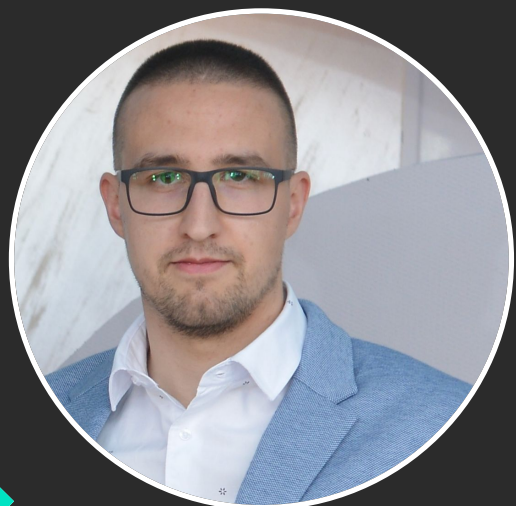




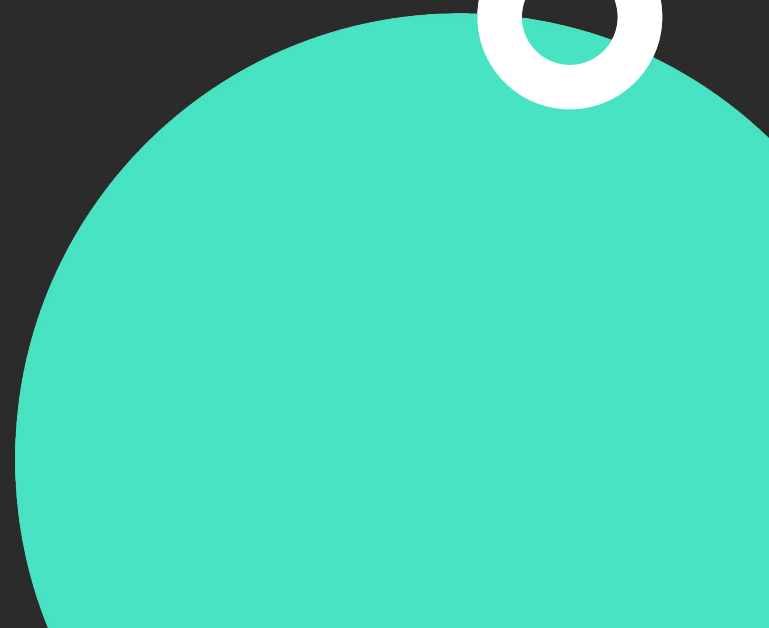
# Getting Started



Stefan Jovanović

✕ @StevdzaS

in in/stevdza-san



# Overview

- . Requirements
- . Introducing with **KMP**
- . Getting familiar with **Ktor**
- . Sending **HTTP** requests with Ktor
- . **Resource plugin** and default requests
- . **Intercepting** and retrying failed requests
- . **Testing** endpoints
- . First Compose Multiplatform app (**Unsplash API**)
- . **Authentication** with Ktor
- . Summary





# Requirements

Up Next



# Requirements

- Basic knowledge about Jetpack Compose
- Basic knowledge about Android Development
- Basic knowledge about Kotlin





# Introducing with KMP

Up Next



# Introducing with KMP

- About KMP technology
- KMP Libraries
- Compose Multiplatform vs KMP
- Flutter vs KMP
- Targets/Source sets
- Expect/Actual mechanism





# Getting familiar with Ktor

Up Next



# Getting familiar with Ktor

- About [Ktor](#) library
- Dependencies
- Project [setup](#)
- Content [serialization](#)







# Sending HTTP requests with Ktor

Up Next



# Sending HTTP requests with Ktor

- HTTP methods
- Path segment
- Query parameters
- Request body
- Text
- Objects





# Resource Plugin and default requests

Up Next



# Resource Plugin and default request

- Type-safe requests
- Resource with query parameters
- Resource with path parameter





# Intercepting and retrying failed requests

Up Next



# Intercepting and retrying failed requests

- Retrying failed requests
- Intercepting requests/responses





# Testing endpoints

Up Next



# Testing endpoints

- Postman







# First Compose Multiplatform App

Up Next



# First Compose Multiplatform App

- Unsplash API
- Ktor Client
- App for iOS and Android





# Authentication with Ktor

Up Next



# Authentication with Ktor

- Authentication vs Authorization
- Tokens



# Summary

- Introduced with a basic concepts of **KMP** technology
- Some of the most useful **features** of Ktor client library for android/iOS.
- Basic **HTTP** requests with Ktor.
- **Resource** plugin to model HTTP requests in a kotlin idiomatic way.
- Properly handling **failed** network requests.
- **Intercepting** HTTP requests.
- Building your first **Compose Multiplatform** app for iOS/Android that utilizes a power of Ktor client.
- Building your second Compose Multiplatform app to handle the user **authentication** on both iOS/Android.





# Thank you!



Stefan Jovanović

✕ @StevdzaS

in in/stevdza-san

