**Channel API:**

Stack/Databases Used:

Play,Scala,Kafka,MongoDB,MySQL

**ENDPOINT**:

/api/v1/updateByChannels

**METHOD-TYPE:**

POST

**REQUEST PAYLOAD:**

{  
 "authenticationType": "mobile",  
 "authenticationValue": "9876543210",  
 "channels": [  
 {  
 "channelId": "CH001",  
 "channelName": "Facebook"  
 },

{  
 "channelId": "CH002",  
 "channelName": "Twitter"  
 }  
 ],  
 "subscriptionType": "Status",  
 "userDataMessage": "This is my new status",  
 "userDataMetadata": "",  
 "timeStamp": "24-08-1997",  
 "referrer": "https://channel.nowfloats.com/"  
}

**Code Structure:**

When the API recieves request it will validate with a case class and validate the request.Failure of validation we will throw error as response

After request validation the method queries the DB to get supported channels/subscriptiontypes(status/location)/limit(charlimit,filelimit etc.,).

Data limit has been applied only for status method and for all the other subscriptiontypes(video,audio etc.,) restrictions can be done by getting file size from the url etc.,

After getting avalid and filtered request we call a producer actor to stream the data to a kafka topic.

Now before pushing to kafka topic,the request can be validated using AVRO schematizer.This will help us preserve the topic schema through multiple versions.

The advantage of using kafka topic over cached database is that since we have unique request everytime due to timestamp,kafka can process multiple processes by creating threads simultaneously and also kafka provides partitions to be able to disstribute the data across multiple servers for heavy traffic.

The consumer runs as a background application and will be on constant pace with topic and every time a new record gets pushed into the topic the consumer triggers and runs multiple processes asynchronously.

Consumer will then hit the respective channel api's to update the data and once it recieves success from the channel it updates in MySQL TRANSACTION\_DETAILS database.For failure cases however the consumer will hit the channel api's again in a thread for 10 times and waits for a success response.After 10 attempts we update the record in database as failure.

Note: In case of queries regarding the database schema please go through MongoSchema.txt and MySQLSchema.txt