

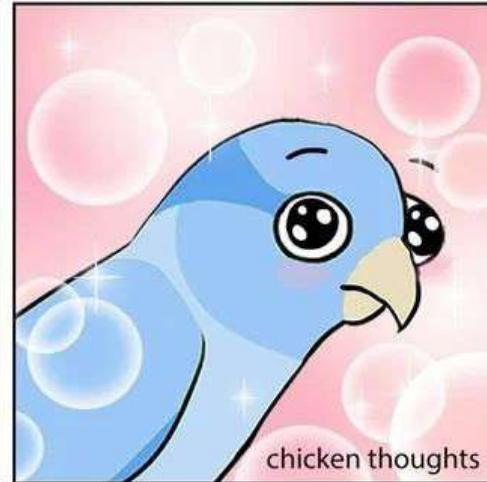
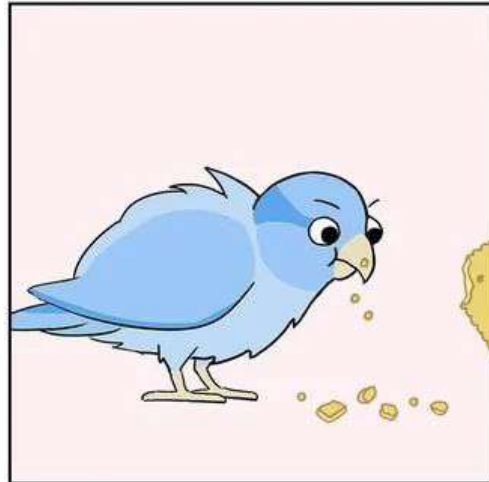


用 Kotlin 讓 Java 踏入微服務的時代

Brandy Chang



About Me



2023 - JCCConf : Arrow KT

TW Kotlin 年會

- ITHome 鐵人賽

- KT Meetup #19


2022 - ITHome 鐵人賽

2021 - Kotlin 讀書會

JCCConf : Reactive with Quarkus



Server Side Kotlin

 **spring** initializr

Project
☒ Gradle - Groovy ☐ Gradle - Kotlin ☐ Maven

Language
☐ Java ☒ Kotlin ☐ Groovy

Spring Boot
☐ 3.2.1 (SNAPSHOT) ☒ 3.2.0 ☐ 3.1.7 (SNAPSHOT) ☐ 3.1.6

Project Metadata

 **QUARKUS** 3.6 io.quarkus.platform ◻ ◀ Back

CONFIGURE YOUR APPLICATION

Group	org.acme	Version	1.0.0-SNAPSHOT
Artifact	code-with-quarkus	Java Version	17 ◻
Build Tool	Maven ◻	Starter Code	Yes ☑ CLOSE

Filters ◻

Extensions found by origin: ✓ 8 in platform 0 in other

☒ **RESTEasy Reactive Kotlin Serialization** [quarkus-resteasy-reactive-kotlin-serialization] 🔗 STARTER CODE

Kotlin Serialization support for RESTEasy Reactive io.quarkus:quarkus-resteasy-reactive-kotlin-serialization is-resteasy extension, or an

☐ **RESTEasy Reactive's REST Client Kotlin Serialization** [quarkus-rest-client-reactive-kotlin-serialization] 🔗

Kotlin serialization support for REST Client Reactive

Migrating From Python to Kotlin for Our Backend Services

📅 May 4, 2021 ⌚ 14 Minute Read 📖 Backend ❤️ 255

Google's Journey from Java to Kotlin for Server Side Programming by James Ward , Brad Hawkes , John

觀看次數 : 3.6萬次 • 1 年前



Devoxx

Kotlin is now a recommended programming language for server-side JVM usage at Google, while sti

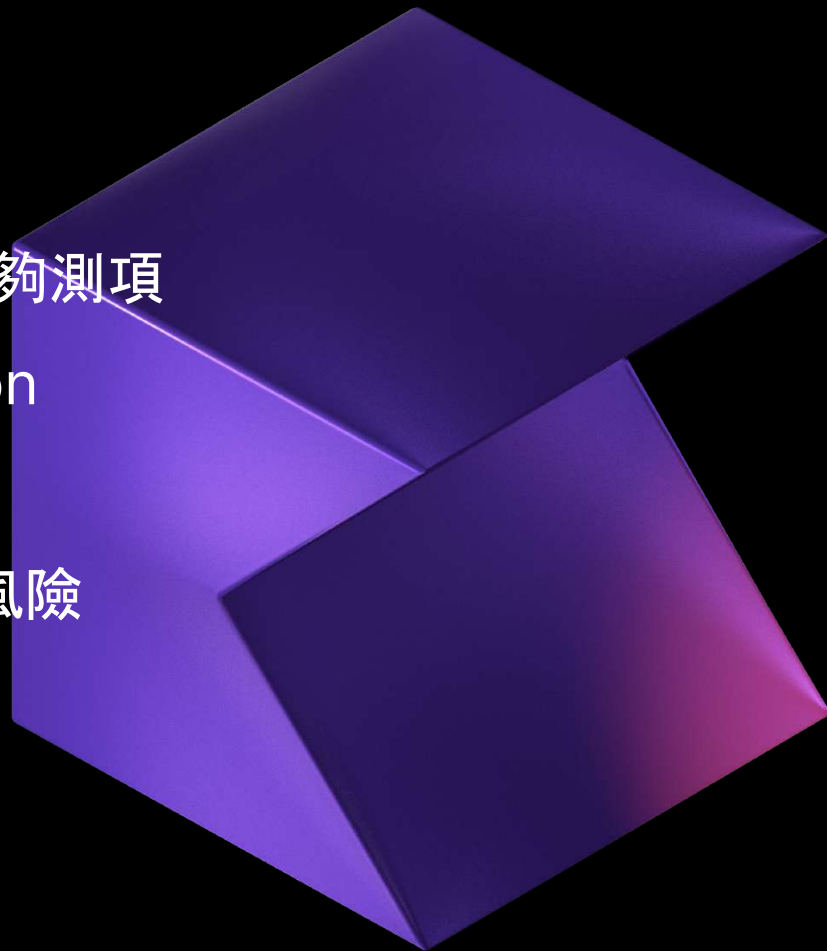


James Ward | Java at Google | Multi-Platform Ui | Single Version Policy | W... 21 個段落



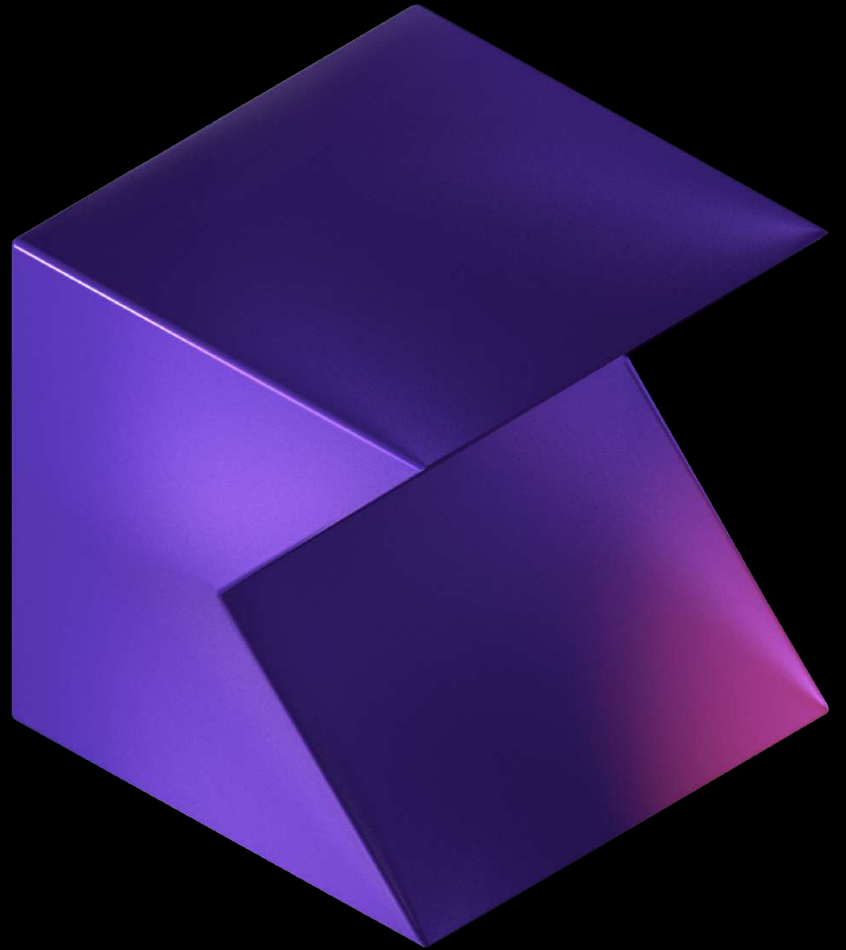
舊 Java 專案的困境

- 舊程式碼有其商業價值但沒有足夠測項
- 不預期的 Null Pointer Exception
- 不一致且重覆的設計模式實現
- 新的語法簡潔，但升級 Java 是風險
- 新的語言有一定的學習曲線





1. 用 Kotlin 幫 Java 專案加上測試



100% 與 Java 互操作, 更簡潔

Before:

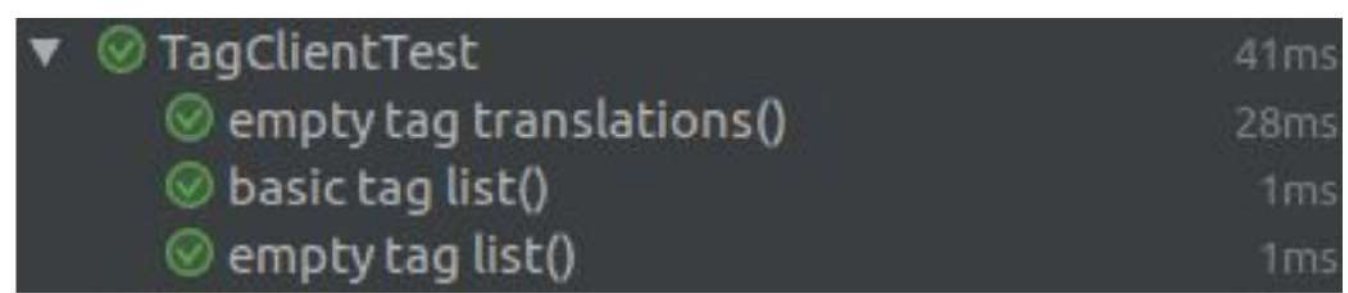
```
assertEquals(request.getAmount(), purchase.getAmount());
assertEquals(request.getShopConfigurationId(), purchase.getConfiguration().getId());
assertEquals(request.getWalletId(), purchase.getWallet().getPublicId());
assertEquals(request.getDescription(), purchase.getDescription());
assertEquals(request.getExternalReference(), purchase.getExternalReference());
```

After:

```
with(purchase) {
    assertEquals(request.amount, amount)
    assertEquals(request.shopConfigurationId, configuration.id)
    assertEquals(request.walletId, wallet.publicId)
    assertEquals(request.description, description)
    assertEquals(request.externalReference, externalReference)
}
```

Test Case Name

```
class TagClientTest {  
    @Test  
    fun `basic tag list`() {}  
    @Test  
    fun `empty tag list`() {}  
}
```



▼	✓ TagClientTest	41ms
	✓ empty tag translations()	28ms
	✓ basic tag list()	1ms
	✓ empty tag list()	1ms

JSON, XML
可讀性↑↑

```
@Test
fun `test add`() {
    val title = "Sample Title"
    Given { this: RequestSpecification
        contentType("application/json")
        body(
            """
                {"widget": {
                    "window": {
                        "title": "$title",
                        "width": 500,
                        "height": 500
                    },
                    "text": {
                        "size": 36,
                        "name": "text1",
                        "hoffset": 250,
                        "voffset": 100,
                        "alignment": "center"
                    }
                }
            """
        ) ^Given
    } When { this: RequestSpecification
        post("/films")
    }
```

```
@Test
fun `test add`() {
    val title = "Sample Title"
    Given { this: RequestSpecification
        contentType("application/json")
        body(
            """
                {"widget\\": {\\n" +
                "  \\\"window\\": {\\n" +
                "    \\\"title\\": \\\"$title\\\",\\n" +
                "    \\\"width\\": 500,\\n" +
                "      \\\"height\\": 500\\n" +
                "    },\\n" +
                "    \\\"text\\": {\\n" +
                "      \\\"size\\": 36,\\n" +
                "      \\\"name\\": \\\"text1\\\",\\n" +
                "      \\\"hoffset\\": 250,\\n" +
                "      \\\"voffset\\": 100,\\n" +
                "      \\\"alignment\\": \\\"center\\\"\\n" +
                "    }\\n" +
                "  }},
            """
        ) ^Given
    } When { this: RequestSpecification
        post("/films")
    }
```



MockK



```
mockk(relaxed=true)
```

```
val clientMock: UserClient = mockk()  
every { clientMock.getUser(any()) }  
    returns User(id = 1, name = "Ben")
```

```
val updater = UserUpdater(clientMock)  
updater.updateUser(1)
```

```
verify { clientMock.getUser(1) }
```



Constructor Injection for Spring-free Testing

@Component

```
class DesignController(  
    private val designRepo: DesignRepository,  
    private val designClient: DesignClient,  
) {}
```

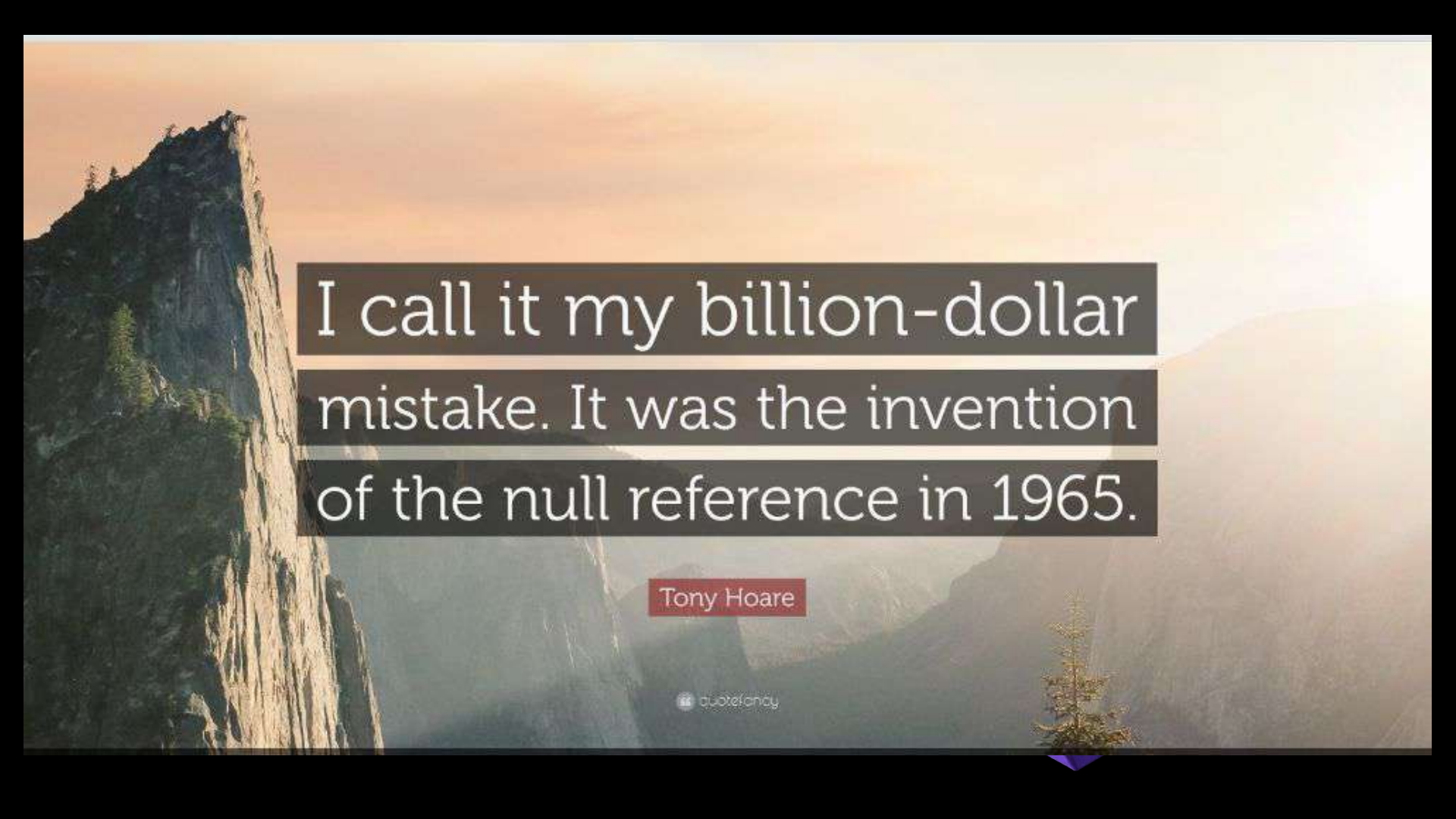
Easy to test Logic without Spring:

```
val repo: DesignRepository = mockk()  
val client: DesignClient = mockk()  
val controller = DesignController(repo, client)
```


用 Kotlin 幫 Java 專案加上測試

- 更人性化的語法
- 提高可維護性





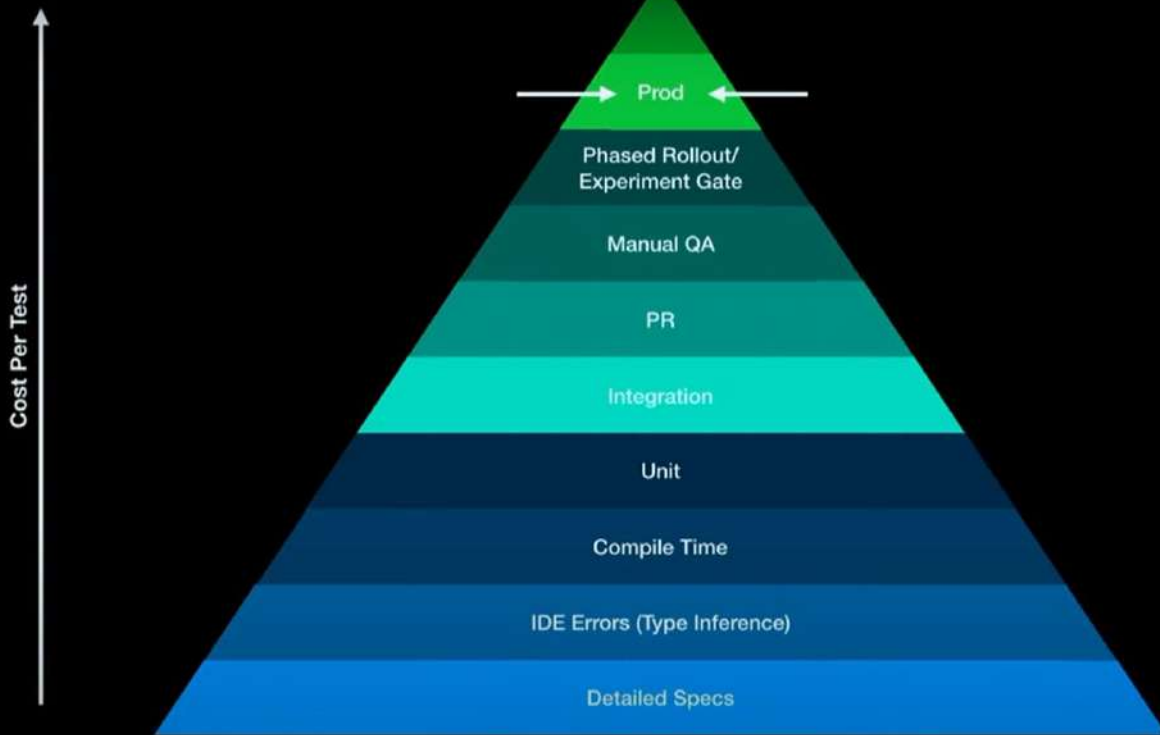
I call it my billion-dollar
mistake. It was the invention
of the null reference in 1965.

Tony Hoare

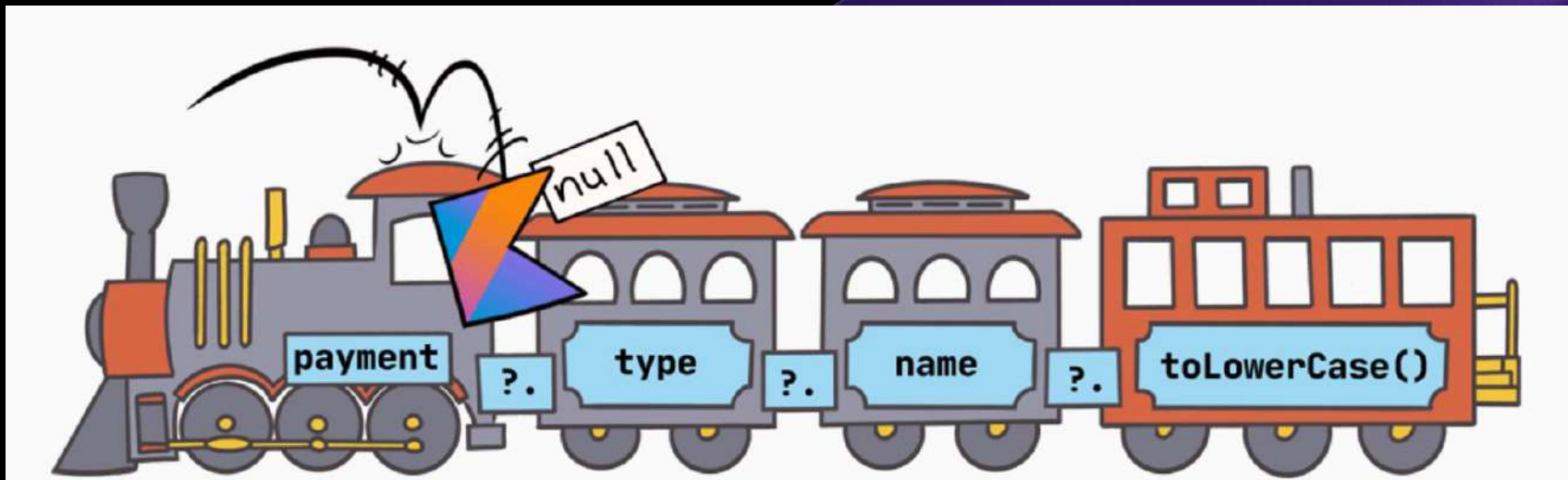
Null Pointer Exception

面對他
接受他
處理他
放下他

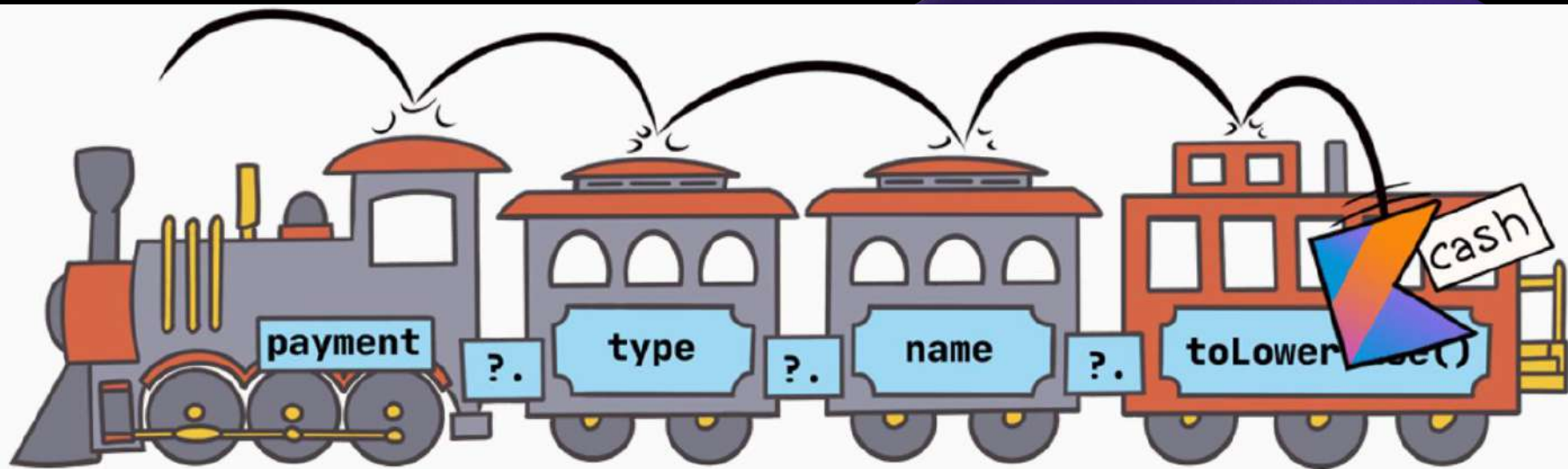
KotlinConf 2019: Testing: Testing by Christina Lee



```
val supportType = payment?.type?.name?.toLowerCase()
```



```
val supportType = payment?.type?.name?.toLowerCase()
```



NULL SAFETY IN KOTLIN

<https://typealias.com/>

val payment: Payment?



When payment is not null...



When payment is null...



SMART CAST

```
if (payment != null) {  
    orderCoffee(payment)  
}
```



No coffee
for me today...



ELVIS OPERATOR

```
orderCoffee(  
    payment ?: getFromFriend()  
)
```



NOT-NULl ASSERTION

```
orderCoffee(payment!!)
```

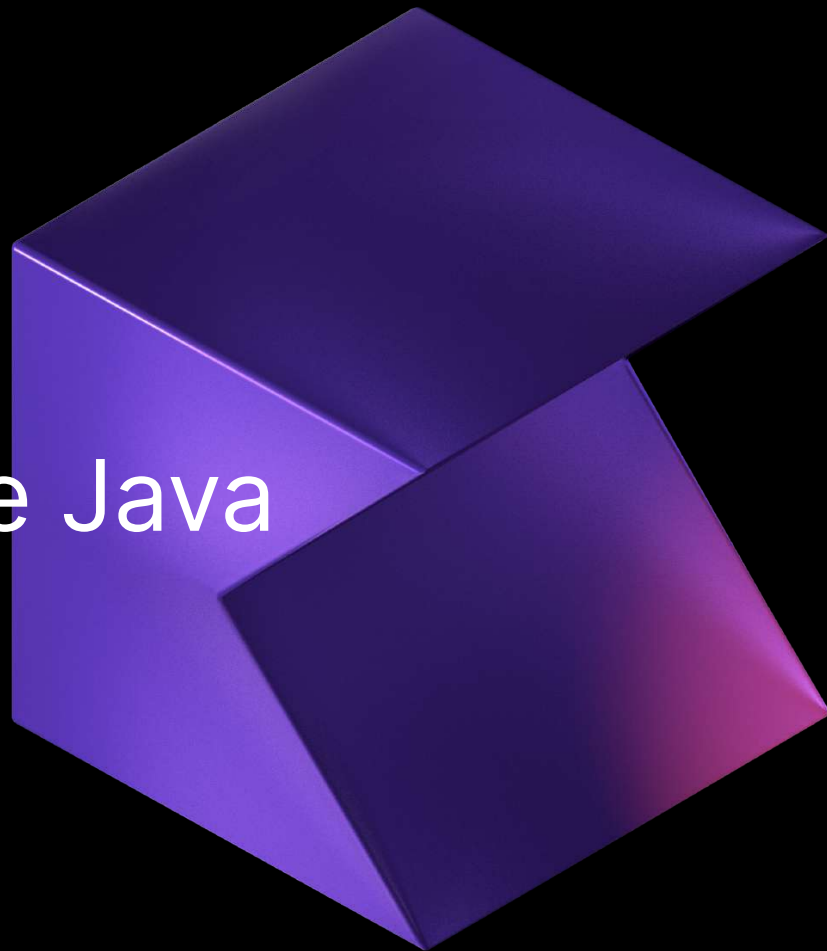


KA-BOOM!

typealias.com

不一致且重覆的設計模式實現

Kotlin 內建 Effective Java



equals A Java Trap

ME, A JAVA DEVELOPER, LEARNING C++ AND
FINDING OUT YOU CAN USE == TO COMPARE STRINGS

Wait. That's illegal.

Avoid Java Common Trap

- No more equals
- split 表明要用 regex 還是要分隔符

```
"path/to/somewhere/file.txt".split|
```

```
f split(regex: Regex, limit: Int = ...) for Char... List<String>
```

```
f split(regex: Pattern, limit: Int = ...) for Ch... List<String>
```

```
f split(vararg delimiters: Char, ignoreCase: Boo... List<String>
```

```
f split(vararg delimiters: String, ignoreCase: B... List<String>
```

```
f splitToSequence(regex: Regex, limit: Int =... Sequence<String>
```

```
f splitToSequence(vararg delimiters: Char, i... Sequence<String>
```

```
f splitToSequence(vararg delimiters: String,... Sequence<String>
```

No More Common-Lang, Common-Lang3

Path, FileName, URLs

```
"path/to/somewhere/file.txt".subString
```

<code>f substring</code>	<code>(range: IntRange) for String in kotlin.text</code>	<code>String</code>
<code>f substring</code>	<code>(startIndex: Int) for String in kotlin.text</code>	<code>String</code>
<code>f substring</code>	<code>(startIndex: Int, endIndex: Int) for String</code>	<code>String</code>
<code>f substringAfter</code>	<code>(delimiter: Char, missingDelimiterValu...</code>	<code>String</code>
<code>f substringAfter</code>	<code>(delimiter: String, missingDelimiterVa...</code>	<code>String</code>
<code>f substringAfterLast</code>	<code>(delimiter: Char, missingDelimiter...</code>	<code>String</code>
<code>f substringAfterLast</code>	<code>(delimiter: String, missingDelimit...</code>	<code>String</code>
<code>f substringBefore</code>	<code>(delimiter: Char, missingDelimiterVal...</code>	<code>String</code>
<code>f substringBefore</code>	<code>(delimiter: String, missingDelimiterV...</code>	<code>String</code>
<code>f substringBeforeLast</code>	<code>(delimiter: Char, missingDelimite...</code>	<code>String</code>

用 Extension Function 取代 Utils

```
object Mapper {  
    val mapper: ObjectMapper = ObjectMapper() ...  
  
    fun Any.toJson(): String =  
        mapper.writeValueAsString(this)  
  
    inline fun <reified T> String.fromJson(): T =  
        mapper.readValue(this, T::class.java)  
}
```

```
data class User(val name: String, val family: String)
```

```
val jsonString = mapOf("name" to "Brandy", "family" to "Chang").toJson()
```

```
val user = jsonString.fromJson<User>()
```

kotlin

```
object Singleton {  
    fun doSomething() {  
        // 實現細節  
    }  
}  
  
fun main() {  
    Singleton.doSomething()  
}
```

Singleton Pattern

java

```
public class Singleton {  
    private static volatile Singleton instance;  
  
    private Singleton() {}  
  
    public static Singleton getInstance() {  
        if (instance == null) {  
            synchronized (Singleton.class) {  
                if (instance == null) {  
                    instance = new Singleton();  
                }  
            }  
        }  
        return instance;  
    }  
  
    public void doSomething() {  
        // 實現細節  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Singleton.getInstance().doSomething();  
    }  
}
```

kotlin

```
interface Base {  
    fun print()  
}  
  
class BaseImpl(val x: Int) : Base {  
    override fun print() { print(x) }  
}  
  
class Derived(b: Base) : Base by b  
  
fun main() {  
    val b = BaseImpl(10)  
    Derived(b).print() // 輸出 10  
}
```

java

```
interface Base {  
    void print();  
}  
  
class BaseImpl implements Base {  
    private int x;  
  
    BaseImpl(int x) { this.x = x; }  
  
    @Override  
    public void print() { System.out.println(x); }  
}  
  
class Derived implements Base {  
    private Base b;  
  
    Derived(Base b) { this.b = b; }  
    //需要為每個委派的方法提供明確的轉發  
    @Override  
    public void print() {  
        b.print(); // 明確轉發  
    }  
}
```

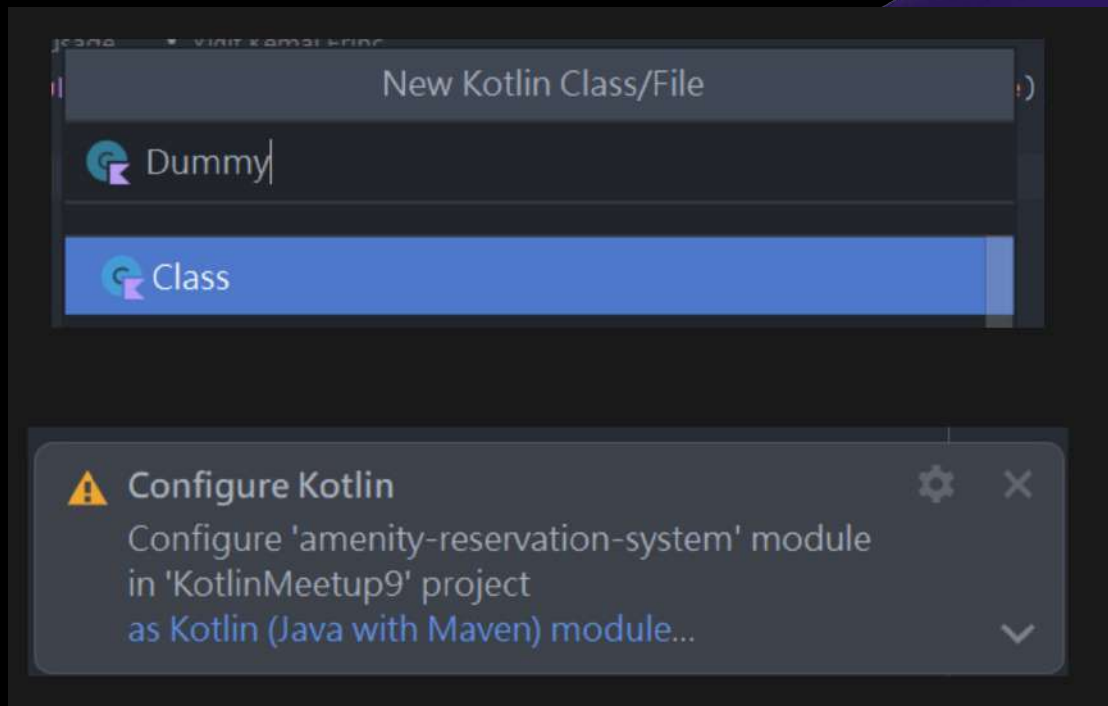
Delegation Pattern

- ★ 新的語法簡潔，但升級 Java 是風險
 - Kotlin 支援 Java 8 以上的專案

Which versions of JVM does Kotlin target?

Kotlin lets you choose the version of JVM for execution. By default, the Kotlin/JVM compiler produces Java 8 compatible bytecode. If you want to make use of optimizations available in newer versions of Java, you can explicitly specify the target Java version from 9 to 21. Note

- ★ 新的語言有一定的學習曲線
 - IDE 幫助及相容漸近的轉換



自動把 Java Code 轉成 Kotlin

```
@Autowired
lateinit var jdbcTemplate: JdbcTemplate
public MemberAccount findMemberAccountByUsername(String username) {
    // TODO Auto-generated method stub
    String sql = " SELECT "
                + "      ID, USERNAME, PASSWORD, SALT "
                + " FROM "
```



Convert Code From Java



Clipboard content seems to be Java code. Do you want to convert it to Kotlin?



Don't show this dialog next time

Yes

No

```
return null;
```

```
}
```

```
tyRowMapper<MemberAccount>()
```

手把手指導更好的實現

```
return if (result.size > 0) {
```

```
    result[0]
```

```
} else null
```


Size check can be replaced with 'isEmpty()'

Replace size check with 'isEmpty' Alt+Shift+Enter More

```
public abstract val size: Int
```

Returns the size of the collection.

```
kotlin.collections.List
```

 Maven: org.jetbrains.kotlin:kotlin-stdlib:1.8.10 (kotlin

Google Server Side Kotlin 導入調查結果

70%

Of devs whose
primary language
is Kotlin say that
Kotlin makes them
more productive

87%

Of Google
Server-side Kotlin
Developers are
"very satisfied"
with Kotlin

20%

Less likely to crash
(Android apps built
with Kotlin)

0%

Degradation of
productivity in
mixed Java +
Kotlin codebase

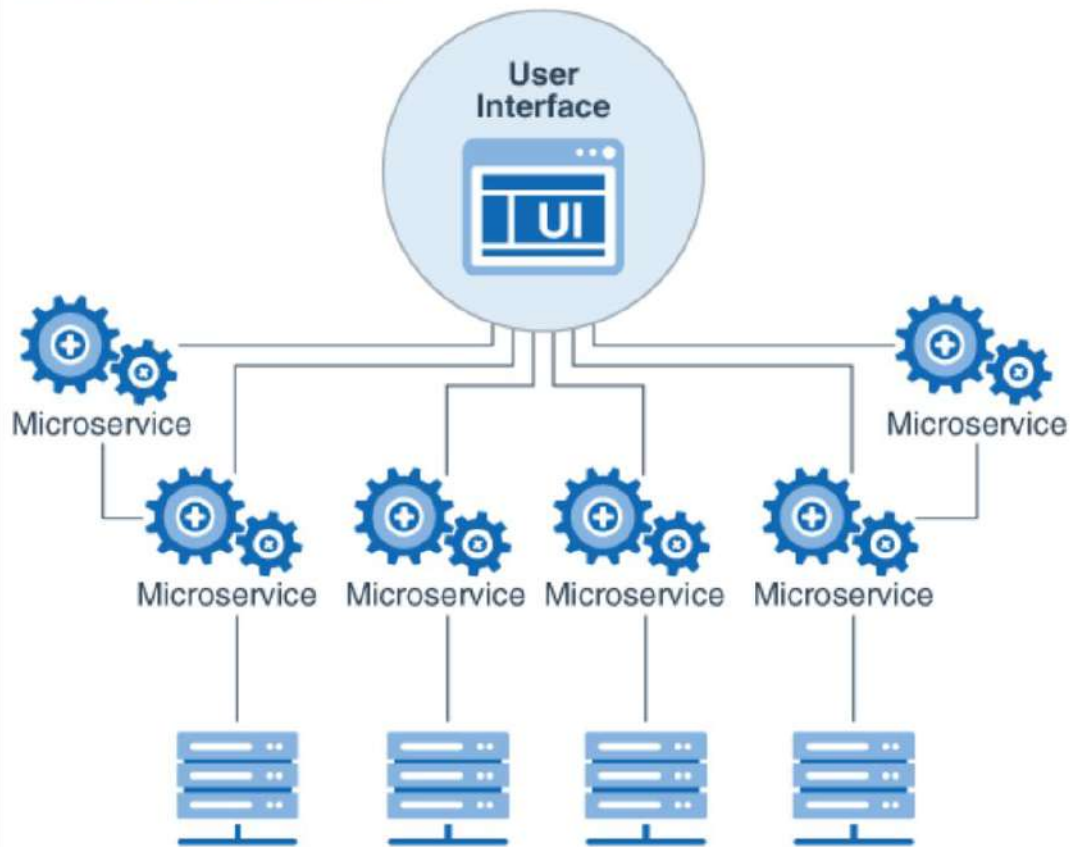
老瓶新裝、漸進導入

- 約 Java 的 $\frac{1}{3}$ - $\frac{2}{3}$ 程式碼
- 讓開發者使用新的語法
- 可維護性提高
- 生產力提高

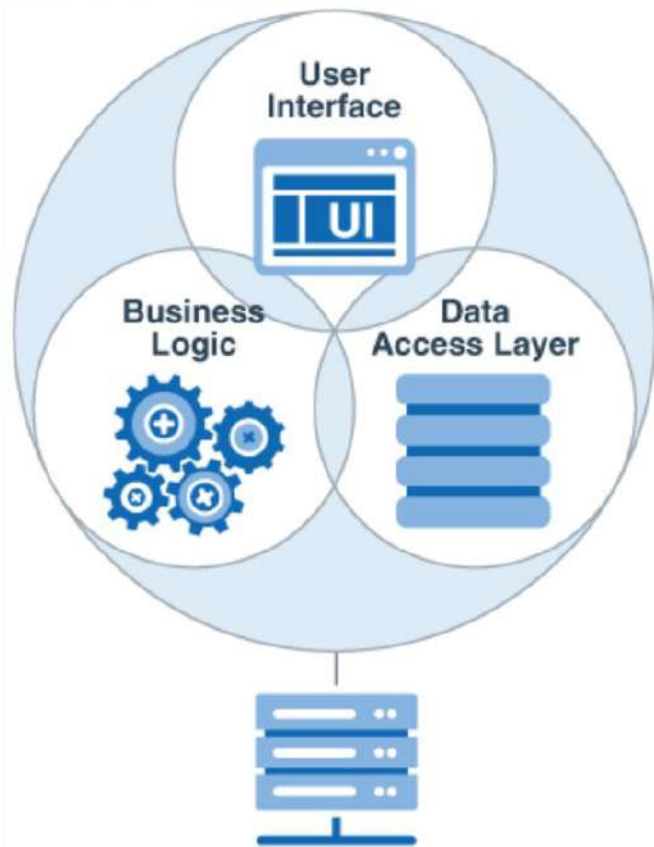


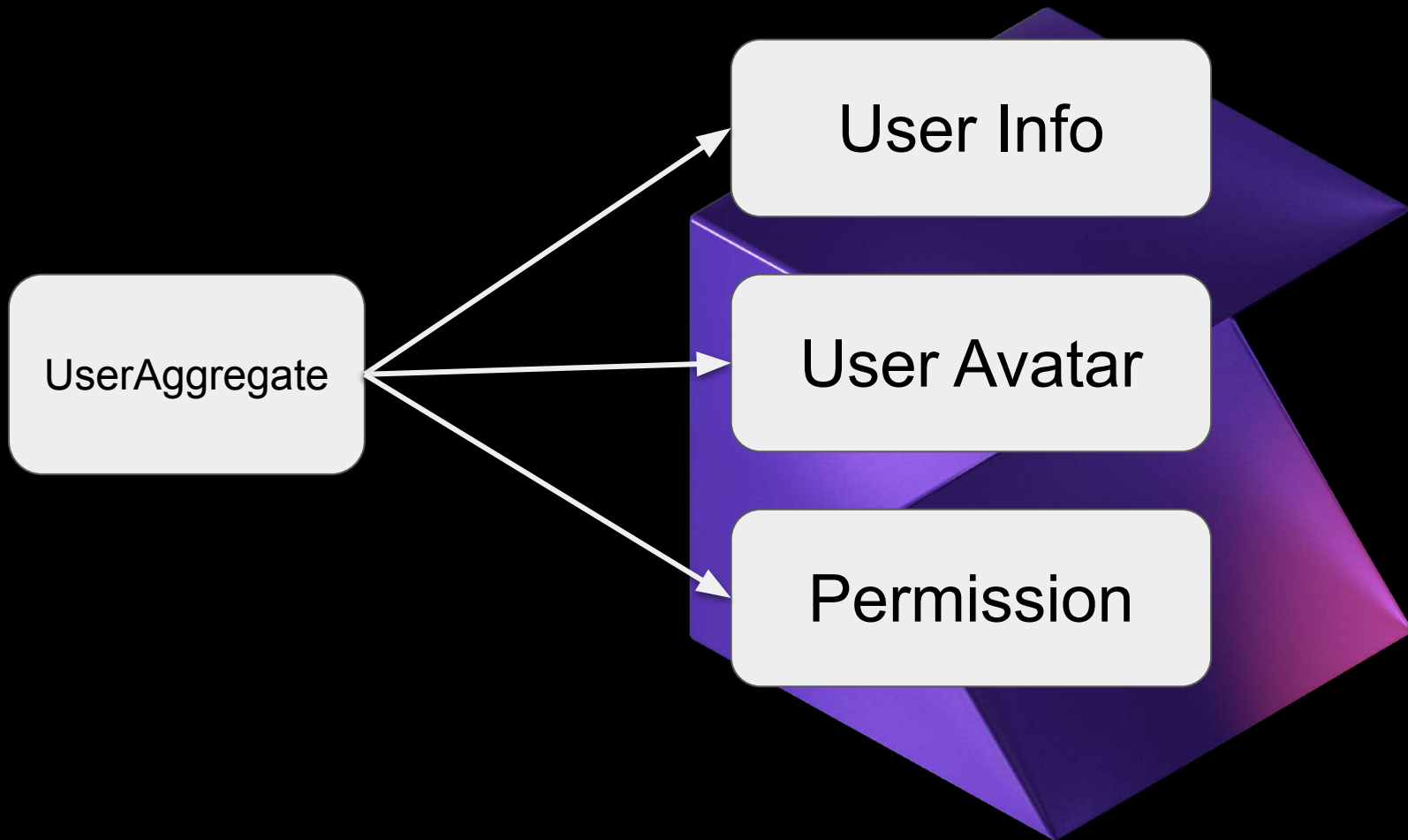
微服務有很多 Service 要打

Microservice Architecture



Monolithic Architecture






```
@Service
public class UserService {
    @Autowired
    private RestTemplate restTemplate;

    public UserAggregate getUserData(String userId) {

        UserInfo userInfo = restTemplate.getForObject(urlA + userId, UserInfo.class);
        Avatar avatar = restTemplate.getForObject(urlB + userId, Avatar.class);
        Permission permission = restTemplate.getForObject(urlC + userId,
Permission.class);

        UserAggregate userAggregate = new UserAggregate(userInfo, avatar, permission);

        // 將 userAggregate 物件緩存到 Redis
        cacheUserData(userId, userAggregate);

        return userAggregate;
    }
}
```

同時打出去

```
@Service
public class UserService {
    @Autowired
    private AsyncRestTemplate asyncRestTemplate;

    public CompletableFuture<UserAggregate> getUserData(String userId) {

        CompletableFuture<UserInfo> userInfoFuture = CompletableFuture.supplyAsync(() ->
            asyncRestTemplate.getForObject(urlA + userId, UserInfo.class));
        CompletableFuture<Avatar> avatarFuture = CompletableFuture.supplyAsync(() ->
            asyncRestTemplate.getForObject(urlB + userId, UserAvatar.class));
        CompletableFuture<Permission> permissionFuture = CompletableFuture.supplyAsync(() ->
            asyncRestTemplate.getForObject(urlC + userId, UserPermission.class));

        return CompletableFuture.allOf(userInfoFuture, avatarFuture, permissionFuture)
            .thenApply(v -> new UserAggregate(userInfoFuture.join(), avatarFuture
                .join(), permissionFuture.join()));
    }
}
```

試試 Kotlin Async, Await

@Service

```
class UserService(private val restTemplate: RestTemplate) {  
  
    suspend fun getUserData(userId: String): UserAggregate {  
        val userInfo = async { getUserInfo(userId) }  
        val avatar = async { getUserAvatar(userId) }  
        val permission = async { getUserPermission(userId) }  
  
        return UserAggregate(  
            userInfo.await(),  
            avatar.await(),  
            permission.await()  
        )  
    }  
  
    ...  
}
```

Arrow KT - parZip

```
import arrow.fx.coroutines.parZip

@Service
class UserService(private val restTemplate: RestTemplate) {

    suspend fun getUserData(userId: String): UserAggregate =
        parZip(
            { getUserInfo(userId) },
            { getAvatar(userId) },
            { getPermission(userId) },
        ){ userInfo, avatar, permission ->
            UserAggregate(userInfo, avatar, permission)
        }

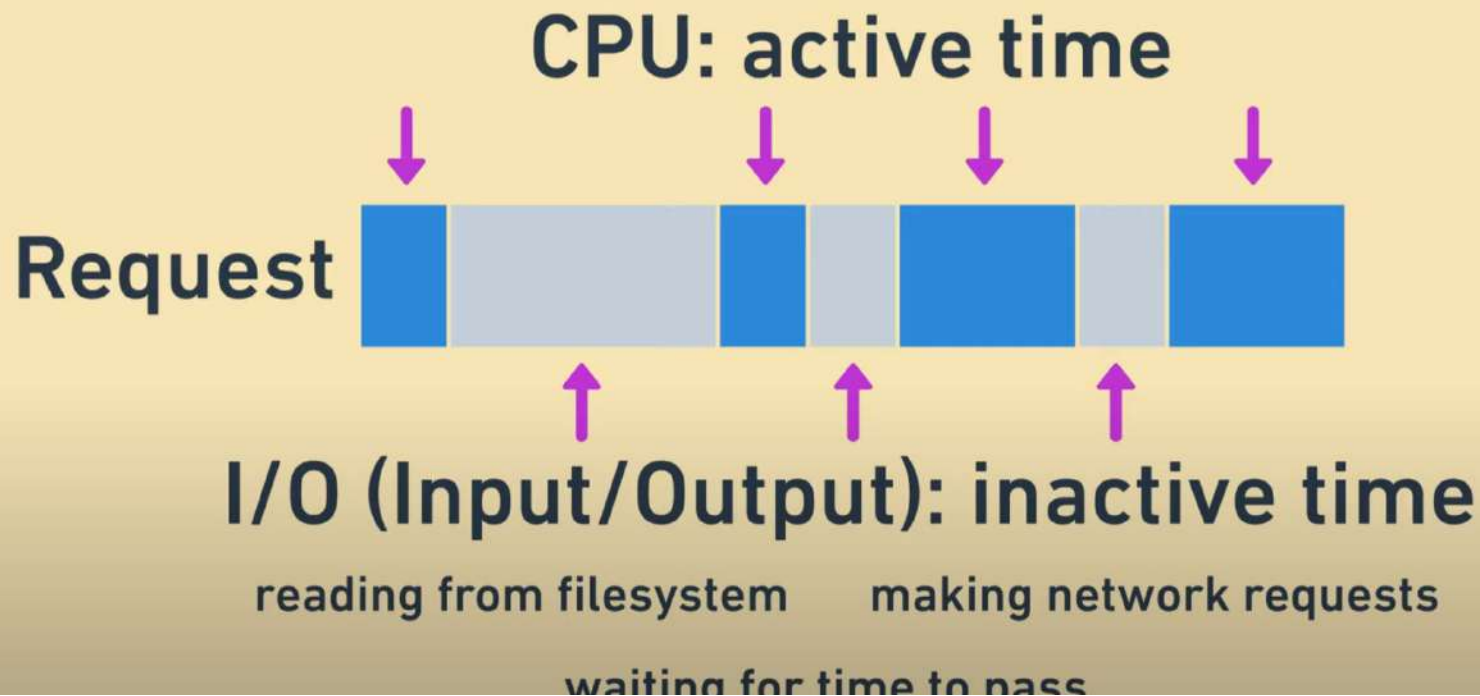
    ...
}
```

Arrow Resilience

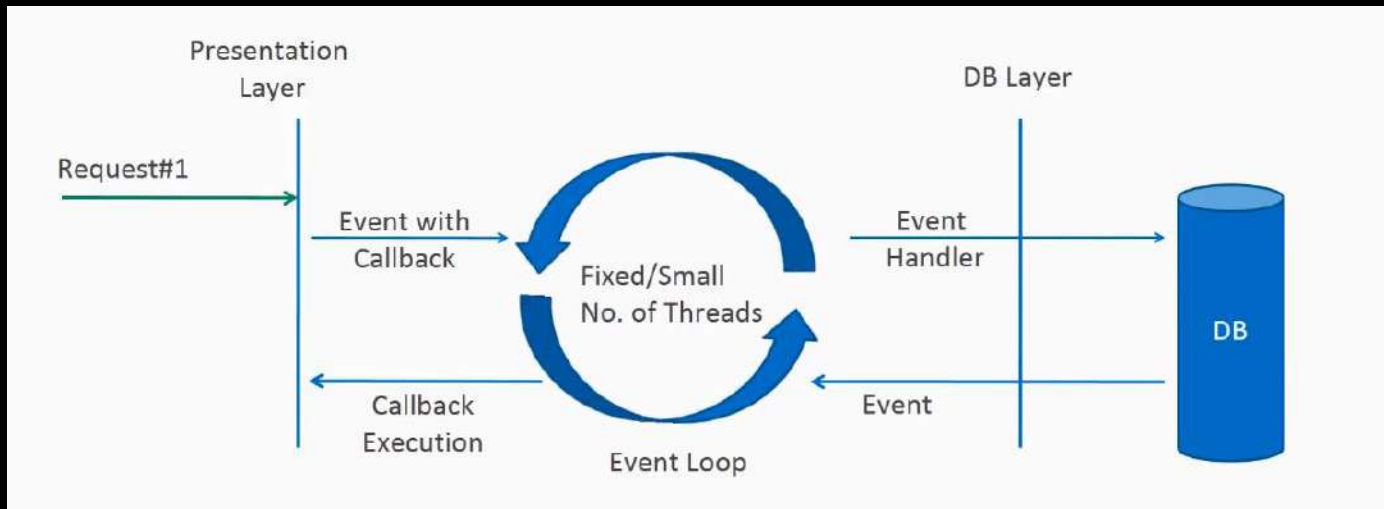
- Repeat
- Retry

```
1 suspend fun download(url: Url): ByteArray =  
2     Schedule.recurs(5).retry {  
3         url.readBytes()  
4     }
```

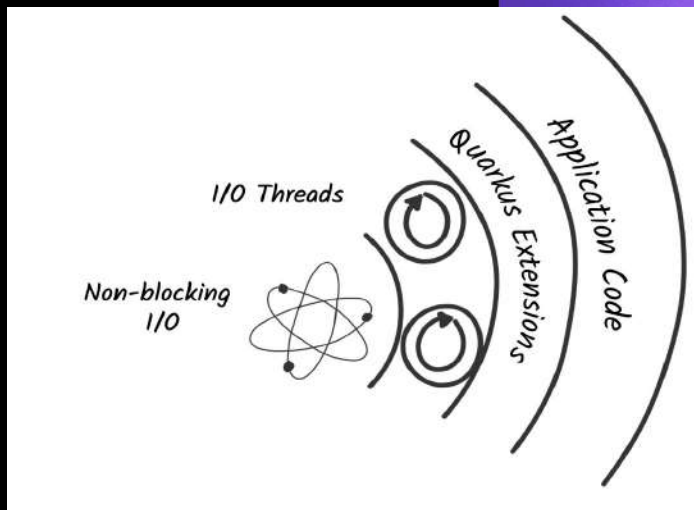
RestTemplate, JDBC are Blocking I/O




Spring Boot Webflux



Quarkus




Spring Boot Webflux

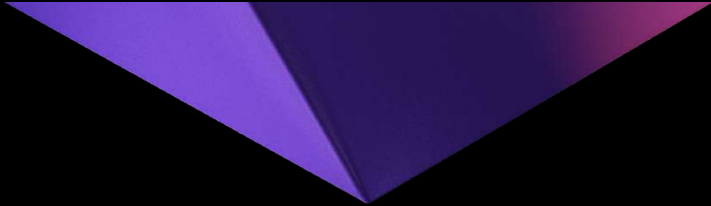


```
@GetMapping("/{name}")  
public Mono<Fruit> getOne(@PathVariable("name") String name) {  
    return Mono.just(new Fruit(name));  
}
```

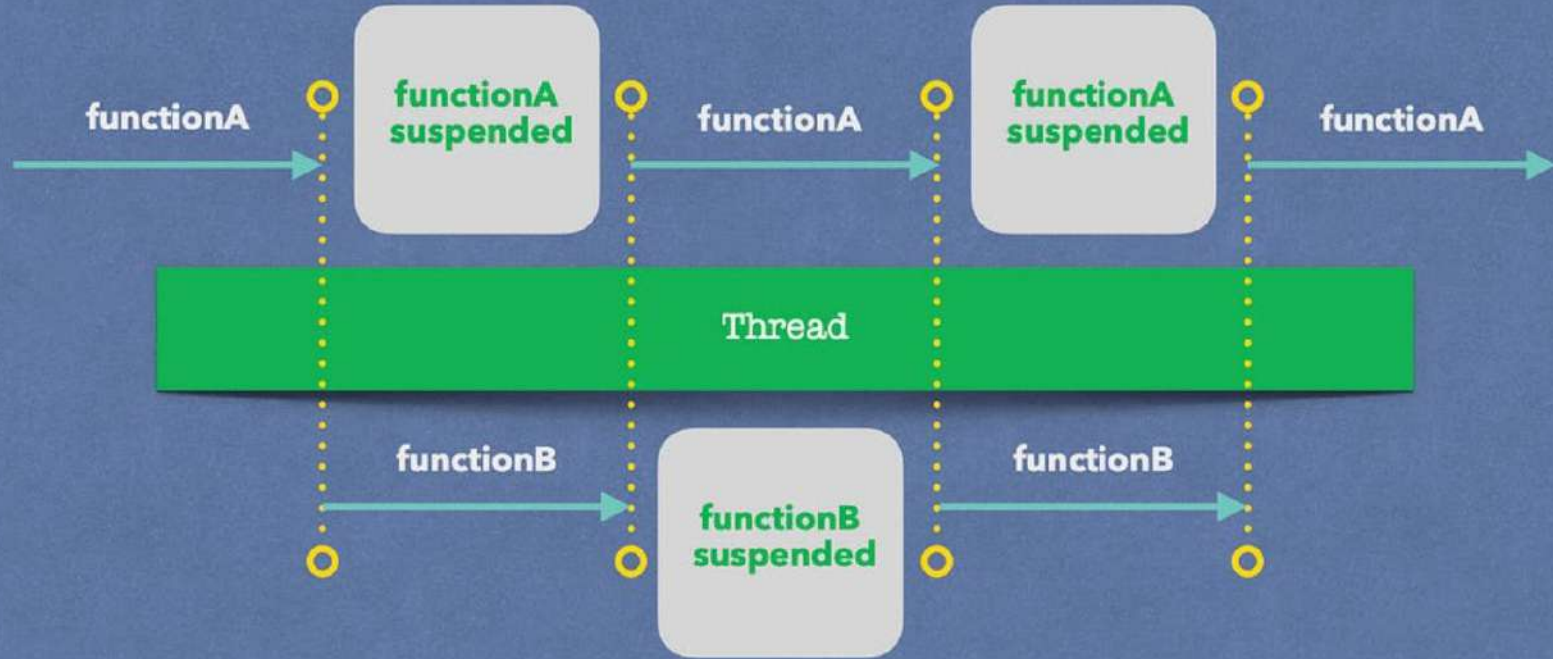
Quarkus




```
@GET  
@Path("/{name}")  
public Uni<Fruit> getOne(String name) {  
    return Uni.createFrom().item(new Fruit(name));  
}
```



Suspend function → Non-blocking I/O



Spring Boot Webflux

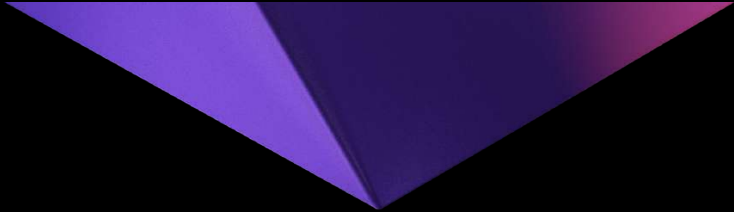


```
@GetMapping("/{name}")
suspend fun getOne(@PathVariable("name") name: String): Fruit {
    return Fruit(name)
}
```


Quarkus



```
@GET
@Path("/{name}")
suspend fun getOne(name: String): Fruit = Fruit(name)
```

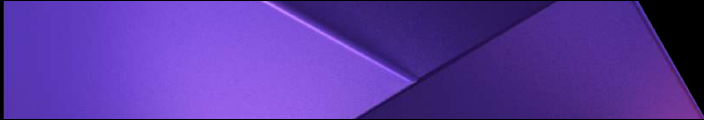


Spring Boot Webflux

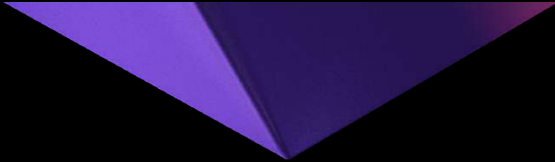


```
@GetMapping("/{name}")
suspend fun getOne(@PathVariable("name") name: String): Fruit {
    return fruitRepository.findByName(name).awaitFirstOrNull()
}
```

Quarkus



```
@GET
@Path("/{name}")
suspend fun getOne(name: String): Fruit =
    fruitRepository.findByName(name).awaitSuspending()
```



Benefits of Reactive

Non-Blocking IO



*Use resources
efficiently*



Event-driven



On event X Do Y



AWS SDK for Kotlin



What is the AWS SDK for Kotlin?

- A new SDK written from scratch
 - **Not** based on the AWS SDK for Java
- Written to be **Kotlin idiomatic**
 - Kotlin syntax and conventions (e.g., DSL syntax)
 - Kotlin types and null-safety
 - Ready for multiplatform (JVM/Android support to come)
 - Support for coroutines

MongoDB Documentation

[← Back To Docs Home](#)

Realm

[Introduction](#)[▶ C++ SDK Preview](#)[▶ Flutter SDK](#)[▶ Java SDK](#)[▼ Kotlin SDK](#)[Install Realm](#)[Quick Start](#)[Docs Home](#) → [Realm](#)

Migrate from the Java SDK to the Kotlin SDK

NOTE**What is the Kotlin SDK?**

The Kotlin SDK is a new Realm client SDK built entirely with the Kotlin programming language. The Kotlin SDK uses an entirely different codebase from the Java SDK. It is designed specifically to take advantage of Kotlin language features such as coroutines and suspend functions. The Java SDK also supports some of these features, as well as Android applications written in Kotlin. But the [Kotlin SDK](#) is more Kotlin-idiomatic than the Java SDK.

DDD with Kotlin

```
fun User.toDTO():UserDTO =
    UserDTO(id = id, userId = userId,
            name = name, phone = phone, email = email)

fun UserDTO.toUser():User {
    requireNotNull(userId)
    requireNotNull(name)

    val user = User()
    user.id = id
    user.userId = userId
    user.name = name
    user.phone = phone
    user.email = email
    return user
}
```

VO

DTO

DO

PO

新的一年
用 Kotlin 幫 Java 專案

除舊佈新



Thanks!

Have a nice Kotlin



@kotlin | Developed by JetBrains