Scalar User Defined Functions (UDFs)

Description

User-Defined Functions (UDFs) are user-programmable routines that act on one row. This documentation lists the classes that are required for creating and registering UDFs. It also contains examples that demonstrate how to define and register UDFs and invoke them in Spark SQL.

UserDefinedFunction

To define the properties of a user-defined function, the user can use some of the methods defined in this class.

- **asNonNullable(): UserDefinedFunction**

  Updates UserDefinedFunction to non-nullable.

- **asNondeterministic(): UserDefinedFunction**

  Updates UserDefinedFunction to nondeterministic.

- **withName(name: String): UserDefinedFunction**

  Updates UserDefinedFunction with a given name.

Examples

Scala Java

```
import org.apache.spark.sql.*;
import org.apache.spark.sql.api.java.UDF1;
import org.apache.spark.sql.expressions.UserDefinedFunction;
import static org.apache.spark.sql.functions.udf;
import org.apache.spark.sql.types.DataTypes;

SparkSession spark = SparkSession
  .builder()
  .appName("Java Spark SQL UDF scalar example")
  .getOrCreate();

// Define and register a zero-argument non-deterministic UDF
// UDF is deterministic by default, i.e. produces the same result for the same input.
UserDefinedFunction random = udf(
  () -> Math.random(), DataTypes.DoubleType
);
random.asNondeterministic();
spark.udf().register("random", random);
spark.sql("SELECT random()").show();
// +-------+
// |UDF()  |
// +-------+
// |xxxxxxx|
// +-------+

// Define and register a one-argument UDF
spark.udf().register("plusOne",
  (UDF1<Integer, Integer>) x -> x + 1, DataTypes.IntegerType);
spark.sql("SELECT plusOne(5)").show();
// +----------+
// |plusOne(5)|
// +----------+
// |         6|
// +----------+

// Define and register a two-argument UDF
UserDefinedFunction strLen = udf(
  (String s, Integer x) -> s.length() + x, DataTypes.IntegerType
);
spark.udf().register("strLen", strLen);
spark.sql("SELECT strLen('test', 1)").show();
// +------------+
// |UDF(test, 1)|
// +------------+
// |           5|
// +------------+

// UDF in a WHERE clause
spark.udf().register("oneArgFilter",
  (UDF1<Long, Boolean>) x -> x > 5, DataTypes.BooleanType);
spark.range(1, 10).createOrReplaceTempView("test");
spark.sql("SELECT * FROM test WHERE oneArgFilter(id)").show();
// +---+
// | id|
// +---+
// |  6|
// |  7|
// |  8|
// |  9|
// +---+
```

Find full example code at "examples/src/main/java/org/apache/spark/examples/sql/JavaUserDefinedScalar.java" in the Spark repo.

Related Statements

- User Defined Aggregate Functions (UDAFs)
- Integration with Hive UDFs/UDAFs/UDTFs