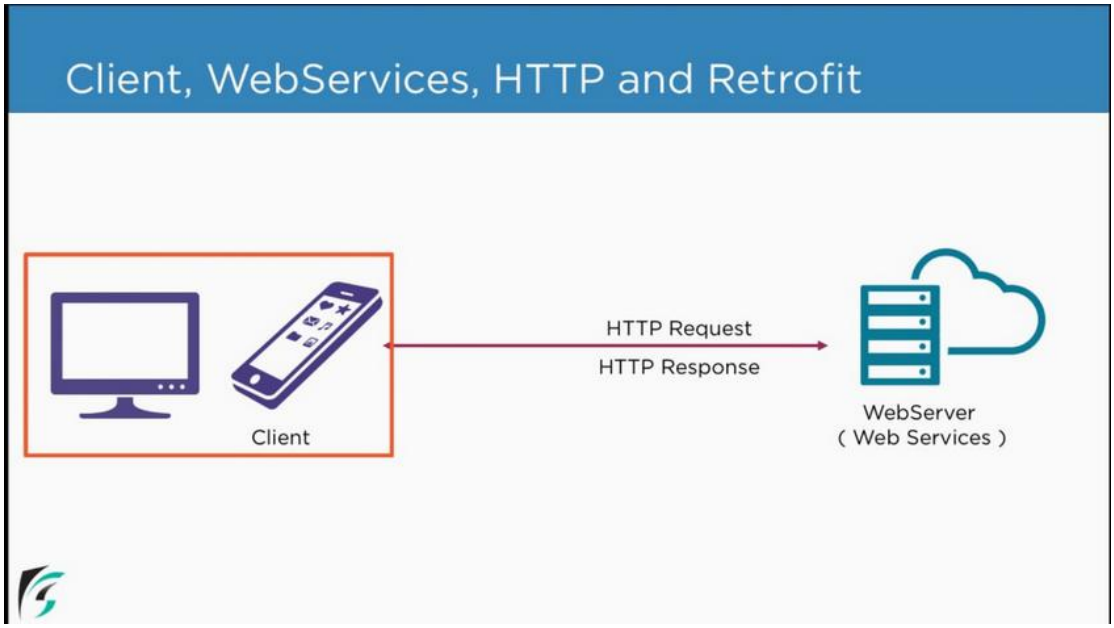
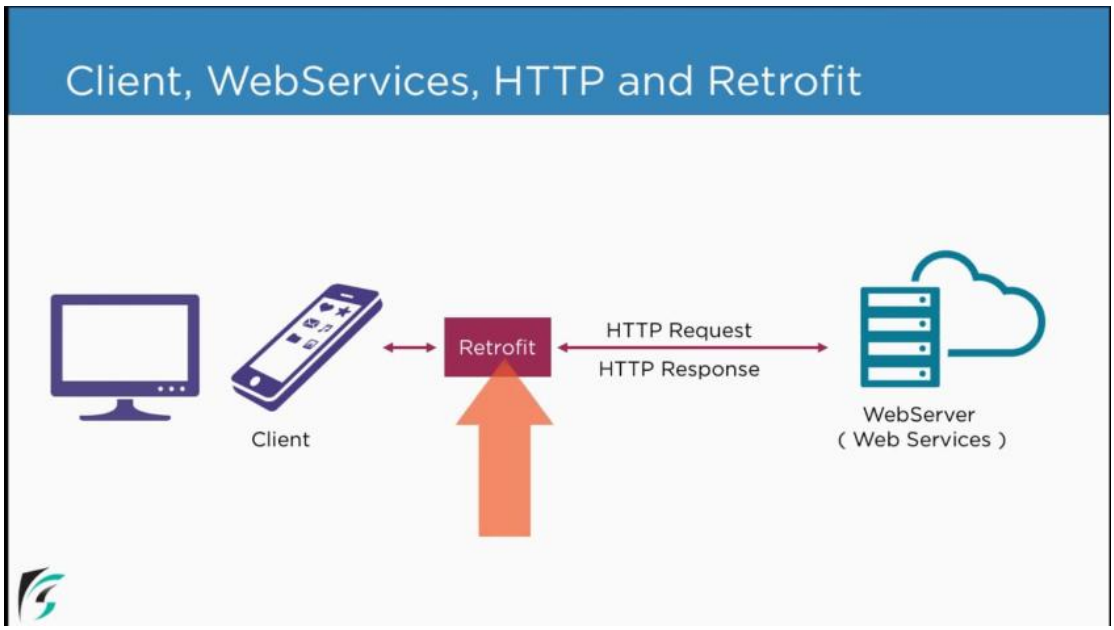
Retrofit



This is the basic communication pattern that will connect an application to the remote world. The client and the server communicate with each other via HTTP requests. The work of retrofit is to control and manage the HTTP connections as shown below:



What makes your app proffesional?

1. Handle Authentication seamlessly
2. Run request in background thread
3. Parse JSON to a usable class object
4. Handle images effectively

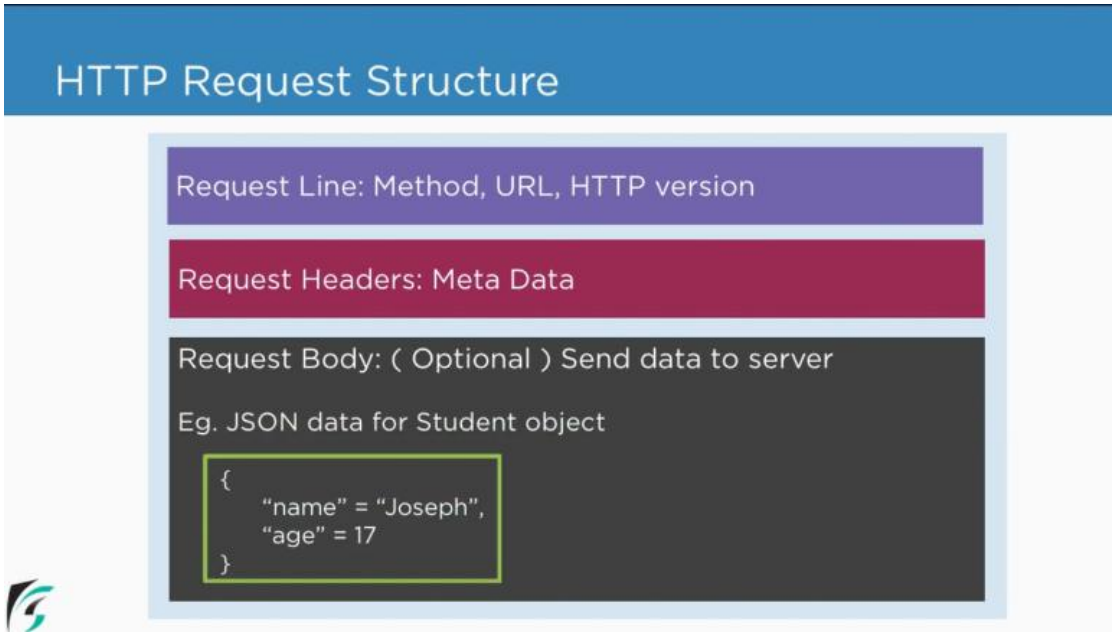
Webservers -> RESTful Web Services

1. HTTP fundamentals

This section simply describes the communication flow between the client and web server. From the client we send a **HTTP request** to the server. In between there are protocols that alter the request to a form understandable by the server. Once the server gets the request it responds to it by sending a **HTTP response.** In between the response is altered to a form understandable to the client.

HTTP request structure

The basic structure of a HTTP request is as shown by the picture below:



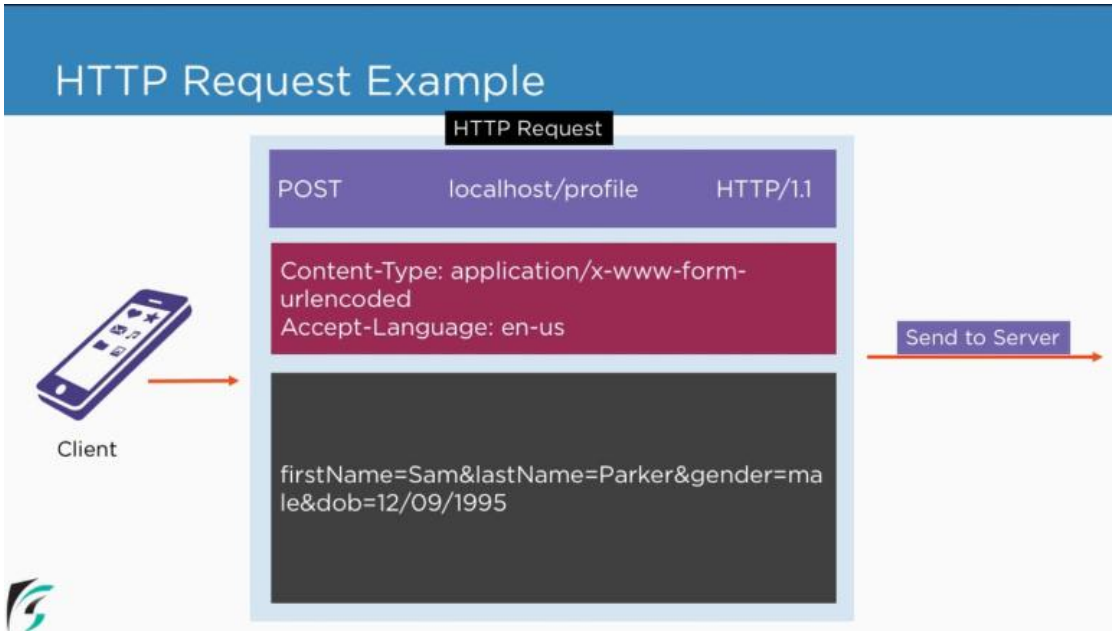
The functionality of the parts of these structure is as described below:

Request line : Method{GET,POST,PUT,PATCH,DELETE}, URL{With whom you want to communicate..location of server},HTTP version

Request Headers: Meta data eg. what kind of data you want to send, what kind of response you want to recieve, what language used in request

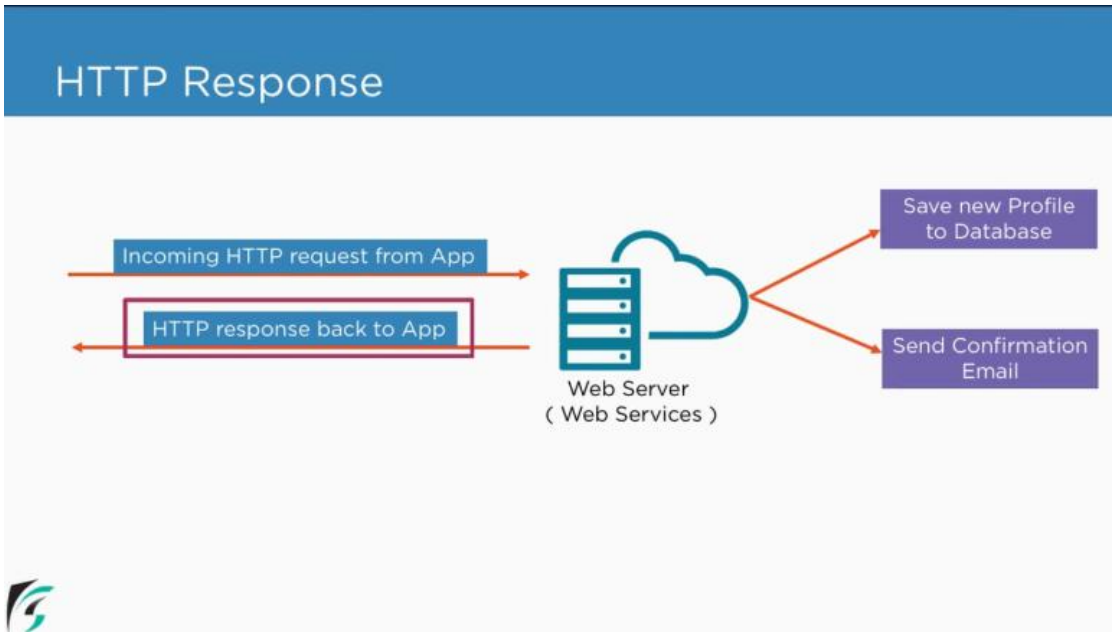
Request body: Contains data you want to send for processing eg JSON data for Student object

A HTTP request with all sections is use is as shown below:



HTTP response

The main function of a HTTP response is to return a flag indicating whether the request from the client is successful or not. If we use the HTTP request example of a user signing in as described in the HTTP request section, once the WebServer receives this request it will store the user in the database and send a confirmation message. If the entire operation was a success it will send a flag in the form of a HTTP response as shown in the pictures below:



Web Server executing the functions of inserting and saving user in DB and sending a confirmation message

HTTP response returning a confirmation flag in the form of a status code

Web service -> A program running inside the web server whose function is to read the request, process it and generate the client response

RESTful web service

Stateless -> In a nutshell clients requests are unaware of each other making the relationship between client and server loosely coupled. This enables one to change the UI of the client without altering the web server

Utilize HTTP methods -> GET,POST,PUT,PATCH,DELETE

HTTP Method	Consistent URL	Web Service Operation
GET	smartherd.com/user	Fetch User
POST	smartherd.com/user	Add User
PUT	smartherd.com/user	Update User
DELETE	smartherd.com/user	Delete User

Utilizing HTTP Methods

Implemented through structured URL -> Since we utilize the HTTP methods, we can use the same URL but with different methods to distinguish the operations that the client is requesting the web serer to perform

You can add identifiers such as ID to request action to a very specific element in the web server

HTTP Method	Consistent URL	Web Service Operation
GET	smartherd.com/user/3	Get user with ID of 3
DELETE	smartherd.com/user/3/comments	Delete comments of user whose ID is 3
GET	smartherd.com/user/3/name	Get name of the user whose ID is 3

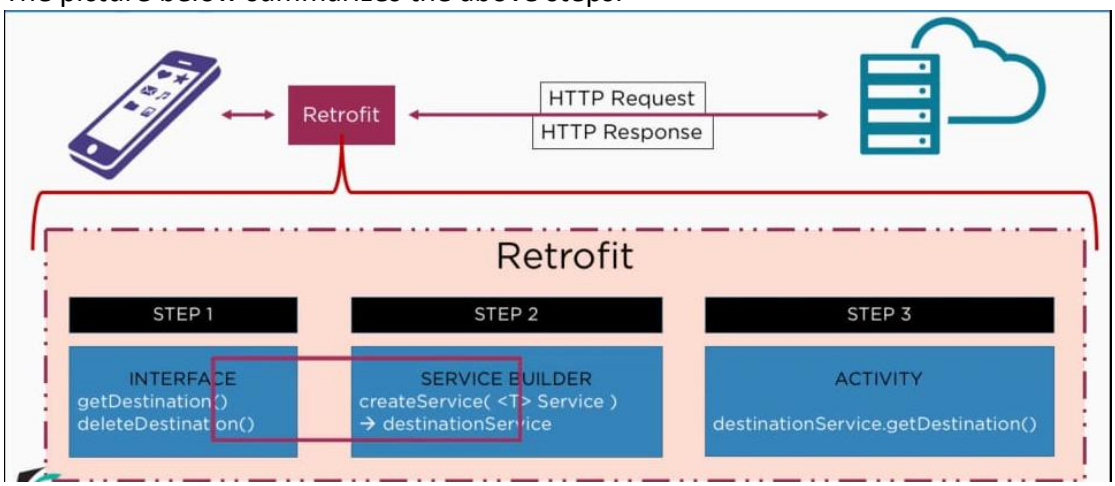
Can use different data types to transfer information -> Uses JSON to send data of different types

Retrofit

The retrofit interface has 3 main steps to enable its functionality

1. Create an interface that will contain various functions that will map to the end point of the web service. The functions can include getDestination()
2. Create a service that will help you to call the functions described in the interface
3. Within the activity that needs the web server, initialize the service and use it to call the functions described in the interface

The picture below summarizes the above steps:



Request parameters

The values highlighted in red are examples of request parameters and they enable the server to know what exactly the client is looking for:

There are two main ways of constructing URL'S in retrofit:

1. **Path parameters :** Adding URL segments to the main URL
2. **Query parameters :** Adding query strings to URL

The image below clearly distinguishes the two:

```
// Path Parameters: If you want to identify a particular resource
www.smartherd.com/users/47        // Retrieve user who has id 47

// Query Parameters: If you want to sort or filter items
www.smartherd.com/users?count=3   // Retrieve first three users
www.smartherd.com/users?occupation=doctor      // Retrieve all Doctors

// QueryMap: When you have a lot of Query Params then use QueryMap
www.smartherd.com/users?count=3&country=India
// Retrieve first three users who belongs to India
```

You can also have situations where both the path parameter and the query parameter are used at the same time as shown below:

www.xdr.com/users/47?/country=India

In the example above the path parameter section is : {**/users/47**}
While the query parameter is : {**?/country=India**}

Data is sent to the server on the request body of the HTTP request. There are two data formats used :

- JSON
- FormURLEncoded

The picture below shows how the 2 data formats can be used to send data: