

Using Data Class as Navigation Arguments

Data class in Navigation Arguments

```
@Serializable
sealed class Route {
    @Serializable
    data class SelectPayment(val totalCharge: Float)

    @Serializable
    data class SelectPaymentAsObject(val m: Money)
}

@Serializable
data class Money(val amount: Float, val currency: String)
```

Declare custom NavType<Money>

```
val MoneyNavType = object: NavType<Money>(isNullableAllowed = false) {  
    override fun get(bundle: Bundle, key: String): Money? {  
        return Json.decodeFromString(bundle.getString(key) ?: return null)  
    }  
  
    override fun put(bundle: Bundle, key: String, value: Money) {  
        bundle.putString(key, Json.encodeToString(value))  
    }  
  
    override fun parseValue(value: String): Money {  
        return Json.decodeFromString(Uri.decode(value))  
    }  
  
    override fun serializeAsValue(value: Money): String {  
        return Uri.encode(Json.encodeToString(value))  
    }  
}
```

Add typeMap to composable<>

```
NavHost(.....) {  
    composable<Route.SelectPayment> {  
        CheckoutScreen(totalCharge = 50f, vm = orderViewModel)  
    }  
    composable<Route.SelectPaymentAsObject> (  
        typeMap = mapOf(typeOf<Money>() to MoneyNavType)  
    ) {  
        val payload = it.toRoute<SelectPaymentAsObject>()  
        CheckoutScreen(totalCharge = payload.amount, vm = orderViewModel)  
    }  
}
```