KVision

Pragmatic web development with KVision and Kotlin/JS

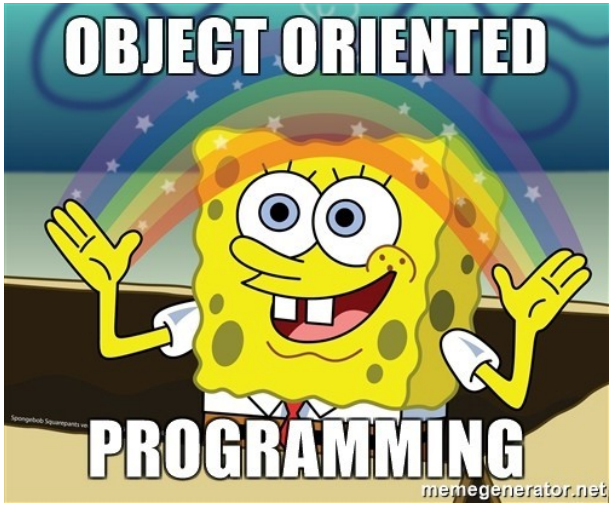
Robert Jaros

04.06.2021
Tricity Kotlin User Group

What is KVision?

An object oriented web framework for Kotlin/JS

Created for both frontend-only and fullstack web applications.



What is KVision?

Old-school UI for the web

similar to MFC, Swing, JavaFX, QT

Modern, reactive approach

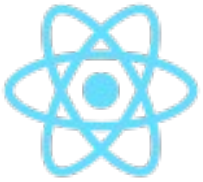
like React, Compose, SwiftUI

JVM

Innovative fullstack interfaces

for Ktor, Spring Boot, Micronaut, Jooby, Javalin and Vert.x

Why KVision?



???

???

???

???

???

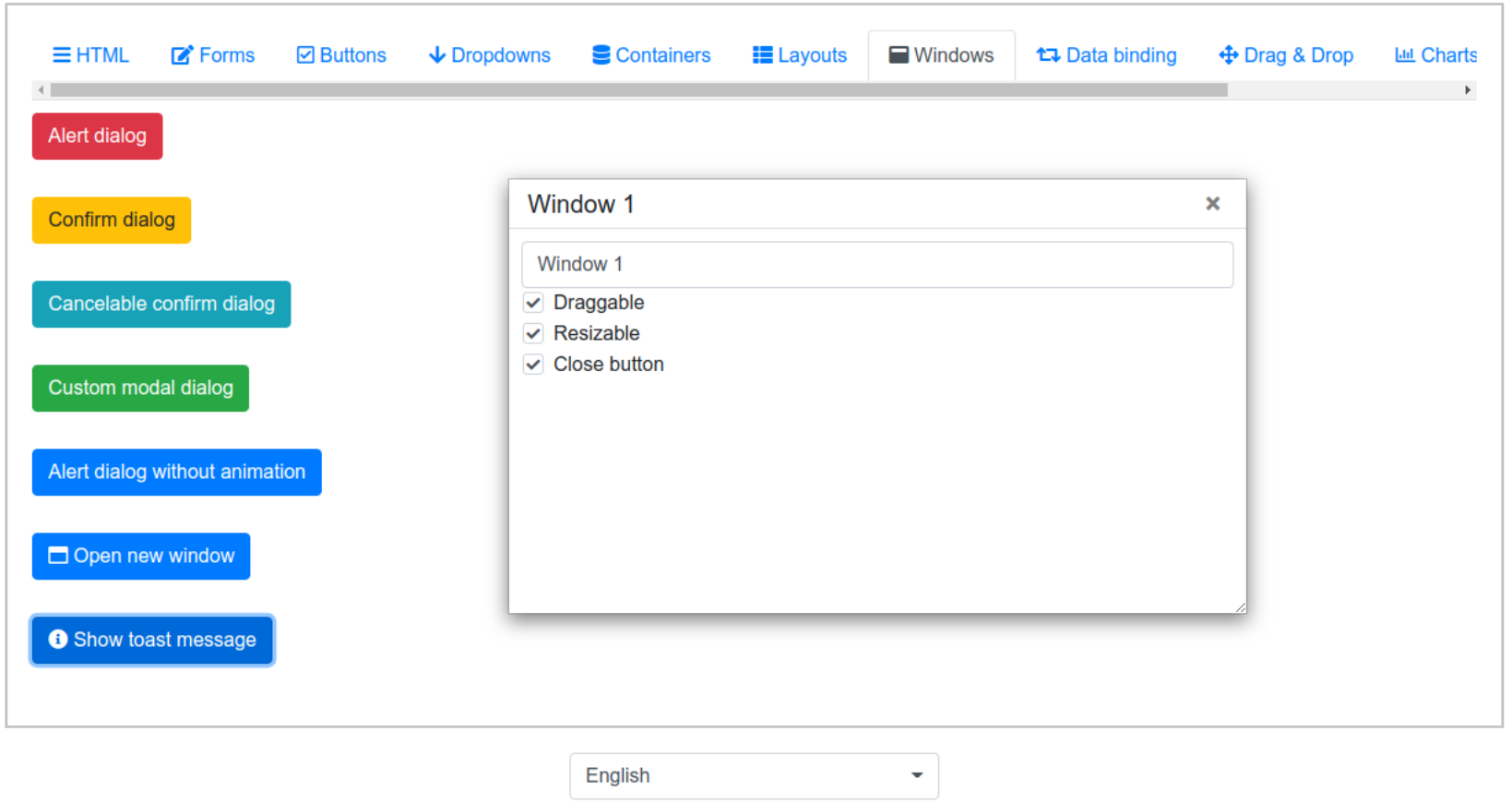
Why KVision?

- Kotlin – compiled and type-safe language
- Efficient UI building with ready to use components
- Flexibility with full support for both reactive and imperative programming models
- „Designed for fullstack” ™

Rich UI components library

- Sophisticated layout containers
- Dedicated form container with type safe data model and built-in validation
- Different text input components including rich text and typeahead
- Date and time picker
- Numeric input components (spinner, slider)
- Advanced select box
- File upload with preview and multi-selection
- Advanced charts
- Reactive tables
- Configurable windows, modals and toasts

The Showcase example app



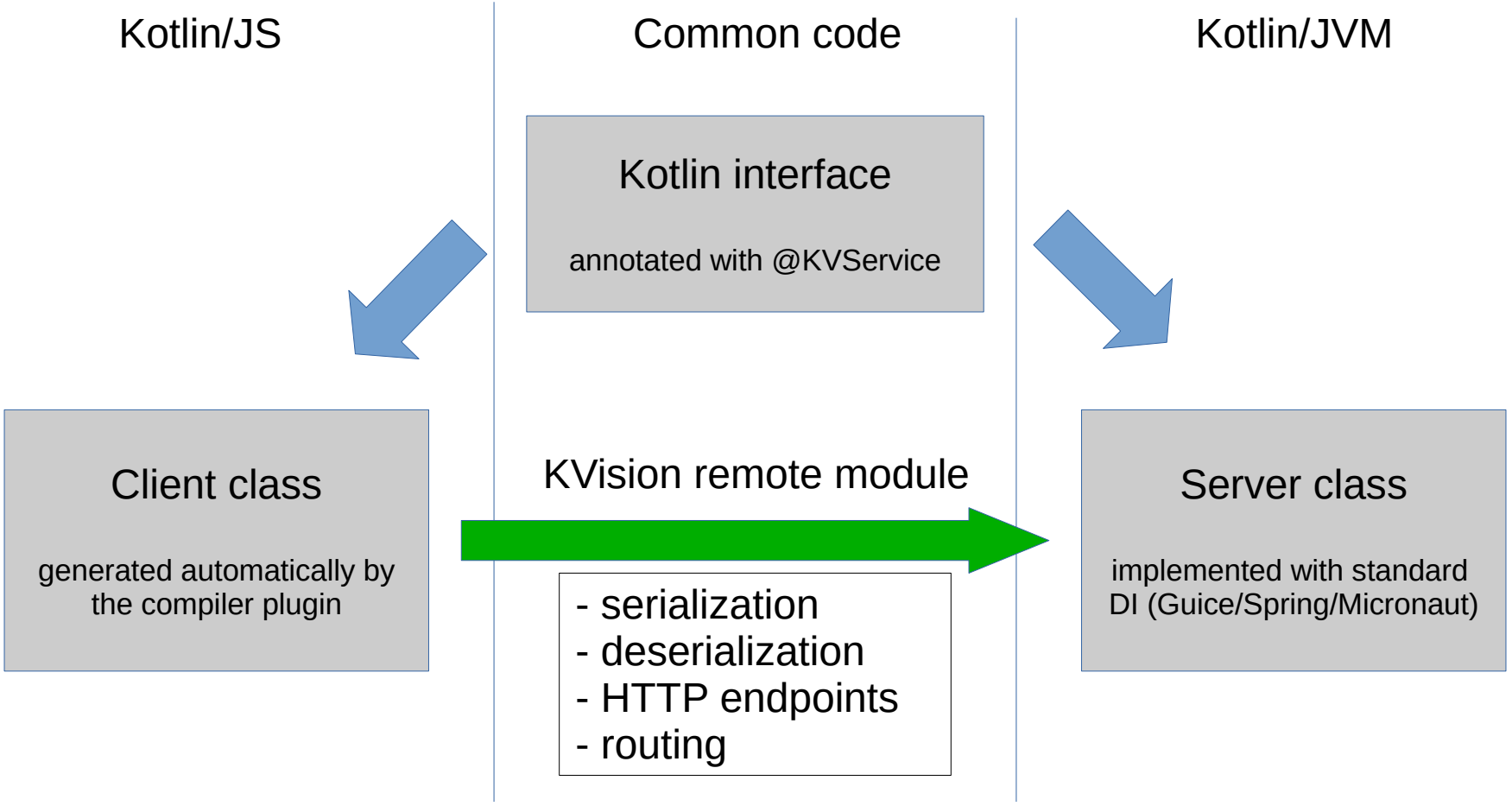
Frontend features

- Support for existing React components
- Mobile web components from Onsen UI
- Localization based on gettext translations
- Font Awesome icons
- JavaScript routing support
- Electron and Cordova support
- Fast development cycle with webpack hot reload

Reactive programming

- Easy to use unidirectional and bidirectional bindings integrated with KVision UI DSL
- Built in observable data structures (ObservableValue, ObservableList, ObservableSet)
- Support for Redux and ReduxKotlin stores
- Support for coroutines StateFlow and SharedFlow
- Optimized, diff-based data bindings for collections

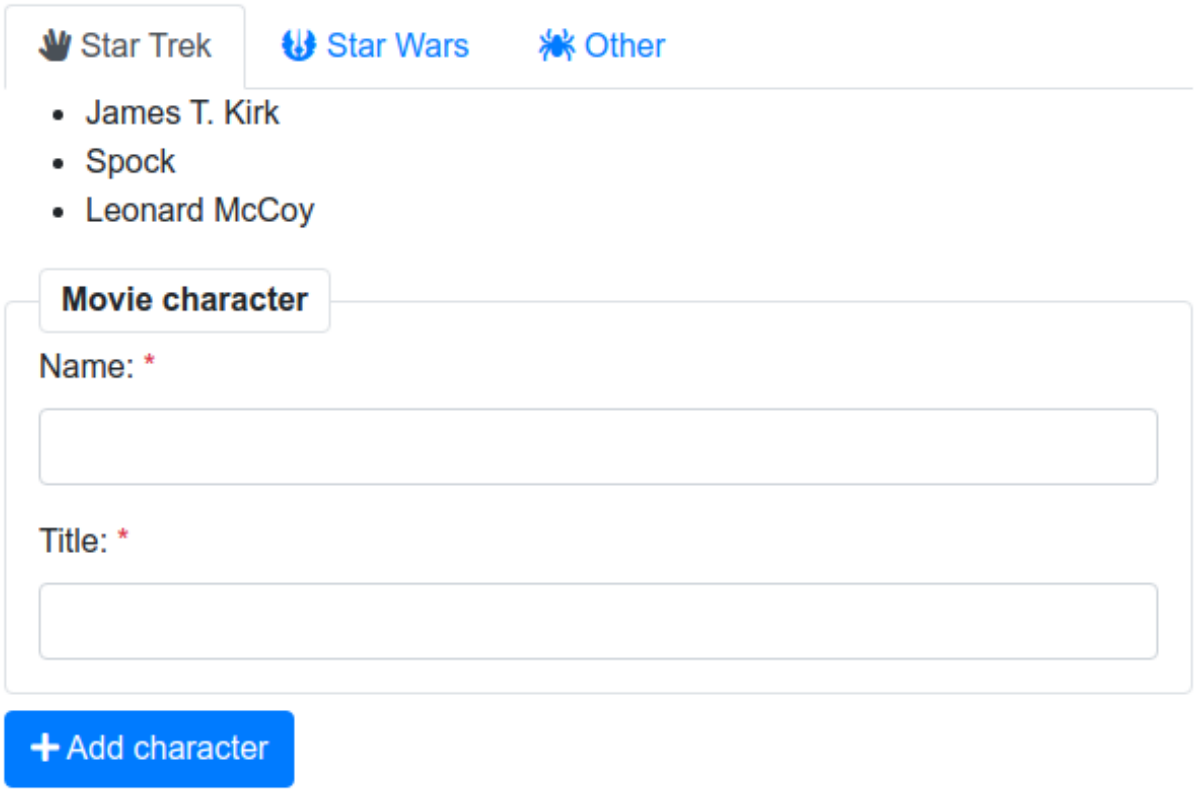
Fullstack interfaces



Fullstack features

- Client-server contracts based on Kotlin interfaces
- Type safety
- Full support for Ktor, Jooby, Spring Boot, Javalin, Vert.x and Micronaut frameworks
- Type safe websockets connections
- No callbacks – everything is based on Kotlin coroutines
- Fully integrated with frontend components (Select, Typeahead, Tabulator)

Ultra fast fullstack example

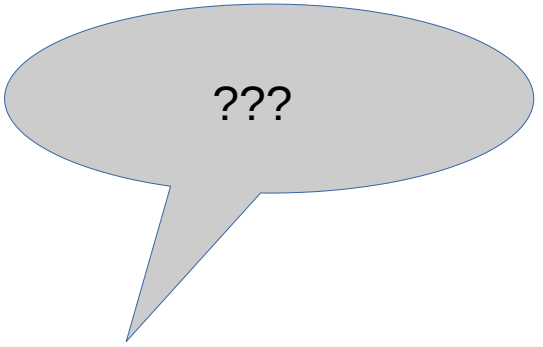


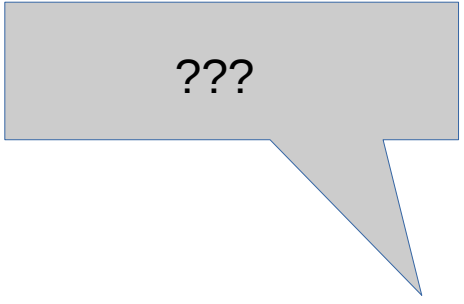
https://github.com/rjaros/tricity-example

More information

- Website **https://kvision.io**
- GitHub page https://github.com/rjaros/kvision
- **#kvision channel on Kotlinlang Slack**
- Blog at Dev.to https://dev.to/t/kvision/latest
- KVision Project Wizard for IntelliJ IDEA
  https://plugins.jetbrains.com/plugin/16533-kvision-project-wizard
- Comprehensive user guide:
  https://kvision.gitbook.io/kvision-guide/
- A lot of different examples, both frontend and fullstack:
  https://github.com/rjaros/kvision-examples

Questions?





Thank you!

Robert Jaros
rjaros@treksoft.pl