
Spring Webflux (by Coroutine)

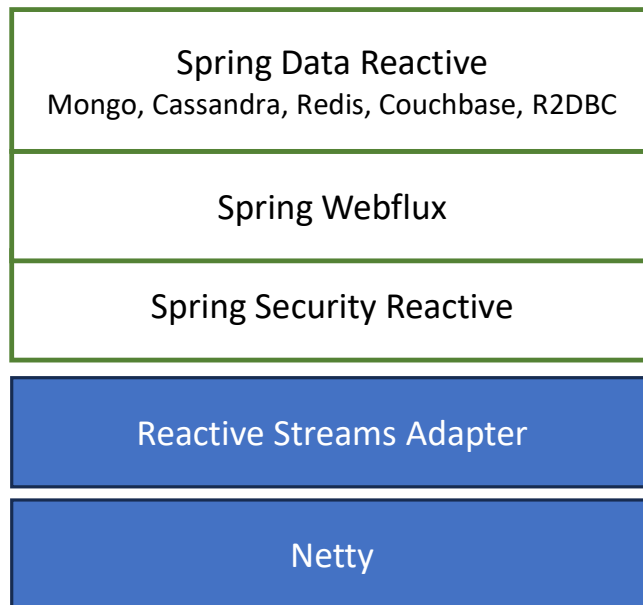
소개 및 환경구성

Spring Webflux

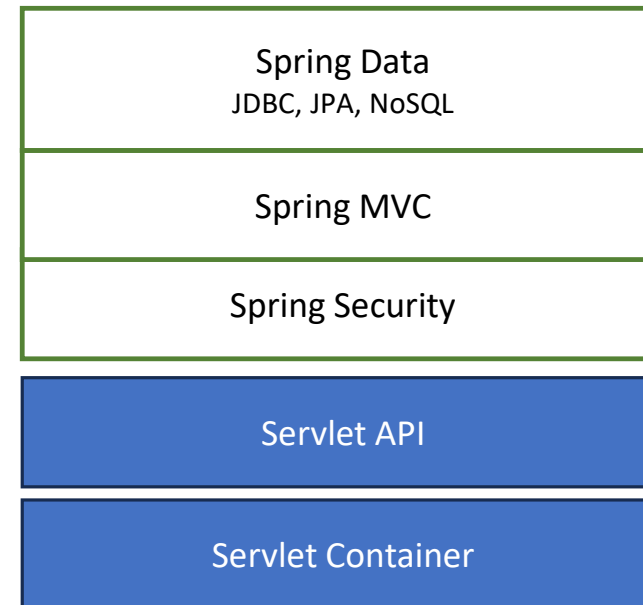
최소한의 H/W 자원으로 동시성을 처리하는 Non-blocking Web stack

First release : 2017.08 (from Spring 5)

Webflux



MVC



구성요소

Netty

Async process

Reactor, Kotlin Coroutine

R2DBC

구성요소

Netty

Async process

Reactor, Kotlin Coroutine

R2DBC

구성요소

Netty

Async process

Reactor, Kotlin Coroutine

R2DBC

from Spring 5.2 (2019.11.04)

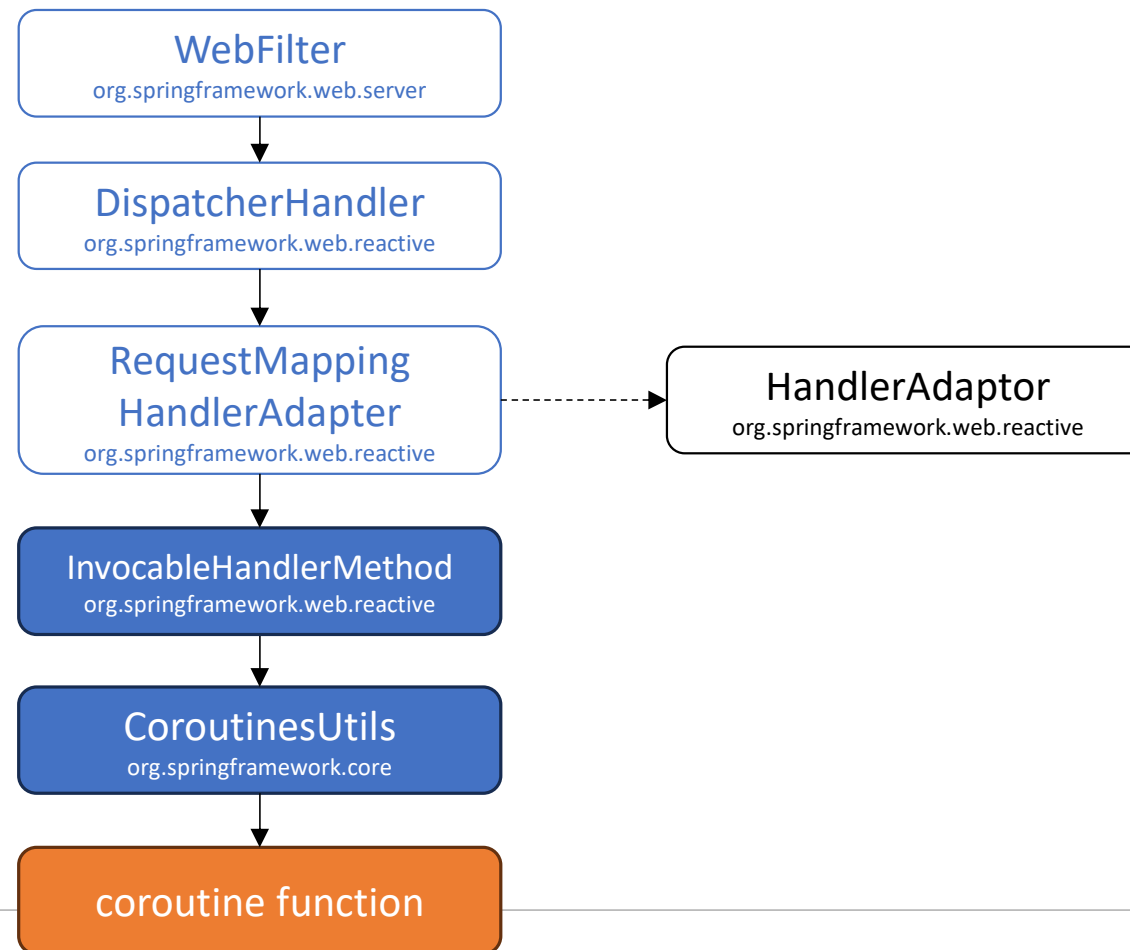
- docs.spring.io/spring-boot/docs/current/reference/html/dependency-versions.html

[< Back to index](#)

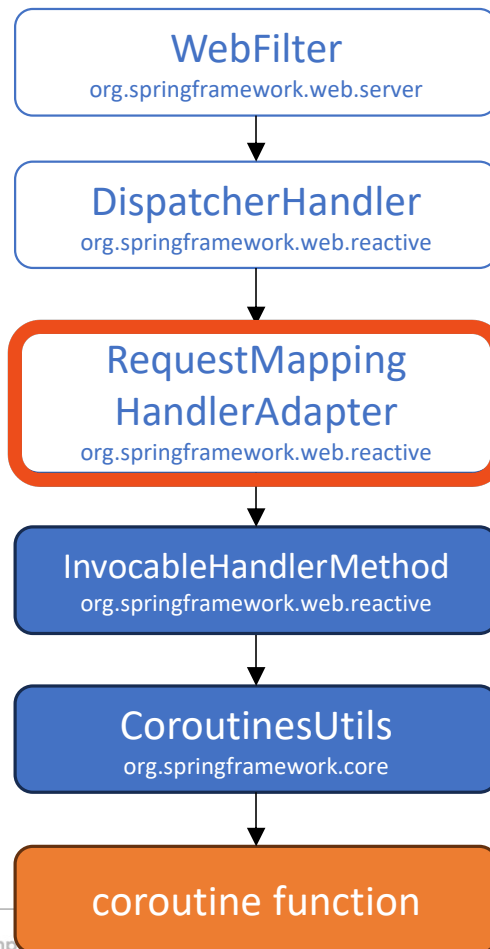
1. Managed Dependency Coordinates
2. Version Properties

org.jetbrains.kotlin	kotlin-test-junit4	1.6.4
org.jetbrains.kotlin	kotlin-test-testng	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-android	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-core	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-core-jvm	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-debug	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-guava	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-javafx	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-jdk8	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-jdk9	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-play-services	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-reactive	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-reactor	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-rx2	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-rx3	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-slf4j	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-swing	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-test	1.6.4
org.jetbrains.kotlinx	kotlinx-coroutines-test-jvm	1.6.4
org.jooq	jooq	3.18.4
org.jooq	jooq-codegen	3.18.4
org.jooq	jooq-kotlin	3.18.4
org.jooq	jooq-meta	3.18.4

Spring Webflux for Kotlin Coroutine

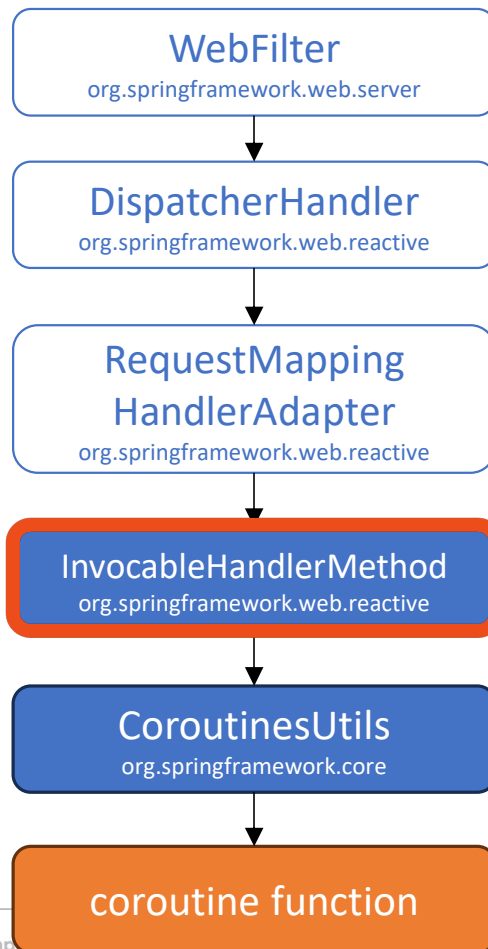


Spring Webflux for Kotlin Coroutine



```
public Mono<HandlerResult> handle(ServerWebExchange exchange, Object handler) {
    HandlerMethod handlerMethod = (HandlerMethod)handler;
    Assert.state(expression: this.methodResolver != null && this.modelInitializer != null, message: "Not initialized");
    InitBinderBindingContext bindingContext = new InitBinderBindingContext(this.getWebBindingInitializer(), this);
    InvocableHandlerMethod invocableMethod = this.methodResolver.getRequestMappingMethod(handlerMethod);
    DispatchExceptionHandler exceptionHandler = (exchange2, ex) -> {
        return this.handleException(exchange2, ex, handlerMethod, bindingContext);
    };
    return this.modelInitializer.initModel(handlerMethod, bindingContext, exchange).then(Mono.defer(() -> {
        return invocableMethod.invoke(exchange, bindingContext, new Object[0]);
    })).doOnNext((result) -> {
        result.setExceptionHandler(exceptionHandler);
    }).doOnNext((result) -> {
        bindingContext.saveModel();
    }).onErrorResume((ex) -> {
        return exceptionHandler.handleError(exchange, ex);
    });
}
```

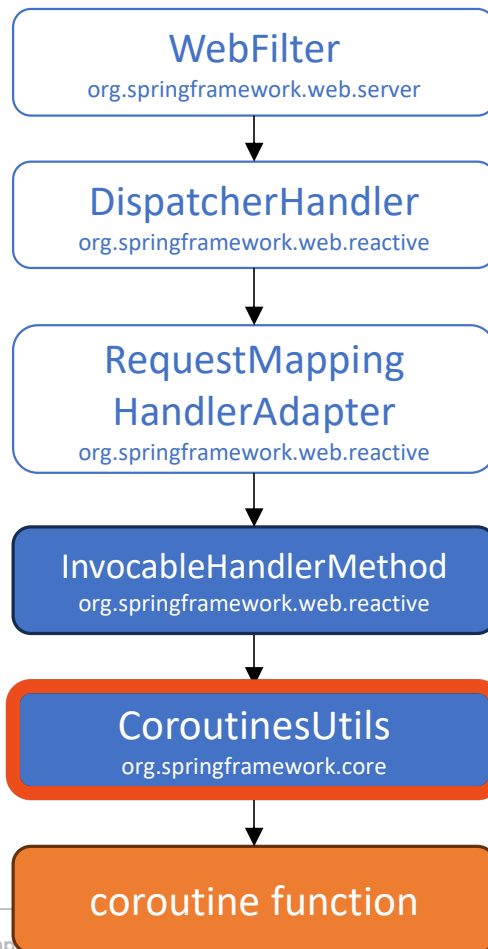

Spring Webflux for Kotlin Coroutine



```
public Mono<HandlerResult> invoke(
    ServerWebExchange exchange, BindingContext bindingContext, Object... providedArgs) {

    return getMethodArgumentValues(exchange, bindingContext, providedArgs).flatMap(args -> {
        Object value;
        Method method = getBridgedMethod();
        boolean isSuspendingFunction = KotlinDetector.isSuspendingFunction(method);
        try {
            if (isSuspendingFunction) {
                value = CoroutinesUtils.invokeSuspendingFunction(method, getBean(), args);
            }
            else {
                value = method.invoke(getBean(), args);
            }
        }
        catch (IllegalArgumentException ex) {
            assertTargetBean(getBridgedMethod(), getBean(), args);
            String text = (ex.getMessage() != null ? ex.getMessage() : "Illegal argument");
            return Mono.error(new IllegalStateException(formatInvokeError(text, args), ex));
        }
        catch (InvocationTargetException ex) {
            return Mono.error(ex.getTargetException());
        }
        catch (Throwable ex) {
            // Unlikely to ever get here, but it must be handled...
            return Mono.error(new IllegalStateException(formatInvokeError(text: "Invocation failure", args), ex));
        }
    });
}
```

구현내용 및 개발환경 구성



```
public static Publisher<?> invokeSuspendingFunction(CoroutineContext context, Method method, Object target, Object... args) {
    Assert.isTrue(KotlinDetector.isSuspendingFunction(method), "message: 'method' must be a suspending function");
    KFunction<?> function = (KFunction)Objects.requireNonNull(ReflectJvmMapping.getKotlinFunction(method));
    if (method.isAccessible() && !KCallablesJvm.isAccessible(function)) {
        KCallablesJvm.setAccessible(function, value: true);
    }

    Mono<Object> mono = MonoKt.mono(context, (scope, continuation) -> {
        return KCallables.callSuspend(function, getSuspendedFunctionArgs(method, target, args), continuation);
    }).filter((result) -> {
        return !Objects.equals(result, Unit.INSTANCE);
    }).onErrorMap(InvocationTargetException.class, InvocationTargetException::getTargetException);
    KClassifier returnType = function.getReturnType().getClassifier();
    if (returnType != null) {
        if (returnType.equals(JvmClassMappingKt.getKotlinClass(Flow.class))) {
            return mono.flatMapMany(CoroutinesUtils::asFlux);
        }
    }
}
```

구현내용 및 개발환경 구성

구현내용

자료구조

```
Article
{
  ** id : 12345
  title : "제목"
  body: "특별한 내용 없습니다."
  authorId : 967782
  createdAt : 2023-07-12T12:23:56
  updatedAt : 2023-07-12T12:23:56
}
```

API

기능	METHOD	URI
등록	post	/article
수정	put	/article/{id}
삭제	delete	/article/{id}
<u>단건</u> 조회	get	/article/{id}
목록 조회	get	/article/all

구현내용 및 개발환경 구성

Spring initializr

- <https://start.spring.io>

<https://start.spring.io/#type=gradle-project&language=kotlin&platformVersion=3.1.2&packaging=jar&jvmVersion=17&groupId=dev.fastcampus&artifactId=webflux.coroutine&name=webflux.coroutine&description=spring%20boot%20coroutine&packageName=dev.fastcampus.webflux.coroutine&dependencies=webflux,data-r2dbc,h2,mariaadb>



Project

☐ Gradle - Groovy
 ☒ Gradle - Kotlin
 ☐ Java
 ☒ Kotlin
 ☐ Groovy

☐ Maven

Spring Boot

☐ 3.2.0 (SNAPSHOT)
 ☐ 3.2.0 (M1)
 ☐ 3.1.3 (SNAPSHOT)
 ☒ 3.1.2

☐ 3.0.10 (SNAPSHOT)
 ☐ 3.0.9
 ☐ 2.7.15 (SNAPSHOT)
 ☐ 2.7.14

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 20 ☒ 17 ☐ 11 ☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Reactive Web WEB

Build reactive web applications with Spring WebFlux and Netty.

Spring Data R2DBC SQL

Provides Reactive Relational Database Connectivity to persist data in SQL stores using Spring Data in reactive applications.

H2 Database SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

MariaDB Driver SQL

MariaDB JDBC and R2DBC driver.

Spring Webflux (by Coroutine)

서비스 구현

구현 실습

Hello world API 구현

구현 실습

CRUD 서비스 구현

구현 실습

Unit 테스트 (Kotest)

Unit 테스트 (Kotest)

Kotest

- <https://kotest.io>
- <https://kotest.io/docs/extensions/spring.html>