Code Snippet:

val jsonData = """

     {

  "resourceType": "Bundle",

  "type": "transaction",

  "entry": [ {

    "fullUrl": "urn:uuid:8c95253e-8ee8-9ae8-6d40-021d702dc78e",

    "resource": {

      "resourceType": "Patient",

      "id": "8c95253e-8ee8-9ae8-6d40-021d702dc78e",

      "meta": {

        "profile": [ "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient" ]

      },

      "text": {

        "status": "generated",

        "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Generated by <a href=\"https://github.com/synthetichealth/synthea\">Synthea</a>.Version identifier: master-branch-latest\n .   Person seed: 8886041252665297312  Population seed: 13235</div>"

      },

      "extension": [ {

        "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-race",

        "extension": [ {

          "url": "ombCategory",

          "valueCoding": {

            "system": "urn:oid:2.16.840.1.113883.6.238",

            "code": "2106-3",

            "display": "White"

          }

        }, {

          "url": "text",

          "valueString": "White"

        } ]

      }, {

        "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-ethnicity",

        "extension": [ {

          "url": "ombCategory",

          "valueCoding": {

            "system": "urn:oid:2.16.840.1.113883.6.238",

            "code": "2186-5",

            "display": "Not Hispanic or Latino"

          }

        }, {

          "url": "text",

          "valueString": "Not Hispanic or Latino"

        } ]

      }, {

        "url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName",

        "valueString": "Holley125 Champlin946"

      }, {

        "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-birthsex",

        "valueCode": "M"

      }, {

        "url": "http://hl7.org/fhir/StructureDefinition/patient-birthPlace",

        "valueAddress": {

          "city": "Barnstable",

          "state": "Massachusetts",

          "country": "US"

        }

      }, {

        "url": "http://synthetichealth.github.io/synthea/disability-adjusted-life-years",

        "valueDecimal": 0.05826471038258488

      }, {

        "url": "http://synthetichealth.github.io/synthea/quality-adjusted-life-years",

        "valueDecimal": 52.94173528961741

      } ],

      "identifier": [ {

        "system": "https://github.com/synthetichealth/synthea",

        "value": "8c95253e-8ee8-9ae8-6d40-021d702dc78e"

      }, {

        "type": {

          "coding": [ {

            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",

            "code": "MR",

            "display": "Medical Record Number"

          } ],

          "text": "Medical Record Number"

        },

        "system": "http://hospital.smarthealthit.org",

        "value": "8c95253e-8ee8-9ae8-6d40-021d702dc78e"

      }, {

        "type": {

          "coding": [ {

            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",

            "code": "SS",

            "display": "Social Security Number"

          } ],

          "text": "Social Security Number"

        },

        "system": "http://hl7.org/fhir/sid/us-ssn",

        "value": "999-86-2571"

      }, {

        "type": {

          "coding": [ {

            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",

            "code": "DL",

            "display": "Driver's License"

          } ],

          "text": "Driver's License"

        },

        "system": "urn:oid:2.16.840.1.113883.4.3.25",

        "value": "S99949530"

      }, {

        "type": {

          "coding": [ {

            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",

            "code": "PPN",

            "display": "Passport Number"

          } ],

          "text": "Passport Number"

        },

        "system": "http://standardhealthrecord.org/fhir/StructureDefinition/passportNumber",

        "value": "X67249552X"

      } ],

      "name": [ {

        "use": "official",

        "family": "Dickens475",

        "given": [ "Aaron697" ],

        "prefix": [ "Mr." ]

      } ],

      "telecom": [ {

        "system": "phone",

        "value": "555-152-6034",

        "use": "home"

      } ],

      "gender": "male",

      "birthDate": "1944-08-28",

      "deceasedDateTime": "1998-08-15T13:05:53+01:00",

      "address": [ {

        "extension": [ {

          "url": "http://hl7.org/fhir/StructureDefinition/geolocation",

          "extension": [ {

            "url": "latitude",

            "valueDecimal": 42.13509612101196

          }, {

            "url": "longitude",

            "valueDecimal": -71.94977170576706

          } ]

        } ],

        "line": [ "859 Altenwerth Run Unit 88" ],

        "city": "Charlton",

        "state": "MA",

        "country": "US"

      } ],

      "maritalStatus": {

        "coding": [ {

          "system": "http://terminology.hl7.org/CodeSystem/v3-MaritalStatus",

          "code": "M",

          "display": "M"

        } ],

        "text": "M"

      },

      "multipleBirthBoolean": false,

      "communication": [ {

        "language": {

          "coding": [ {

            "system": "urn:ietf:bcp:47",

            "code": "en-US",

            "display": "English"

          } ],

          "text": "English"

        }

      } ]

    },

    "request": {

      "method": "POST",

      "url": "Patient"

 }

  } ]

}

    """

import org.apache.spark.sql.{DataFrame, SparkSession}

import org.apache.spark.sql.types.\_

    // Create a Spark session

    val spark = SparkSession.builder()

      .appName("Test")

      .master("local[\*]") // Use "local" for local testing

      .getOrCreate()

    // Define the schema with StructType and StructFields

val schema = StructType(Seq(

      StructField("resourceType", StringType),

      StructField("type", StringType),

      StructField("entry", ArrayType(StructType(Seq(

        StructField("fullUrl", StringType),

        StructField("resource", StructType(Seq(

          StructField("resourceType", StringType),

          StructField("id", StringType),

          StructField("meta", StructType(Seq(

            StructField("profile", ArrayType(StringType))

          ))),

          StructField("text", StructType(Seq(

            StructField("status", StringType),

            StructField("div", StringType)

          ))),

          StructField("extension", ArrayType(StructType(Seq(

            StructField("url", StringType),

            StructField("extension", ArrayType(StructType(Seq(

              StructField("url", StringType),

              StructField("valueCoding", StructType(Seq(

                StructField("system", StringType),

                StructField("code", StringType),

                StructField("display", StringType)

              )))

            )))

          ))))

            )

            ))))

        )))

      ))

    // Read JSON data into a DataFrame with the defined schema

    val df: DataFrame = spark.read.schema(schema).json(Seq(jsonData).toDS())

(or)

val df: DataFrame = spark.read.schema(schema).json(“dbfs://paths/file/”)

    // Show the DataFrame

    display(df)

df.select("resourceType","type","entry.resource.resourceType","entry.resource.id","entry.resource.meta.profile",

  "entry.resource.text.status","entry.resource.text.div","entry.resource.extension.url","entry.resource.extension.extension","entry.resource.extension.extension.url","entry.resource.extension.extension.valueCoding","entry.resource.extension.extension.valueCoding.system","entry.resource.extension.extension.valueCoding.display")

  // .withColumnRenamed("","")

  .write.mode("Overwrite")

  .insertInto("Test\_table")

  // .createOrReplaceTempView("Test\_table")

    // Stop the Spark session

    spark.stop()