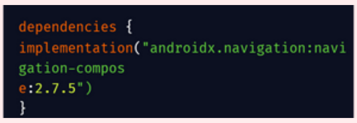
JETPACK COMPOSE NAVIGATION CHEATSHEET

Navigation Setup

Dependency:



```
dependencies {
implementation("androidx.navigation:navi
gation-compos
e:2.7.5")
}
```

Key Components:

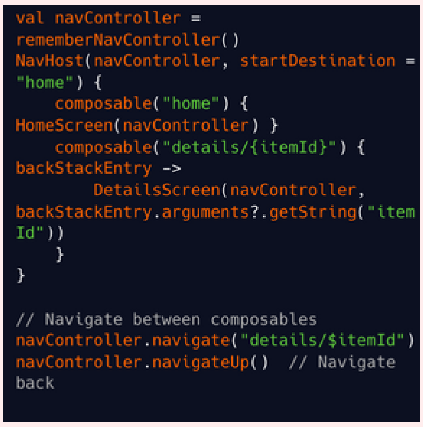
NavHost:
Displays composables based on navigation state.

NavController:
Manages navigation within a NavHost.

NavGraph:
Defines navigation structure.

NavBackStackEntry:
Represents an entry in the back stack.

Basic Example:

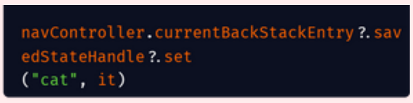


```
val navController =
rememberNavController()
NavHost(navController, startDestination =
"home") {
    composable("home") {
HomeScreen(navController) }
    composable("details/{itemId}") {
backStackEntry ->
        DetailsScreen(navController,
backStackEntry.arguments?.getString("item
Id"))
    }
}

// Navigate between composables
navController.navigate("details/$itemId")
navController.navigateUp()  // Navigate
back
```

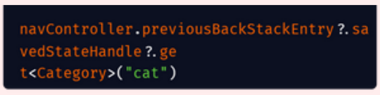
Saving and Retrieving Data

Save Data:



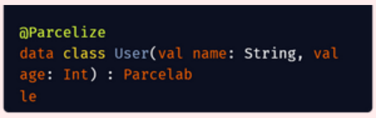
```
navController.currentBackStackEntry?.sav
edStateHandle?.set
("cat", it)
```

Retrieve Data:



```
navController.previousBackStackEntry?.sa
vedStateHandle?.ge
t<Category>("cat")
```

Parcelable Example:



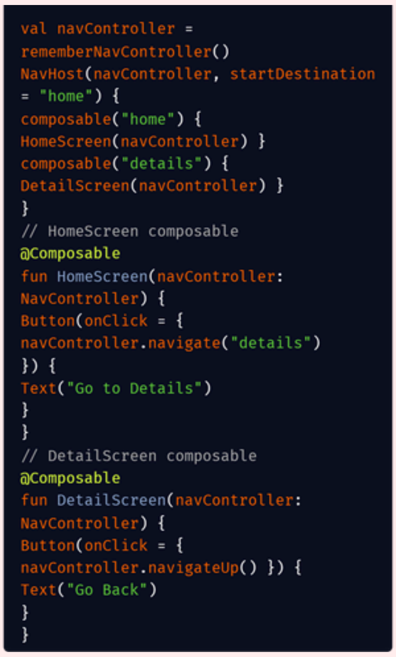
```
@Parcelize
data class User(val name: String, val
age: Int) : Parcelab
le
```

@Parcelize: Auto-generates Parcelable implementation, optimized for Android.

NavController Details

Concept: Manages navigation, knows routes, handles back stack, transitions between destinations.

Usage:

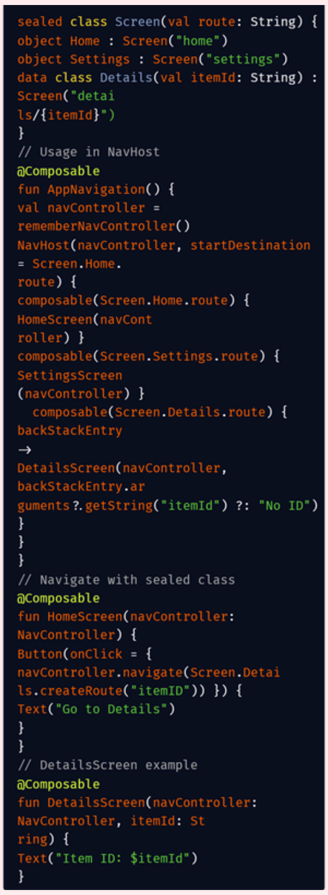


```
val navController =
rememberNavController()
NavHost(navController, startDestination
= "home") {
composable("home") {
HomeScreen(navController) }
composable("details") {
DetailScreen(navController) }
}
// HomeScreen composable
@Composable
fun HomeScreen(navController:
NavController) {
Button(onClick = {
navController.navigate("details")
}) {
Text("Go to Details")
}
}
// DetailScreen composable
@Composable
fun DetailScreen(navController:
NavController) {
Button(onClick = {
navController.navigateUp() }) {
Text("Go Back")
}
}
```

Sealed Classes

Definition: Restricts class inheritance; useful for handling known types in when expressions.

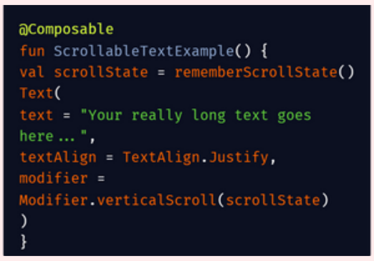
Example:



```
sealed class Screen(val route: String) {
object Home : Screen("home")
object Settings : Screen("settings")
data class Details(val itemId: String) :
Screen("detai
ls/{itemId}")
}
// Usage in NavHost
@Composable
fun AppNavigation() {
val navController =
rememberNavController()
NavHost(navController, startDestination
= Screen.Home.
route) {
composable(Screen.Home.route) {
HomeScreen(navCont
roller) }
composable(Screen.Settings.route) {
SettingsScreen
(navController) }
  composable(Screen.Details.route) {
backStackEntry
→
DetailsScreen(navController,
backStackEntry.ar
guments?.getString("itemId") ?: "No ID")
}
}
}
// Navigate with sealed class
@Composable
fun HomeScreen(navController:
NavController) {
Button(onClick = {
navController.navigate(Screen.Detai
ls.createRoute("itemID")) }) {
Text("Go to Details")
}
}
// DetailsScreen example
@Composable
fun DetailsScreen(navController:
NavController, itemId: St
ring) {
Text("Item ID: $itemId")
}
```

Making Text Scrollable

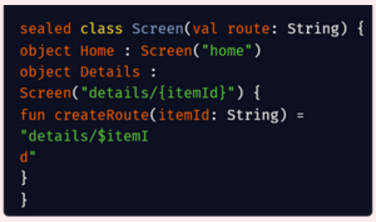
Scrollable Text Example:



```
@Composable
fun ScrollableTextExample() {
val scrollState = rememberScrollState()
Text(
text = "Your really long text goes
here ... ",
textAlign = TextAlign.Justify,
modifier =
Modifier.verticalScroll(scrollState)
)
}
```

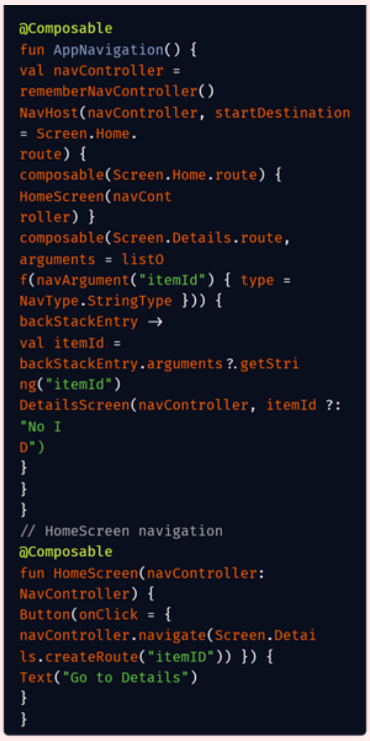
Passing Variables Between Screens

Define Route with Argument:



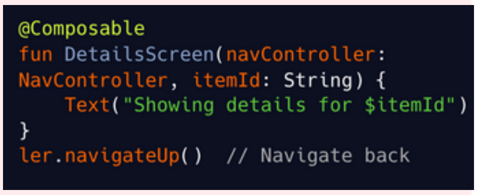
```
sealed class Screen(val route: String) {
object Home : Screen("home")
object Details :
Screen("details/{itemId}") {
fun createRoute(itemId: String) =
"details/$itemI
d"
}
}
```

Setup Navigation:



```
@Composable
fun AppNavigation() {
val navController =
rememberNavController()
NavHost(navController, startDestination
= Screen.Home.
route) {
composable(Screen.Home.route) {
HomeScreen(navCont
roller) }
composable(Screen.Details.route,
arguments = listO
f(navArgument("itemId") { type =
NavType.StringType })) {
backStackEntry →
val itemId =
backStackEntry.arguments?.getStri
ng("itemId")
DetailsScreen(navController, itemId ?:
"No I
D")
}
}
}
// HomeScreen navigation
@Composable
fun HomeScreen(navController:
NavController) {
Button(onClick = {
navController.navigate(Screen.Detai
ls.createRoute("itemID")) }) {
Text("Go to Details")
}
}
```

Receive Variable in Destination:



```
@Composable
fun DetailsScreen(navController:
NavController, itemId: String) {
    Text("Showing details for $itemId")
}
ler.navigateUp()  // Navigate back
```