

ING



```
data class KotlinMeetup(val title: String, val talks: List<Talk>)
data class Talk(val title: String, var author: String, var startTime: LocalTime, var endTime: LocalTime)
```

TYPE-SAFE BUILDERS USING FUNCTIONAL INTERFACES WITH RECIPIENT



































































































































































































































































































































































```
fun talk(title: String, init: Talk.() → Unit): Talk {  
    val talk = Talk(title)  
    talk.init()  
    return talk  
}
```

```
fun talk(title: String, init: Talk.() → Unit) = Talk(title).apply(init)
```

```
public inline fun <T> T.apply(block: T.() → Unit): T {  
    contract {  
        callsInPlace(block, InvocationKind.EXACTLY_ONCE)  
    }  
    block()  
    return this  
}
```

```
data class KotlinMeetup(val title: String, val talks: List<Talk>)  
fun talk(title: String, init: Talk.() → Unit) = Talk(title).apply(init)
```

TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER

```
data class KotlinMeetup(val title: String, val talks: List<Talk>)  
fun talk(title: String, init: Talk.() → Unit): Talk
```

TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER