Talend Advanced Course - Resources

Dear student!
In this document you can find a recompilation of the most important code snippets, infos and links for your course.
Happy learning!

Cheers, Samuel

Instructor Profile Page

https://www.udemy.com/user/samuel-lenk/

YouTube Channel

https://www.youtube.com/c/SamuelLenk

Documentation Short Links

Formatter (Java SE 11 & JDK 11) https://bit.ly/formatString
String (Java SE 11 & JDK 11) https://bit.ly/stringDocs
BigDecimal (Java SE 11 & JDK 11)https://bit.ly/bigDecimal
SimpleDateFormat (Java SE 11 & JDK 11) https://bit.ly/SimpleDateFormats
System (Java SE 11 & JDK 11) https://bit.ly/propertiesDocs
StringUtils (Apache Commons Lang 3.12.0 API) https://bit.ly/apacheCommonsStringUtils
Apache Commons StringUtil docs https://bit.ly/mavenRepoApacheCommons3

Code Snippets & Information for Jobs

Section 3: Advanced Database Techniques

Lecture: Bulk Load

verify if bulk loading is enabled:

```
show global variables where Variable_name like 'local%';
```
works on MySql after:
```
set global local_infile=true;
```

Lecture: ELT vs. ETL

```
"SELECT
employees.ID, employees.NAME, employees.ID_MANAGER, managers.NAME
FROM
 employees INNER JOIN  " +context.inTableName+ " managers ON(  managers.ID =
employees.ID_MANAGER )"
```

Lecture: Slowly Changing Dimensions (SCD)

SCD types:
0 = retain original -> name
1 = overwrite -> age
2 = add new row (with timestamp / active flag / version) -> role
3 = add new attribute -> salary

Lecture: Transactions

```
"drop table if exists " + context.masterTable + ";"
"drop table if exists " + context.childTable + ";"
"create table " + context.masterTable + " (
      id int not null auto_increment,
      name varchar(10),
      primary key(id))
      engine=innodb;
"
"create table " + context.childTable + " (
      id_child int not null,
      years int)
      engine=innodb;
"
```

Section 5: More Integrations

Lecture: Email

```
"Sent from Talend at " +
TalendDate.formatDate("HH:mm:ss",TalendDate.getCurrentDate())
```

Section 6: Convert Data Types

Lecture: General Conversions

Widening

```
Byte.valueOf("" + row2.shortValue)
Integer.valueOf(row2.shortValue)
Long.valueOf(row2.shortValue)
Float.valueOf(0.0f + row2.shortValue)
Double.valueOf(0.0 + row2.shortValue)
```

Narrowing

```
row4.doubleValue.longValue()
```

Precision

```
String value = "0.1";
String result = "0.0";

// adding Double:
double valueD = Double.valueOf(value);
double resultD = Double.valueOf(result);
for (int i = 0; i < 10; i++) {
    resultD += valueD;
}
System.out.println("Double result: " + resultD);

// adding Float:
float valueF = Float.valueOf(value);
float resultF = Float.valueOf(result);
for (int i = 0; i < 10; i++) {
    resultF += valueF;
}
System.out.println("Float result: " + resultF);

// adding BidDecimal:
BigDecimal valueB = new BigDecimal(value);
BigDecimal resultB = new BigDecimal(result);
for (int i = 0; i < 10; i++) {
    resultB = resultB.add(valueB);
}
System.out.println("BigDecimal result: " + resultB);
```

Formatting

```
System.out.printf("some %s value: %,d%n", "Long", Long.MAX_VALUE);
System.out.printf("some %s value: %.3f%n", "Double", Math.PI);

String.format("some %s value: %,d%n", "Long", Long.MAX_VALUE)
String.format("some %s value: %.3f\n", "Double", Math.PI)
```

Lecture: Convert Strings

```
String abc = "\tabc or whatEVer else 4 want here ... ";
System.out.println("ORIGINAL INPUT: \"" + abc + "\"");

// all this and more comes out of the box with Java
System.out.println("charAt: " + abc.charAt(2));
System.out.println("codePointAt: " + abc.codePointAt(2));
System.out.println("compareTo: " + abc.compareTo(abc));
System.out.println("compareToIgnoreCase: " + abc.compareToIgnoreCase(abc));
System.out.println("concat: " + abc.concat(abc)); // equivalent -> abc + abc
System.out.println("contains: " + abc.contains(abc));
System.out.println("endsWith: " + abc.endsWith(abc));
System.out.println("hashCode: " + abc.hashCode());
System.out.println("indexOf: " + abc.indexOf(0));
System.out.println("lastIndexOf: " + abc.lastIndexOf("c"));
System.out.println("length: " + abc.length());
System.out.println("matches: " + abc.matches(abc));
System.out.println("replace: " + abc.replace("a", "B").replace("e", "#")); //
method chaining
System.out.println("startsWith: " + abc.startsWith(abc));
System.out.println("strip: " + abc.strip());
System.out.println("stripTrailing: " + abc.stripTrailing());
System.out.println("substring: " + abc.substring(0, 2));
System.out.println("toUpperCase: " + abc.toUpperCase());
System.out.println("toLowerCase: " + abc.toLowerCase());
// find Docs here: bit.ly/stringDocs
```

Lecture: Avoid NullPointerExceptions

TERNARY OPERATOR
-> avoid NullPoimterException
-> syntax: booleanExpression ? expression1 : expression2
-> useful (not only) for null checks

```
output_row.number = input_row.number;
// NOT GOOD:
// output_row.txt = input_row.txt.toUpperCase();
// BETTER: include NULL check
// output_row.txt = (null == input_row.txt) ? "EX-NULL-VALUE" :
input_row.txt.toUpperCase();
// EVEN BETTER: include NULL and DUMMY check
output_row.txt = (null == input_row.txt) ? "EX_NULL-VALUE" :
(input_row.txt.equals("dummy") ? "no valid value".toUpperCase() :
input_row.txt.toUpperCase());
output_row.date = input_row.date;
output_row.flag = input_row.flag;
```

Section 7: Use Talend And Custom Routines

Lecture: Character And String

```
String abc = "\tabc or whatEVer else 4 want here ... ";
System.out.println("ORIGINAL INPUT: \"" + abc + "\"");

// all this and more comes out of the box with Java
System.out.println("charAt: " + abc.charAt(2));
System.out.println("codePointAt: " + abc.codePointAt(2));
System.out.println("compareTo: " + abc.compareTo(abc));
System.out.println("compareToIgnoreCase: " + abc.compareToIgnoreCase(abc));
System.out.println("concat: " + abc.concat(abc)); // equivalent -> abc + abc
System.out.println("contains: " + abc.contains(abc));
System.out.println("endsWith: " + abc.endsWith(abc));
System.out.println("hashCode: " + abc.hashCode());
System.out.println("indexOf: " + abc.indexOf(0));
System.out.println("lastIndexOf: " + abc.lastIndexOf("c"));
System.out.println("length: " + abc.length());
System.out.println("matches: " + abc.matches(abc));
System.out.println("replace: " + abc.replace("a", "B").replace("e", "#")); //
method chaining
System.out.println("startsWith: " + abc.startsWith(abc));
System.out.println("strip: " + abc.strip());
System.out.println("stripTrailing: " + abc.stripTrailing());
System.out.println("substring: " + abc.substring(0, 2));
System.out.println("toUpperCase: " + abc.toUpperCase());
System.out.println("toLowerCase: " + abc.toLowerCase());
// find Docs here: bit.ly/stringDocs
```

Lecture: Date And Time

```
TalendDate.parseDate("EEE MMM dd yyyy hh:mm aa",row1.stringDate)
TalendDate.addDate(TalendDate.addDate(Var.vParsedDate, -45, "dd"), 11, "mm")
TalendDate.diffDate(row3.dateOne,row3.dateTwo,"dd")
```

Lecture: Arithmetic Operations

```
Mathematical.SADD(row1.numA, row1.numB)

System.out.println("## DEFINITION ##");
double a = 9.0;
System.out.println("a = " + a);
double b = 4.0;
System.out.println("b = " + b);
double result;

System.out.println("## BASIC OPERATIONS ##");

// addition:
result = a + b;
System.out.println("a + b = " + result);
result = Mathematical.SADD("" + a, "" + b);
System.out.println("a + b = " + result);
```

```
// subtraction:
result = a - b;
System.out.println("a - b = " + result);
String subResult = Mathematical.SSUB("" + a, "" + b);
System.out.println("a - b = " + subResult);

// muliplication:
result = a * b;
System.out.println("a * b = " + result);
result = Mathematical.SMUL("" + a, "" + b);
System.out.println("a * b = " + result);

// division:
result = a / b;
System.out.println("a : b = " + result);
result = Mathematical.DIV(a, b);
System.out.println("a : b = " + result);
result = Mathematical.SDIV((int)a, (int)b);
System.out.println("a : b = " + result);

// division remainder = modulo operator "%":
result = a % b;
System.out.println("a % b = " + result);
result = Mathematical.MOD(a, b);
System.out.println("a % b = " + result);

System.out.println("## OTHER METHODS ##");

System.out.println("floating to fixed width string: " + Mathematical.FFIX(a,
2));
System.out.println("return integer: " + Mathematical.INT("-3"));
System.out.println("make number negative: " + Mathematical.NEG(a));
System.out.println("make number positive: " + Mathematical.ABS(-3.45));
System.out.println("is numeric: " + Mathematical.NUM("3"));
System.out.println("random number: " + Mathematical.RND(10.5));
System.out.println("square root: " + Mathematical.SQRT(a));
```

Lecture: Currency Handling

```
import java.math.RoundingMode;

Var.usdAmountBD.setScale(2, RoundingMode.HALF_EVEN)
```

Lecture: Develop And Use Custom Routines

```
package routines;

import java.util.Arrays;
import java.util.Random;
import java.util.UUID;

public class DemoRoutine {

    /**
```

```
     * helloExample : return "hello" + message.
     * {talendTypes} String
     * {Category} User Defined
     * {param} string("world") input: The string to be included.
     * {example} helloExample("world") # hello world!.
     */
      public static String helloExample(String message) {
            if (message == null || message.equals("")) {
                  message = "World";
            }
            return "Hello " + message + "!";
      }

    /**
     * getZipCode : return random zip code.
     * {talendTypes} String
     * {Category} User Defined
     * {example} getZipCode(4) # 5782
     */
      public static String getZipCode(int len) {
            // get a good zip code length
            int zipLen = 0;
            if (len < 1 || len > 5) {
                  zipLen  = 4;
            } else {
                  zipLen = len;
            }
            // create the zip code
            StringBuilder sb = new StringBuilder(zipLen);
            Random random = new Random();
            int min = 0;
            int max = 9;
            for (int i = 0; i < zipLen; i++) {
                  sb.append(random.nextInt(max - min) + min);
            }
            return sb.toString();
      }

    /**
       * getRandomDay : Return a random open day
       * {talendTypes} String
       * {Category} User Defined
       * {example} getRandomDay() # Monday
       */
      public static String getRandomOpenDay() {
            String day[] = { "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday" };
            int i = ((Long) Math.round(Math.random() * (4))).intValue();
            return day[i];
      }

      /**
       * calculatePercentage : Get the percentage for a value from a reference
       * {talendTypes} Double
       * {Category} User Defined
     * {param} Double (value, reference) input: the value and reference to be
used.
```

```
       * {example} addPercent(105.0, 140.0) # returns 75.0
       */
      public static Double calculatePercentage(Double value, Double reference)
{
            return value * 100.0 / reference;
      }

      /**
       * getFileFetchUrl : Return the file fetch URL
       * {talendTypes} String
       * {Category} User Defined
       */
      public static String getFileFetchUrl() {
            return "http://talendforge.org/file_fetch.txt";
      }

      /**
     * getRandomUUID: return random UUID.
     * {talendTypes} String
     * {Category} User Defined
     * {example} getRandomUUID() # f901ed16-3910-4160-a69a-082bee15ff78
     */
      public static String getRandomUUID() {
            return "" + UUID.randomUUID();
      }

      /**
     * padString: pad a 'input' string to length 'len' with given
'paddingChar'.
     * {talendTypes} String
     * {Category} User Defined
     * {example} padString("Hello", 7, '#') # "Hello##"
     */
      public static String padString(String input, int len, char paddingChar) {
            if (null == input || len == 0) {
            return null;
        }
            if (len < input.length()) {
            return input.substring(0,len);
        }
        char[] target = new char[len - input.length()];
        Arrays.fill(target, paddingChar);
        return input + new String(target);
      }

}

String greeting = DemoRoutine.helloExample(" ");
System.out.println(greeting);

DemoRoutine.calculatePercentage(row2.value + 0.0, row2.reference + 0.0)

DemoRoutine.getFileFetchUrl()

DemoRoutine.getRandomOpenDay()
```

Section 8: Custom Java Code

Lecture: tJava Component

```
String name = "Samuel";
System.out.println("My name: " + name);

System.out.println("My int: " + Numeric.random(0,10));
```

Lecture: tJavaRow Component

```
//Code generated according to input schema and output schema
output_row.number = input_row.number;
output_row.txt = input_row.txt;
output_row.date = input_row.date;

if (input_row.flag) { // = true
	output_row.flag = 1;
} else { // = false
	output_row.flag = 0;
}
// equivalent one-liner:
// output_row.flag = (input_row.flag) ? 1 : 0;
```

Lecture: tJavaFlex Component

Start:

```
System.out.println("## START ##");
int i = 1;
```

Main:

```
System.out.print("# row" + i);
System.out.print(" # number:" + row4.number);
System.out.print(" | txt:" + row4.txt.toUpperCase());
System.out.print(" | date:" + row4.date);
if(row4.flag) {
	System.out.println(" | flag:1");
} else {
	System.out.println(" | flag:0");
}
i++; // i = i + 1;
```

End:

```
System.out.println("## END ##");
```

Lecture: Java Properties

```
import java.util.Properties;
import java.util.Set;
Properties props = System.getProperties();
Set<String> keys = props.stringPropertyNames();

for (String key : keys) {
	if (key.contains("user")) {
		System.out.println(key + " = " + props.getProperty(key));
	}
}
```

```
import java.util.Map;

Map<String, String> env = System.getenv();

for (String envName : env.keySet()) {
	System.out.format("%s=%s%n", envName, env.get(envName));
}
```

Lecture: Use External Libraries

```
StringUtils.reverse(row1.myString)
StringUtils.countMatches(row1.myString, "e")
StringUtils.swapCase(row1.myString)
StringUtils.capitalize(row1.myString)
StringUtils.repeat(row1.myString, "-", 3)
StringUtils.defaultString(null, "NULL")
```

Section 10: Use Cases

Lecture: Nested Conditions with Ternary Operator and If Statements

```
row2.name.toLowerCase().charAt(0)
```

```
Var.vC >= 'a' && Var.vC <= 'h' ? "A-H" :
Var.vC >= 'i' && Var.vC <= 'p' ? "I-P" :
Var.vC >= 'q' && Var.vC <= 'z' ? "Q-Z" :
"unknown"
```

```
output_row.id = input_row.id;
output_row.name = input_row.name;
if (input_row.name == null || input_row.name.length() == 0) {
	output_row.nameGroup = "unknown";
} else {
	char c = input_row.name.toLowerCase().charAt(0);
	if (c >= 'a' && c <= 'h') {
		output_row.nameGroup = "A-H";
	} else if (c >= 'i' && c <= 'p') {
		output_row.nameGroup = "I-P";
```

```
	} else if (c >= 'q' && c <= 'z') {
		output_row.nameGroup = "Q-Z";
	} else {
		output_row.nameGroup = "unknown";
	}
}
```

Lecture: Iterate Over Date And Time

```
TalendDate.addDate(context.startDate, context.increment *
(((Integer)globalMap.get("tLoop_1_CURRENT_ITERATION"))-1), context.datePortion)
TalendDate.addDate(context.startDate, context.increment *
((Integer)globalMap.get("tLoop_1_CURRENT_ITERATION")), context.datePortion)
```

Lecture: Cancel Jobs With tJavaRow

```
output_row.id = input_row.id;
output_row.name = input_row.name;
if (null == input_row.country || input_row.country.equals("") ||
input_row.country.equals("N/A")) {
	System.out.println("Id '" + input_row.id + "' has no valid country (" +
input_row.country + ")");
	System.exit(99);
} else {
	output_row.country = input_row.country;
}
```

Section 12: Bonus

Lecture: Data Tests

```
output_row.name = (null == input_row.name) ? "NULL" : input_row.name;
```

```
row4.name.equals("NULL") || row4.name.equals(row8.name.toUpperCase())
```

What's next?

Learn Java!
This way you can not only use Talend even more professionally, but you can program nearly any type of device you can imagine.

If you have any queries, please feel free to contact me at samuel@diamond-ug.de