

Un poco de Java y +

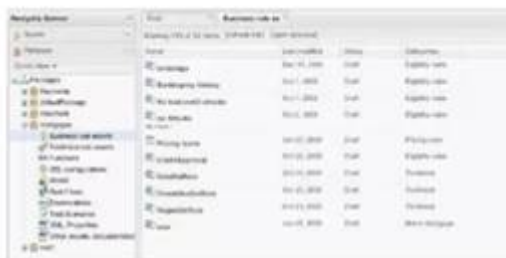
Otra forma de hablar de nuestro día a día...

JAVA, TUTORIALES

¿Cómo usar Drools con Tablas de Decisión en excel?

Date: 18 julio 2018 **Author:** LuisMi Gracia ☐ 0 Comentarios

Hacía años (muchos años) que no hablábamos de **Drools**, con decirnos que este post es de 2010...!!!



Rule	When	Then	Comments
1. Customer is a client	Customer is a client	Customer is a client	
2. Customer is a client	Customer is a client	Customer is a client	
3. Customer is a client	Customer is a client	Customer is a client	
4. Customer is a client	Customer is a client	Customer is a client	
5. Customer is a client	Customer is a client	Customer is a client	
6. Customer is a client	Customer is a client	Customer is a client	
7. Customer is a client	Customer is a client	Customer is a client	
8. Customer is a client	Customer is a client	Customer is a client	
9. Customer is a client	Customer is a client	Customer is a client	
10. Customer is a client	Customer is a client	Customer is a client	

Drools Versión 5.1:
Conceptos Básicos

24 agosto 2010 — 0 Comentarios

A día de hoy Drools va por su versión 7.8, se ha integrado con jBPM (como motor de procesos) y forma parte de la suite de Red Hat.

Pero hoy queríamos ver cómo usar tablas de decisión con Drools, de modo que pueda definir las reglas en Excel, y lo veremos con un ejemplo (original aquí):

En el ejemplo queremos definir el descuento para un Cliente en función del tipo de Cliente (individual o empresa) y el número de años como cliente.

Para crearlo:

Tengo mi clase **Customer**:

```

public class Customer {

    private CustomerType type;

    private int years;

    private int discount;

    public enum CustomerType {
        INDIVIDUAL, BUSINESS;
    }

    public Customer(CustomerType type, int numOfYear) {
        super();
        this.type = type;
        this.years = numOfYear;
    }
}

```

Mi Tabla de decisión es un XLS como este:

RuleSet	com.baeldung.drools			
Import	com.baeldung.drools.model.Customer, com.baeldung.drools.model.Customer.CustomerType			
Rule Table Discount rates				
NAME	CONDITION	CONDITION	CONDITION	ACTION
	\$customer:Customer			
	\$customer.getType() in (\$param)	\$customer.getYears() >= (\$param)	\$customer.getYears() < (\$param)	\$customer.setDiscount(\$param);
NAME	Customer type	Years	Years	Discount
Individual > 3y	CustomerType.INDIVIDUAL	3		15
Individual < 3y	CustomerType.INDIVIDUAL	0	3	5
Business Any	CustomerType.BUSINESS			20

Que es bastante autoexplicativo:

- Defino los Imports del RuleSet (la clase Customer y CustomerType)
- Defino las condiciones (tantas como quiera) en **CONDITION** y la acción (**ACTION**). En el ejemplo en función del tipo de Cliente y de los años que lleva de cliente defino el descuento, que cargo sobre el mismo objeto "customer"

Y finalmente aplico las reglas, por ejemplo en un Test tendría:

El setup donde cargo el recurso y creo la sesión:

```

@Before
public void setup() {
    Resource resource = ResourceFactory.newClassPathResource("com/baeldung/drools/rules/Discount.xls", getClass());
    kSession = new DroolsBeanFactory().getKieSession(resource);
}

```

Y el lanzamiento de las reglas, en este caso para un cliente personal con 5 años de cliente:

```

@Test
public void giveIndividualLongStanding_whenFireRule_thenCorrectDiscount() throws Exception {
    Customer customer = new Customer(CustomerType.INDIVIDUAL, 5);
    kSession.insert(customer);

    kSession.fireAllRules();

    assertEquals(customer.getDiscount(), 15);
}

```

En nuestro proyecto Maven tendré que añadir estas dependencias:

```
<!-- ... -->
<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-ci</artifactId>
  <version>${drools-version}</version>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-decisiontables</artifactId>
  <version>${drools-version}</version>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-core</artifactId>
  <version>${drools-version}</version>
</dependency>
<dependency>
  <groupId>org.drools</groupId>
  <artifactId>drools-compiler</artifactId>
  <version>${drools-version}</version>
</dependency>
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>${apache-poi-version}</version>
</dependency>
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
  <version>${apache-poi-version}</version>
</dependency>
```

Donde drools-version=7.4.1.Final

© 2018 UN POCO DE JAVA Y +

