Lab 12 - Sending Feedback

`COMP4107 - SDDT - HKBU - Spring2023`

Today we'll continue building our InfoDay app using Jetpack Compose. We'll discuss how to **send post request with `Ktor`**.

Clone your project from GitHub https://classroom.github.com/a/t86LHCCf and let's keep developing.

Sending Feedback

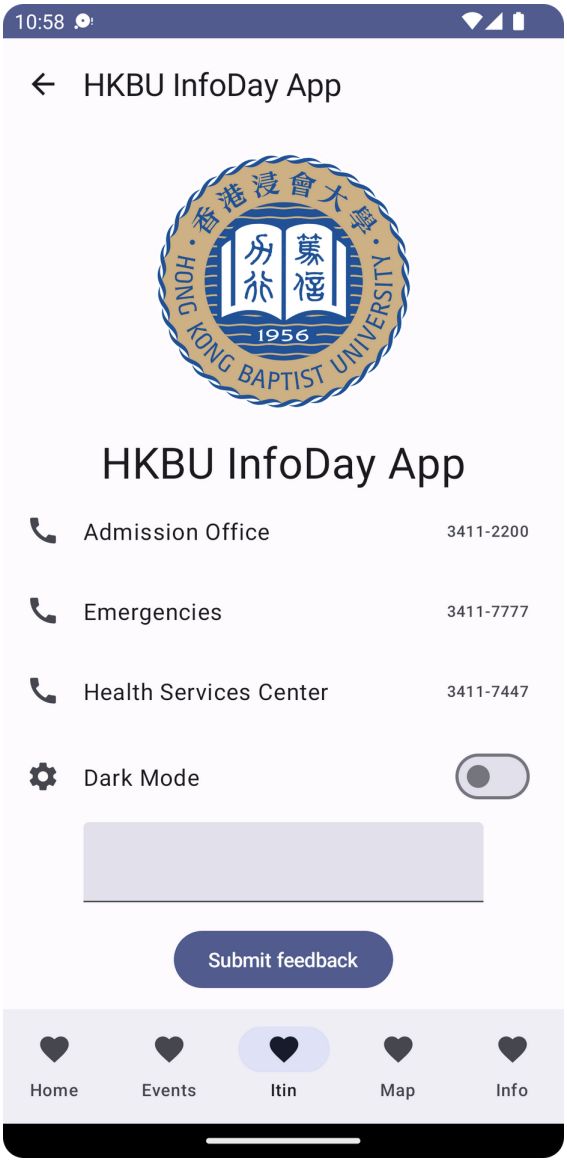
In the `InfoScreen`, develop a couple of new elements:

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun FeedBack(snackbarHostState: SnackbarHostState) {
    val padding = 16.dp
    var message by remember { mutableStateOf("") }
    val coroutineScope = rememberCoroutineScope()

    Column(horizontalAlignment = Alignment.CenterHorizontally) {
        TextField(
            maxLines = 1,
            value = message,
            onValueChange = { message = it }
        )
        Spacer(Modifier.size(padding))

        Button(onClick = {
            coroutineScope.launch {
                val stringBody: String = KtorClient.postFeedback(message)
                snackbarHostState.showSnackbar(stringBody)
            }
        }) {
            Text(text = "Submit feedback")
        }
    }
}
```

Here, we have established `message` as a state, with its value **immune to change during recomposition**. This `message` is linked to the `TextField`. Also, we have a **button that will trigger a network request** when clicked.



In the `KtorClient` class, we defined a new `suspend` function for POST requests.

```
suspend fun postFeedback(feedback: String): String {
    return httpClient.post("https://httpbin.org/post") {
        setBody(feedback)
    }.body()
}
```

Now, you can enter content into the text box and click the button. A notification (snack bar) will appear showing the echoed information.

Process the Received Data

The `httpbin.org` echo server will return exactly what you send it. An example is shown below:

```
{
  "args": {},
  "data": "",
  "files": {},
```

```
  "form": {
    "gridRadios": "option1"
  },
  "headers": {
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "Accept-Encoding": "gzip, deflate, br",
    "Accept-Language": "en-GB,en;q=0.9",
    "Content-Length": "18",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "www.httpbin.org",
    "Origin": "http://127.0.0.1:5500",
    "Referer": "http://127.0.0.1:5500/",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.5 Safari/605.1.15",
    "X-Amzn-Trace-Id": "Root=1-62f89dc4-525ffe5d37d8398257ff2c7c"
  },
  "json": null,
  "origin": "158.182.194.232",
  "url": "https://www.httpbin.org/post"
}
```


To access its **first-level information**, you can create a data class like this:

```
@Serializable
data class HttpBinResponse(
    val args: Map<String, String>,
    val data: String,
    val files: Map<String, String>,
    val form: Map<String, String>,
    val headers: Map<String, String>,
    val json: String?,
    val origin: String,
    val url: String
)
```

To display the `X-Amzn-Trace-Id` information, we can modify the `postFeedback` function as follows:

```
suspend fun postFeedback(feedback: String): String {

    val response: HttpBinResponse = httpClient.post("https://httpbin.org/post") {
        setBody(feedback)
    }.body()
```

```
    return response.headers["X-Amzn-Trace-Id"].toString()
}
```

Token-based Authentication

We could set up a `token` property in the `KtorClient` singleton to store the login token and include it in all subsequent requests. This would ensure the token is sent with each request,
allowing the server to identify the user and authorize the request.

```
private var token: String = ""
```

Then, the `defaultRequest` could possibly be modified as follows:

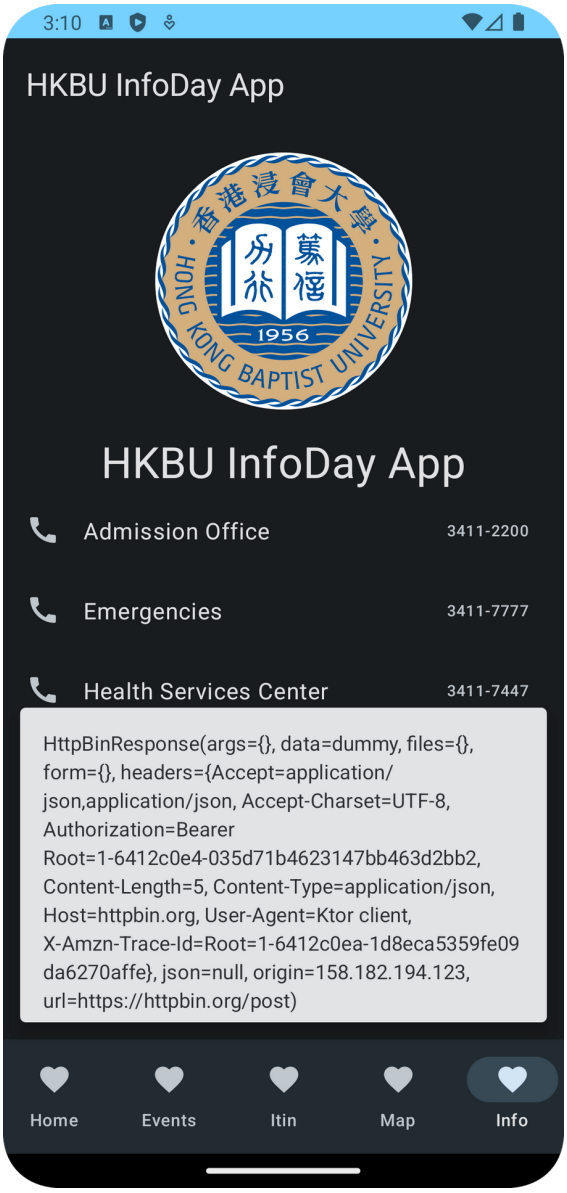
```
defaultRequest {
    contentType(ContentType.Application.Json)
    accept(ContentType.Application.Json)
    header("Authorization", "Bearer " + token)
}
```

Finally, we store the `X-Amzn-Trace-Id` to the `token` property:

```
suspend fun postFeedback(feedback: String): String {

    val response: HttpBinResponse = httpClient.post("https://httpbin.org/post") {
        setBody(feedback)
    }.body()

    token = response.headers["X-Amzn-Trace-Id"].toString()
    return response.toString()
}
```



The access token is now included in the request headers.

16/03/2023 14:52