ui {
        verticalLayout {
            routerLink(null, "返回", AllStudentsView::class)
            h1("學生資料修改")
            editor = studentEditor()
        }
    }
    override fun setParameter(event: BeforeEvent?, studentId: Long?) {
        editor.student = Student.getById(studentId!!)
    }
}
```

Refactor CreateStudentView

```
@Route("create-student", layout = MainLayout::class)
class CreateStudentView : KComposite() {
    private lateinit var editor: StudentEditor
    private val root = ui {
        verticalLayout {
            h1("新增學生資料")
            editor = studentEditor {
                student = Student()
            }
        }
    }
}
```

Rest API



Rest API Example

```
fun Javalin.studentRest(){
    get("/rest/student/{id}"){ ctx ->
        val id = ctx.pathParam("id").toLong()
        ctx.json(Student.findById(id)
            ?: throw NotFoundResponse("找不到學生編號 $id") )
    }
    get("/rest/students"){ ctx ->
        ctx.json(Student.findAll())
    }
    get("/rest/student/id/{student_id}"){
        val id = it.pathParam("student_id")
        it.json(Student.getOneBy{ Student::student_id eq id } )
    }
}
```

Rest API Example

```
fun Javalin.configureRest(): Javalin {  
    gsonMapper(VokRest.gson)  
    studentRest()  
    return this  
}
```

student.id = 1

```
get("/rest/student/{id}") { ctx ->
    val id = ctx.pathParam("id").toLong()
    ctx.json(Student.findById(id)
        ?: throw NotFoundResponse("找不到學生編號 $id")
    }
```

```
GET http://localhost:8080/rest/student/1
Accept: application/json

http://localhost:8080/rest/student/1

HTTP/1.1 200 OK
Date: Sat, 26 Nov 2022 17:16:38 GMT
Content-Type: application/json
Content-Length: 170
Server: Jetty(10.0.11)

{
  "id": 1,
  "name": "利雅萍",
  "birthday": "2002-02-11",
  "created": "2022-11-27",
  "email": "tet@ttt.com",
  "gender": "Female",
  "height": 157.5,
  "weight": 48.7,
  "student_id": "KUG20220001"
}
```


student.student_id = 1

```
get("/rest/student/id/{student_id}") {  
    val id = it.pathParam("student_id")  
    it.json(Student.getOneBy{ Student::student_id eq id } )  
}
```



The screenshot shows a REST client interface with a GET request to `http://localhost:8080/rest/student/id/KUG20220001` and an `Accept: application/json` header. The response is a JSON object representing a student record.

```
http://localhost:8080/rest/student/id/KUG20220001  
  
HTTP/1.1 200 OK  
Date: Sat, 26 Nov 2022 16:38:07 GMT  
Content-Type: application/json  
Content-Length: 151  
Server: Jetty(10.0.11)  
  
{  
  "id": 1,  
  "name": "諸葛淑玲",  
  "birthday": "2002-01-10",  
  "created": "2022-11-27",  
  "gender": "Custom",  
  "height": 179.3,  
  "weight": 53.8,  
  "student_id": "KUG20220001"  
}
```

Simpler Way

First Name

Henry

Liam

Justin

Grid Pro

Complete lazy-load
context menus, drag
editing.

Grant	Andrews	https://vaadin.com/andrews	andrews@vaadin.com
Rogers	Cook	https://vaadin.com/cook	rogers@vaadin.com
Wells	Faulstich	https://vaadin.com/faulstich	wells@vaadin.com
Fischer	Welch	https://vaadin.com/welch	fischer@vaadin.com
Barber	Stevens	https://vaadin.com/stevens	barber@vaadin.com
Lacey	Lloyd	https://vaadin.com/lloyd	lacey@vaadin.com
Ostrowsky	Carroll	https://vaadin.com/carroll	ostrowsky@vaadin.com
Hunt	Mcdonald	https://vaadin.com/mcdonald	hunt@vaadin.com
Crawford	Zimmerman	https://vaadin.com/zimmerman	crawford@vaadin.com
Rachael	Alvarado	https://vaadin.com/alvarado	rachael@vaadin.com
Glenn	Norris	https://vaadin.com/norris	glenn@vaadin.com
Davis	Leach	https://vaadin.com/leach	davis@vaadin.com
Curry	Fox	https://vaadin.com/fox	curry@vaadin.com
Wilkinson	Tyler	https://vaadin.com/tyler	wilkinson@vaadin.com

CRUD

Simplified creation of master-detail views.

Charts

Feature-rich and interactive charting
libraries for Vaadin.

Vaadin Rich Text Editor

- Easy Java and Web Component API
- Text formatting, images and content blocks

Rich Text Editor

Inline image upload
options.

Unsaved changes

Do you want to save or discard
before navigating away?

Confirm Dialog

Customizable confirm dialogs for user
actions.



Board

Layouts for building responsive dashboards
together with Vaadin Charts.



Map

Displaying geographic maps from various
sources.

Vaadin Pro - CRUD

```
dependencies {  
    :  
    implementation("com.vaadin:vaadin-bom:${properties["vaadin_version"]}")  
    implementation("com.vaadin:vaadin-crud-flow:${properties["vaadin_version"]}")  
    :  
    implementation(kotlin("reflect"))  
}
```

Vaadin Pro - CRUD

```
@Route("crud-student", layout = MainLayout::class)
class CRUDStudentView: KComposite() {
    private val root = ui{
        verticalLayout {
            val grid = Grid(Student::class.java)
            val crud = Crud(Student::class.java, grid, createCrud())
            with(crud){
                dataProvider = Student.dataLoader.asDataProvider()
                addSaveListener { it.item.save() }
                addDeleteListener { it.item.delete() }
                Crud.addEditColumn(grid)
                grid.removeColumnByKey("id")
                :
            }
        }
        add(crud)
    }
}

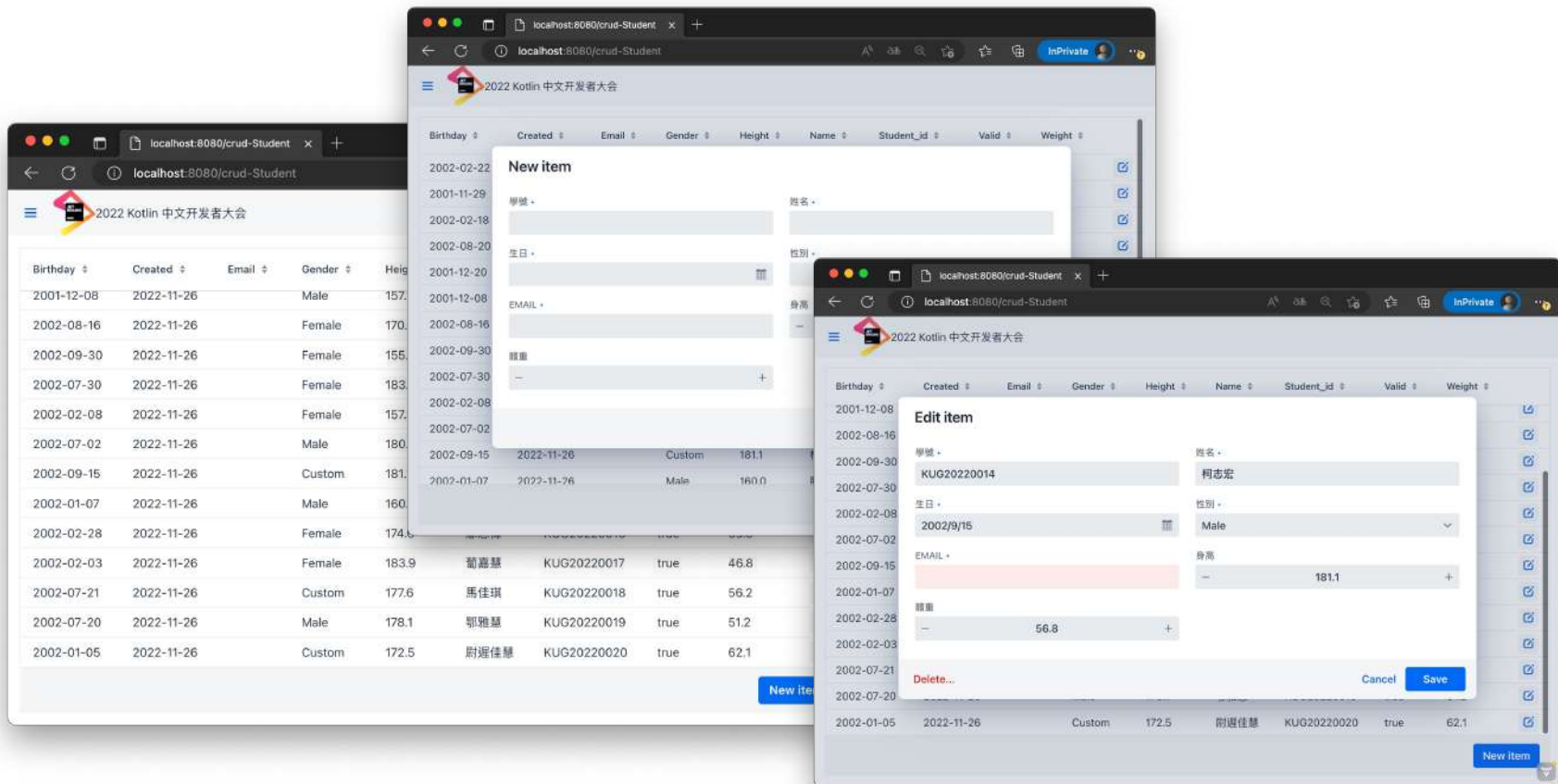
private fun createCrud(): CrudEditor<Student>? {... (next page)}
```

Vaadin Pro - CRUD

```
private fun createCrud(): CrudEditor<Student>? {  
    val studentId = TextField("學號")  
    val name = TextField("姓名")  
    val brithday = DatePicker("生日")  
    :  
    val form = FormLayout(studentId, name, brithday, gender, email, height, weight)  
    val binder = Binder(Student::class.java).also {  
        it.forField(studentId).asRequired()  
            .bind(Student::student_id, Student::student_id.setter)  
        it.forField(name).asRequired().bind(Student::name, Student::name.setter)  
        it.forField(brithday).asRequired().bind(Student::birthday, Student::birthday.setter)  
        :  
    }  
    return BinderCrudEditor(binder, form)  
}
```

implementation(kotlin("reflect"))

CRUD Pro



CRUD Pro



Volkswagen



Resource

- [Vaadin Flow](#)
- [Vaadin Docs](#)
- [Vaadin Components](#)
- [Add-on Directory](#)
- [Vaadin on Kotlin](#)
- [中文讨论区 \(Discussion in Chinese\)](#)
- [Vaadin Hilla](#)
- [Online Tutorial](#)
- [Vaadin Pro](#)



Thanks!
Have a nice Kotlin



Kotlin talk

炉边漫谈

KOTLIN
USER GROUP

