

Retrieving data from databases

⇒ Coroutines ⇒ Better supported by Kotlin

⇒ Annotations to better define syntax of SQL instead of directly giving SQL commands

⇒ Registering of actions to avoid mixing of code of different classes

Retrieving data over the network follows a similar concept.

Network uses two protocols ⇒ 1) HTTP (Trad. run over TCP) [packet reordering and retransmission; eventual delivery];
HTTP3 (built on top of QUIC) meaning reliability guarantees so no guarantees on latency.

2) Real-time streams of data ⇒ UDP
↓
RTP (Real-Time Protocol)
Does Youtube utilize this??

How does HTTP actually work?

⇒ Get, Post, Put, Delete, Head
↓
Send this command to get some data/resource
↓
Send this command along with some data/resource
Extra detail about ensuring data is written

↓
List of strings

↓
Send a list
of strings

Retrofit Library

```
interface Data Access API {
```

```
    @GET("/")
```

```
    suspend fun
```

```
    fetchContents(): String
```

```
}
```

Must-have; otherwise your app can get unresponsive

```
fun onViewCreated() {
```

```
    val retrofit: Retrofit = Retrofit.Builder()
```

```
        .baseUrl("iitd.ac.in")
```

```
        .addConverterFactory(ScalarsConverterFactory.create())
```

```
        .addConverterFactory(...)
```

```
        .build()
```

```
    val dataAccessAPI: Data Access API
```

```
        = retrofit.create<List<...>>()
```

```
    { dataAccessAPI.fetchContents() }
```

Array of Integers $\xrightarrow{\text{Serialization}}$ Writing the array members in the form of a string

Binary Tree $\xRightarrow{\text{Inorder Traversal} + \text{Preorder}}$ Convert both traversals to a String

Mapping on object \Rightarrow Moshi Library
 \Rightarrow Map objects from HTTP and JSON.

• add Converter Factory (Moshi Converter Factory) create(), build()
 \Rightarrow Create a class with the same variable names and map the data accordingly.

Map the variables defined in another class to the data coming in.

```
{  
  <var_name>: <val>  
  :  
  {  
    :  
    :  
  }  
}
```

-
- 1) Get request can send huge amount of data
 \Rightarrow 100 records at a time
Paging } \Rightarrow Sends the records in units of pages, by default 100.
 - 2) Failure to connect is handled via exception handling.

try {

} catch (ex: Exception) {
 Log(...)
}

{ id? — name? } - - -

⇓
{ Build up this string and
 Send it with the GET request.

③ { Accessing network requires permission
 ⇒ Non-sensitive ⇒ Mention it in the manifest file
 User is not explicitly asked about it.
 Possible to deny permission by the user,

Saturday at 4pm.