
如何第一次用 Ktor Client 就撞牆

Jimin Hsieh

Agenda

- ❖ 如何第一次用 Ktor 就撞牆
- ❖ 又破牆
- ❖ Ktor Client Post
- ❖ Kotlin Coroutines

Ktor

- ❖ What is Ktor
 - ❖ A framework for asynchronous server and clients
 - ❖ Supported by JetBrains

What I want to do is an HTTP post

- ❖ Official sample code

- ❖ <https://ktor.io/clients/http-client/quick-start/requests.html#specifying-a-body-for-requests>

```
data class HelloWorld(val hello: String)
```

```
val client = HttpClient(Apache) {  
    install(JsonFeature) {  
        serializer = GsonSerializer {  
            // Configurable .GsonBuilder  
            serializeNulls()  
            disableHtmlEscaping()  
        }  
    }  
}
```

```
client.post<Unit> {  
    url("http://127.0.0.1:8080/")  
    body = HelloWorld(hello = "world")  
}
```


老夫寫代碼模式

不要給我說什麼
底层原理、框架内核！

老夫敲代碼就是

一把 **梭**！

复制！

粘貼！

拿起鍵盤就是

干！



V1

```
fun post(): HttpStatusCode {  
    val client = HttpClient(Apache) {  
        install(JsonFeature) {  
            serializer = GsonSerializer {  
                serializeNulls()  
                disableHtmlEscaping()  
            }  
        }  
    }  
}
```

```
    val response = client.post<HttpResponse>() {  
        url("http://127.0.0.1:8787/")  
        body = person  
    }  
    return response.status  
}
```

Compiling Error

- ❖ Error message
 - ❖ Suspend function 'post' should be called only from a coroutine or another suspend function

What is suspend function?

- ❖ A function can be paused and resumed at a later time.

V2, Same Error but at main

```
class HttpPostV2(val person: Person) {  
  
    suspend fun post(): HttpStatusCode {  
        val client = HttpClient(Apache) {  
            install(JsonFeature) {  
                serializer = GsonSerializer {  
                    serializeNulls()  
                    disableHtmlEscaping()  
                }  
            }  
        }  
  
        val response = client.post<HttpResponse>() {  
            url("http://127.0.0.1:8787/")  
            body = person  
        }  
        return response.status  
    }  
}  
  
fun main() {  
    val post = HttpPostV2(Person("Jimin", 181))  
    post.post()  
}
```

The meaning of error message?

- ❖ Suspend function 'post' should be called only from **a coroutine** or another suspend function

Kotlin coroutines

- ❖ What is Kotlin coroutines?
 - ❖ Green threads or light-weight threads

How to start a coroutine?

- ❖ CoroutineScope.launch
- ❖ CoroutineScope.async

How to start a coroutine?

- ❖ CoroutineScope.launch
 - ❖ Return Job type
- ❖ CoroutineScope.async
 - ❖ Return Deferred type

How to start a coroutine?

- ❖ CoroutineScope.launch
 - ❖ Return Job type
 - ❖ Job doesn't have result
- ❖ CoroutineScope.async
 - ❖ Return Deferred type
 - ❖ Deferred type is the subclass of Job, but it has a result. It's like Java Future.

V3

```
class HttpPostV3(val person: Person) {

    fun post(): HttpStatusCode {
        val client = HttpClient(Apache) {
            install(JsonFeature) {
                serializer = GsonSerializer {
                    serializeNulls()
                    disableHtmlEscaping()
                }
            }
        }

        val response = GlobalScope.async {
            client.post<HttpResponse>() {
                url("http://127.0.0.1:8787/")
                body = person
            }
        }

        return runBlocking {
            response.await().status
        }
    }
}

fun main() {
    val post = HttpPostV3(Person("Jimin", 181))
    post.post()
}
```

It works...but...

- ❖ Exception in thread "main" java.lang.IllegalStateException: Fail to send body. Content has type: class tw.jug.lite.model.Person (Kotlin reflection is not available), but OutgoingContent expected.

V4

```
fun post(): HttpStatusCode {  
    val client = HttpClient(Apache) {  
        install(JsonFeature) {  
            serializer = GsonSerializer {  
                serializeNulls()  
                disableHtmlEscaping()  
            }  
        }  
    }  
}
```

```
    val response = GlobalScope.async {  
        client.post<HttpResponse>() {  
            contentType(ContentType.Application.Json)  
            url("http://127.0.0.1:8787/")  
            body = person  
        }  
    }  
}
```

```
    return runBlocking {  
        response.await().status  
    }  
}
```

V5

```
fun post(): HttpStatusCode {  
    val client = HttpClient(Apache) {  
        install(JsonFeature) {  
            serializer = GsonSerializer {  
                serializeNulls()  
                disableHtmlEscaping()  
            }  
        }  
    }  
  
    return runBlocking {  
        val response = async {  
            client.post<HttpResponse>() {  
                contentType(ContentType.Application.Json)  
                url("http://127.0.0.1:8787/")  
                body = person  
            }  
        }  
        response.await().status  
    }  
}
```

runBlocking

- ❖ What is runBlocking?
 - ❖ Run a new coroutine and **wait** for the completion
 - ❖ Used in main function and in tests

V6

```
class HttpPostV6(val person: Person) {  
  
    suspend fun post(): HttpStatusCode {  
        val client = HttpClient(Apache) {  
            install(JsonFeature) {  
                serializer = GsonSerializer {  
                    serializeNulls()  
                    disableHtmlEscaping()  
                }  
            }  
        }  
  
        return client.post<HttpResponse>() {  
            contentType(ContentType.Application.Json)  
            url("http://127.0.0.1:8787/")  
            body = person  
        }.status  
    }  
}  
  
fun main() = runBlocking {  
    val post = HttpPostV6(Person("Jimin", 181))  
    println(post.post())  
}
```

What's the issue of the beginning?

- ❖ Error Message
 - ❖ Suspend function 'post' should be called only from a coroutine or another suspend function

What's the issue of the beginning?

- ❖ Error Message
 - ❖ Suspend function 'post' should be called only from a coroutine or another suspend function
- ❖ Didn't run a coroutine under the **scope**.

CoroutineScope

- ❖ What is CoroutineScope?
 - ❖ The scope of coroutine resource

CoroutineContext

- ❖ What is CoroutineContext
 - ❖ Coroutines always execute in some context represented by a value of the CoroutineContext type, defined in the Kotlin standard library.
 - ❖ The **runtime environment** of coroutines
 - ❖ A set of elements
 - ❖ Job
 - ❖ Dispatcher
 - ❖ ...etc

Things I don't consider for this sample

- ❖ Test
- ❖ Resource utilization