

Mobile Applications

CSCI 448

Lecture 30



Paging v3

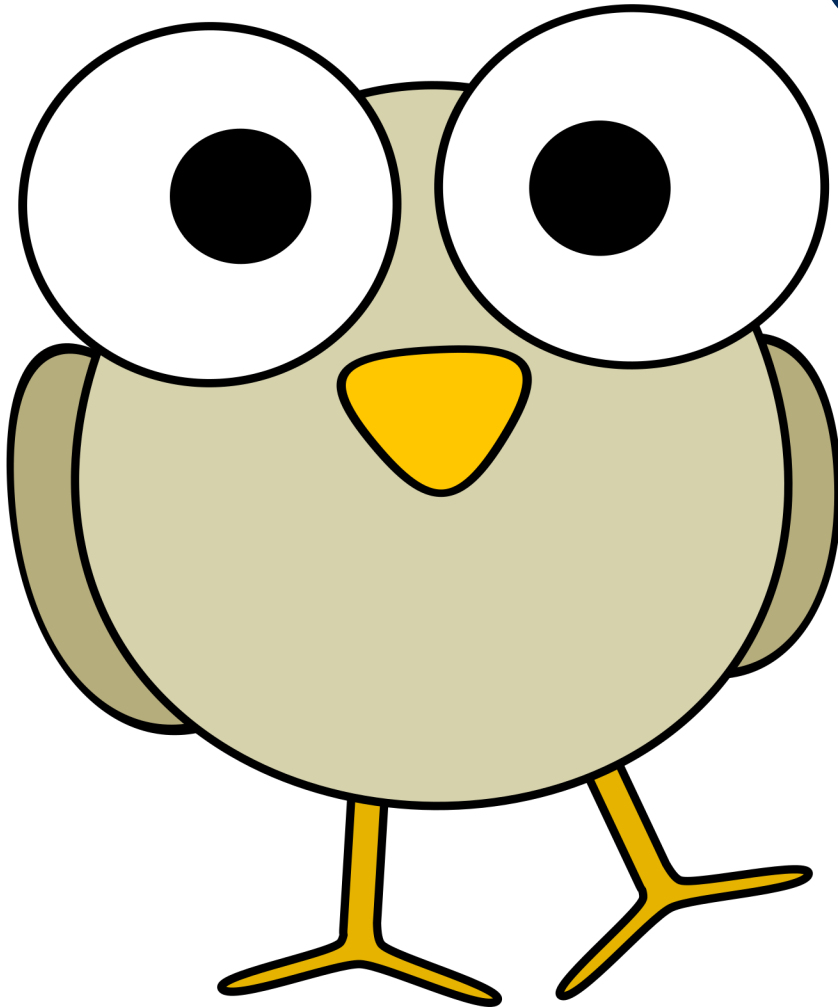


Previously in CSCI 448



- Firebase
 - User Authentication in 11 easy lines of code!
 - Remote Database as JSON Tree via
 - REST API
 - Android API

Questions?



??

Learning Outcomes For Today



- Discuss when paginated data should be used
- Explain how to work with paginated data

On Tap For Today



- Paging
- Practice!

When To Use Paging

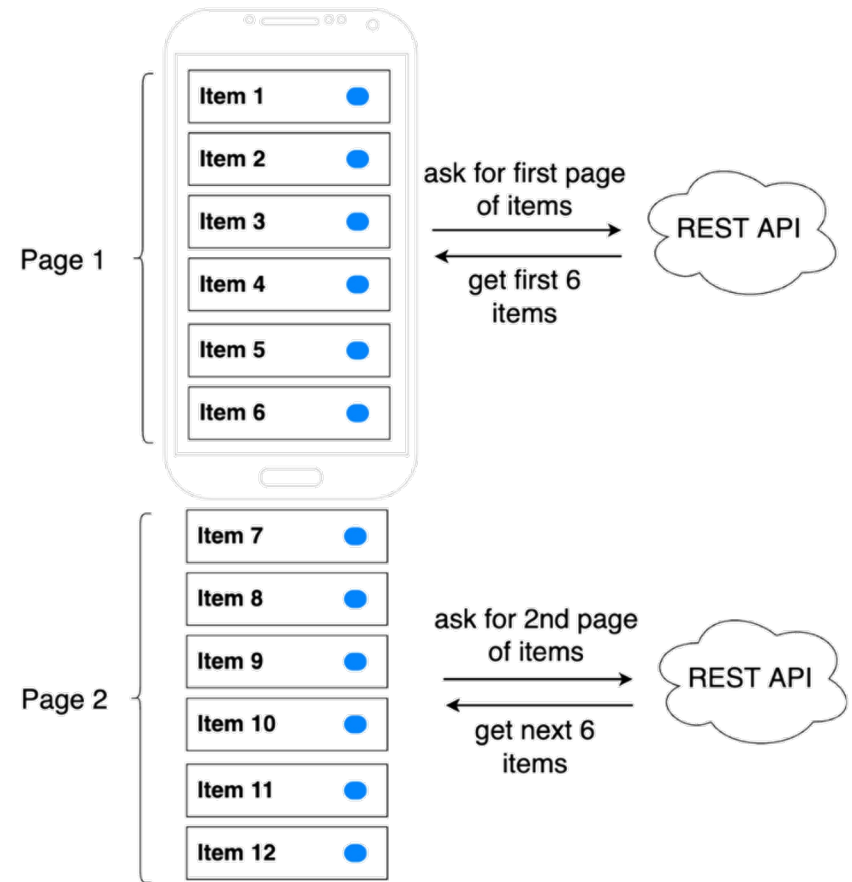


Figure 11.1 – Observing how infinite lists can be achieved with pagination

Paging v3



- Add dependency to build.gradle

```
implementation "androidx.paging:paging-runtime:3.1.1"
```

```
implementation "androidx.paging:paging-compose:1.0.0-alpha08"
```

Paging Architecture

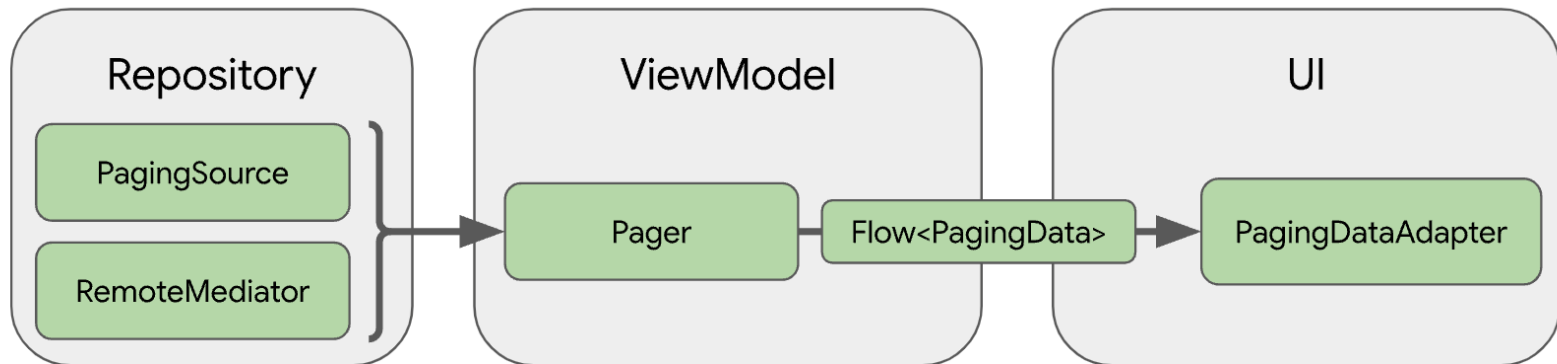


Figure 1. An example of how the Paging library fits into your app architecture.

On Tap For Today



- Paging
 - Model: PagingSource
 - ViewModel: Pager
 - View: PagingDataAdapter
- Practice!

On Tap For Today



- Paging
 - Model: PagingSource
 - ViewModel: Pager
 - View: PagingDataAdapter
- Practice!

PagingSource<Key, Value>



- Key – page number (**Int**)
- Value – instance of a page item
- **load()**
 - Perform lookup for page
 - Return result

LoadResult.Page



```
LoadResult.Page(  
    data = ,          // list of items  
    prevKey = ,       // index of prior page  
    nextKey =         // index of next page  
)
```

- If this is the first page, then **prevKey** is **null**
 - Otherwise, **prevKey** = current page number - 1
- If this is the last page and don't load any more, then **nextKey** is **null**
 - Otherwise, **nextKey** = current page number + 1 and therefore infinite scrolling

On Tap For Today



- Paging
 - Model: PagingSource
 - ViewModel: Pager
 - View: PagingDataAdapter
- Practice!

Pager



- Interfaces with PagingSource
 - Requests new pages
 - Caches already collected pages
 - Exposes current page from collected pages
- Can specify
 - Page size
 - Prefetch distance

Pager()



```
class MyViewModel : ViewModel() {  
    private val pager = Pager(  
        config = PagingConfig(),  
        pagingSourceFactory = {  
            PagingSource()  
        }  
    )  
    val pagerFlow = pager.flow.cachedIn(viewModelScope)  
}
```

On Tap For Today



- Paging
 - Model: PagingSource
 - ViewModel: Pager
 - View: PagingDataAdapter
- Practice!

Collect Flow



```
val lazyPagingItems = viewModel.pagerFlow.collectAsLazyPagingItems()
```

```
LazyColumn(  
    modifier = ...  
) {  
    items(lazyPagingItems) { item ->  
        ItemRow(item)  
    }  
}
```

- *collectAsLazyPagingItems()* and DSL *items()* know to call back up to pager when more pages needed

Good UI/UX



- Access LoadState
 - Display loading indicator
 - `CircularProgressIndicator()`
 - Display error message
- Supports refresh or append actions

Paged Flow In Action

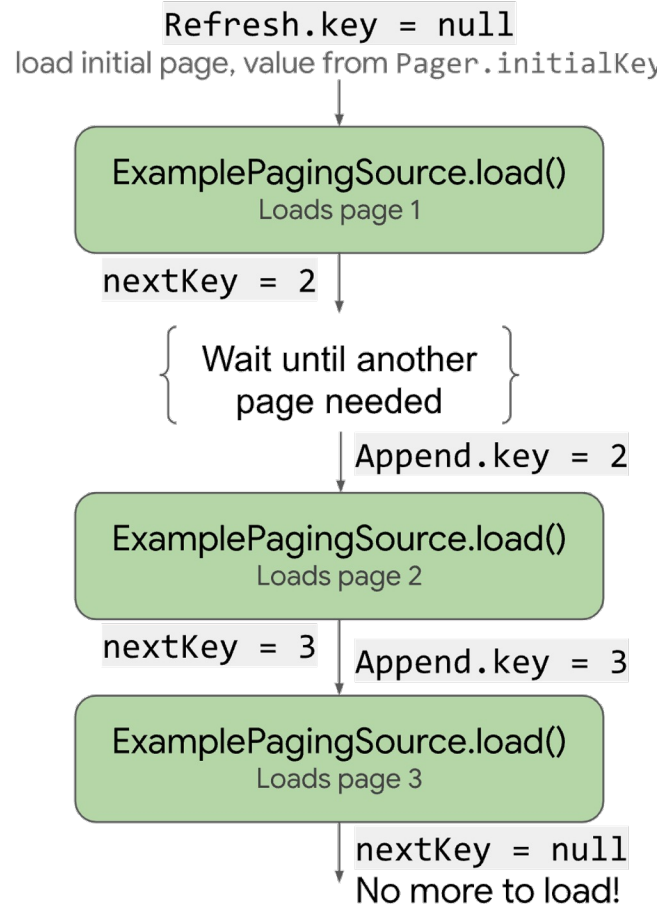


Figure 1. Diagram showing how `load()` uses and updates the key.

On Tap For Today



- Paging
 - Model: PagingSource
 - ViewModel: Pager
 - View: PagingDataAdapter
- Practice!

Next Time



- Exam 2 out today – due Friday
 - Think again like a quiz
- Exam 1 grades almost done, coming soon
 - Sorry 😞
- Don't forget to complete Sales Pitch Survey in Canvas