**FIFA-WORLD-CUP-SCRAPE APPLICATION**

**FifaWorldCupScrapeApplication.java**

package com.example.fifaworldcup;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class FifaWorldCupScrapeApplication {

    public static void main(String[] args) {

        SpringApplication.run(FifaWorldCupScrapeApplication.class, args);

    }

}

**Dto -> MatchDetails.java**

package com.example.fifaworldcup.dto;

public class MatchDetails {

    private String year;

    private String winner;

    private String score;

    private String runners;

    public MatchDetails(String year, String winner, String score, String runners) {

        this.year = year;

        this.winner = winner;

        this.score = score;

        this.runners = runners;

    }

    public String getYear() {

        return year;

    }

    public String getWinner() {

        return winner;

    }

    public String getScore() {

        return score;

    }

    public String getRunners() {

        return runners;

    }

}

**Service-> ScraperService.java**

package com.example.fifaworldcup.service;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

import org.springframework.stereotype.Service;

import com.example.fifaworldcup.dto.MatchDetails;

@Service

public class ScraperService {

    public List<MatchDetails> scrapeFifaData() throws IOException {

        String url = "https://en.wikipedia.org/wiki/List\_of\_FIFA\_World\_Cup\_finals";

        Document document = Jsoup.connect(url).get();

        Elements tableRows = document.select("table.wikitable tbody tr");

        List<MatchDetails> matchDetailsList = new ArrayList<>();

        for (int i = 1; i <= 10; i++) {

            Element row = tableRows.get(i);

            Elements cols = row.select("td");

            MatchDetails details = new MatchDetails(

                    cols.get(0).text(), // Year

                    cols.get(1).text(), // Winner

                    cols.get(2).text(), // Score

                    cols.get(3).text() // Runners-up

            );

            matchDetailsList.add(details);

        }

        return matchDetailsList;

    }

}

**Service -> GoogleSheetsService.java**

package com.example.fifaworldcup.service;

import java.io.FileInputStream;

import java.io.IOException;

import java.security.GeneralSecurityException;

import java.util.List;

import org.springframework.stereotype.Service;

import com.google.api.client.googleapis.javanet.GoogleNetHttpTransport;

import com.google.api.client.json.JsonFactory;

import com.google.api.client.json.jackson2.JacksonFactory;

import com.google.api.services.sheets.v4.Sheets;

import com.google.api.services.sheets.v4.model.ValueRange;

import com.google.auth.http.HttpCredentialsAdapter;

import com.google.auth.oauth2.GoogleCredentials;

@Service

public class GoogleSheetsService {

        private static final String SPREADSHEET\_ID = "1-PIXS1xbU8h1szvV0AugTeb3HORDVY8PCzTRoFh2i64";

        private static final String RANGE = "Sheet1!A2:D";

        public void appendToGoogleSheet(List<List<Object>> data) throws IOException, GeneralSecurityException {

                GoogleCredentials credentials = GoogleCredentials

                                .fromStream(new FileInputStream("src/main/resources/credentials.json"))

                                .createScoped(List.of("https://www.googleapis.com/auth/spreadsheets"));

                JsonFactory jsonFactory = JacksonFactory.getDefaultInstance();

                Sheets sheetsService = new Sheets.Builder(

                                GoogleNetHttpTransport.newTrustedTransport(),

                                jsonFactory,

                                new HttpCredentialsAdapter(credentials))

                                .setApplicationName("FIFA World Cup Data Scraper")

                                .build();

                ValueRange body = new ValueRange().setValues(data);

                sheetsService.spreadsheets().values()

                                .append(SPREADSHEET\_ID, RANGE, body)

                                .setValueInputOption("RAW")

                                .execute();

        }

}

**Controller-> DataController.java**

package com.example.fifaworldcup.controller;

import java.io.IOException;

import java.security.GeneralSecurityException;

import java.util.ArrayList;

import java.util.List;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.example.fifaworldcup.service.GoogleSheetsService;

@RestController

public class DataController {

    private final GoogleSheetsService googleSheetsService;

    public DataController(GoogleSheetsService googleSheetsService) {

        this.googleSheetsService = googleSheetsService;

    }

    @PostMapping("/append")

    public void appendToGoogleSheet(@RequestBody List<List<String>> data) throws IOException {

        try {

            List<List<Object>> convertedData = new ArrayList<>();

            for (List<String> row : data) {

                convertedData.add(new ArrayList<>(row));

            }

            googleSheetsService.appendToGoogleSheet(convertedData);

        } catch (IOException e) {

            e.printStackTrace();

        } catch (GeneralSecurityException e) {

            e.printStackTrace();

        }

    }

}

**POM.XML**

<project xmlns="http://maven.apache.org/POM/4.0.0"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.4.0</version>

        <relativePath /> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.example</groupId>

    <artifactId>fifa-world-cup-scrape</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>fifa-world-cup-scrape</name>

    <description>Demo project for Spring Boot</description>

    <properties>

        <java.version>21</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.jsoup</groupId>

            <artifactId>jsoup</artifactId>

            <version>1.16.1</version>

        </dependency>

        <!-- Google OAuth2 authentication library -->

        <dependency>

            <groupId>com.google.auth</groupId>

            <artifactId>google-auth-library-oauth2-http</artifactId>

             <version>1.17.0</version>

        </dependency>

        <!-- Google HTTP transport -->

        <dependency>

            <groupId>com.google.http-client</groupId>

            <artifactId>google-http-client-gson</artifactId>

            <version>1.43.0</version>

        </dependency>

        <!-- Jackson JSON processing -->

        <dependency>

            <groupId>com.fasterxml.jackson.core</groupId>

            <artifactId>jackson-databind</artifactId>

            <version>2.15.2</version>

        </dependency>

        <!-- Google HTTP Client with Jackson -->

    <dependency>

        <groupId>com.google.http-client</groupId>

        <artifactId>google-http-client-jackson2</artifactId>

        <version>1.43.0</version>

    </dependency>

        <!-- Jackson core library -->

    <dependency>

        <groupId>com.fasterxml.jackson.core</groupId>

        <artifactId>jackson-core</artifactId>

        <version>2.15.2</version>

    </dependency>

        <dependency>

            <groupId>com.google.api-client</groupId>

            <artifactId>google-api-client</artifactId>

            <version>1.34.0</version>

        </dependency>

        <dependency>

            <groupId>com.google.apis</groupId>

            <artifactId>google-api-services-sheets</artifactId>

            <version>v4-rev20230815-2.0.0</version>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <repositories>

        <repository>

            <id>google-maven-repo</id>

            <name>Google Maven Repository</name>

            <url>https://maven-central.storage.googleapis.com</url>

        </repository>

    </repositories>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**MOCKED CREDENTIALS.JSON( I’m sharing some dummy credentials):**

{

"type": "service\_account",

"project\_id": "sample-project-12345",

"private\_key\_id": "098zyx765wvu432tsr210qpo987nml654kjh321gfe",

"private\_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASC...\n-----END PRIVATE KEY-----\n",

"client\_email": "sample-service-account@sample-project-12345.iam.gserviceaccount.com",

"client\_id": "123456789012345678901",

"auth\_uri": "https://accounts.google.com/o/oauth2/auth",

"token\_uri": "https://oauth2.googleapis.com/token",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/sample-service-account%40sample-project-12345.iam.gserviceaccount.com"

}

**OUTPUT :**





