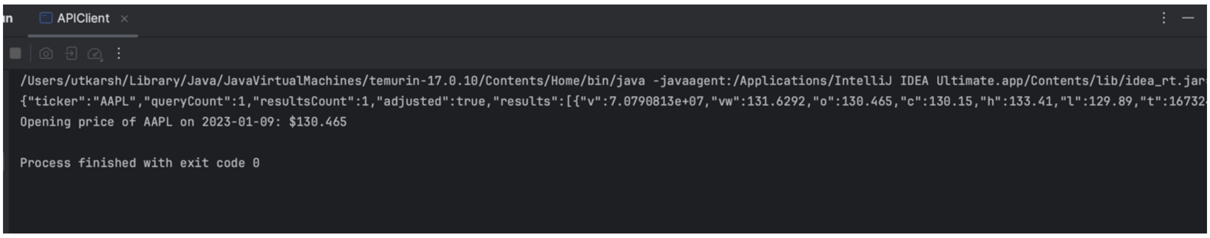
Polygon.io API URL
https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2023-01-09/2023-01-09?apiKey=LpLxmrARONFWxOriE1k4eMzBCDEG2cBT

My mobile application, "TradeHub," will prompt users to input their preferred stocks or financial instruments, leveraging the Polygon.io API to retrieve real-time stock quotes, historical data, and insightful market analysis, facilitating informed trading decisions on the go.

```java
import org.json.JSONArray;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class APIClient {
    public static void main(String[] args) {
        try {
            // Define the URL of the API endpoint
            URL url = new URL("https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2023-01-09/2023-
01-09?apiKey=LpLxmrARONFWxOriE1k4eMzBCDEG2cBT");

            // Create a HttpURLConnection object to open a connection to the API
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("GET");

            // Set up reading from the connection
            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            StringBuilder response = new StringBuilder();
            String line;

            // Read the response from the API
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }

            // Close the reader and the connection
            reader.close();
            connection.disconnect();

            // Extracting and printing the opening price from the response
            double openingPrice = parseResponse(response.toString());
            System.out.println("Opening price of AAPL on 2023-01-09: $" + openingPrice);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // Method to parse the JSON response and extract the opening price
    private static double parseResponse(String responseData) {
        // Parse the JSON response as a JSON array
        System.out.println(responseData);
        JSONObject jsonObject = new JSONObject(responseData);
        JSONArray jsonArray = jsonObject.getJSONArray("results");

        // Assume the first element in the array contains the data for the specified date
        JSONObject firstDataPoint = jsonArray.getJSONObject(0);

        // Extract the opening price from the first data point
        double openingPrice = firstDataPoint.getDouble("o");

        return openingPrice;
    }
}
```

1b.



```
/Users/utkarsh/Library/Java/JavaVirtualMachines/temurin-17.0.10/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA Ultimate.app/Contents/lib/idea_rt.jar
{"ticker":"AAPL","queryCount":1,"resultsCount":1,"adjusted":true,"results":[{"v":7.0790813e+07,"vw":131.6292,"o":130.465,"c":130.15,"h":133.41,"l":129.89,"t":16732
Opening price of AAPL on 2023-01-09: $130.465

Process finished with exit code 0
```

2c.

```
import com.mongodb.ConnectionString;
import com.mongodb.MongoClientSettings;
import com.mongodb.MongoException;
import com.mongodb.ServerApi;
import com.mongodb.ServerApiVersion;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.client.MongoClients;


public class MongoClientConnection {
    public static void main(String[] args) {
        String connectionString =
"mongodb+srv://Username:<password>@cluster0.agmir6c.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";

        ServerApi serverApi = ServerApi.builder()
                .version(ServerApiVersion.V1)
                .build();

        MongoClientSettings settings = MongoClientSettings.builder()
                .applyConnectionString(new ConnectionString(connectionString))
                .serverApi(serverApi)
                .build();

        // Create a new client and connect to the server
        try (MongoClient mongoClient = MongoClients.create(settings)) {
            try {
                // Send a ping to confirm a successful connection
                MongoDatabase database = mongoClient.getDatabase("admin");
                database.runCommand(new Document("ping", 1));
                System.out.println("Pinged your deployment. You successfully connected to MongoDB!");
            } catch (MongoException e) {
                e.printStackTrace();
            }
        }
    }
}
```