



2023 RULE TEMPLATE

Knowledge based systems



Rule template

- Predstavljaju način za generisanje DRL pravila u run-time-u koristeći template file-ove i tabularne izvore podataka
- Pod tabularnim izvorima podataka podrazumevaju se podaci koji mogu biti predstavljeni tabelama (spreadsheets i tabele u bazama podataka)
- Na ovaj način je moguće generisati pravila sa podacima koji se nalaze van same aplikacije
- Osnovna prednost korišćenja rule template-a ogleda se u tome da se:
 - Jedan rule template file može koristiti za različite setove podataka
 - Jedan set podataka može koristiti za različite template file-ove
- Rule template-i omogućavaju:
 1. Skladištenje podataka u baze podataka (u bilo kom formatu)
 2. Uslovno generisanje pravila na osnovu vrednosti u podacima
 3. Korišćenje podataka u bilo kom delu pravila (operatorima, nazivima klasa, property-jima klasa itd.)
 4. Izvršavanje različitih template-a nad istim podacima

Struktura rule template file-a

- Drools rule template je tekstualni file koji sadrži specijalne ključne reči kojima se označavaju:
 - različite sekcije template-a
 - kao i promenljive i njihove namene
- Ekstenzija za template file-ove je .drt

Struktura rule template file-a

```
template header
2  age
3  type
4  log
5
6  package org.drools.examples.templates;
7
8  global java.util.List list;
9
10 template "cheesefans"
11
12 rule "Cheese fans_{row.rowNumber}"
13 when
14     Person(age == @{age})
15     Cheese(type == "{@type}")
16 then
17     list.add("@{log}");
18 end
19
20 end template
```

- Linija 1: Svi template-i počinju sa *template header*
- Linije 2-4: Nazivi promenljivih - definišu se inline i nisu deo data set-a
- Linija 5: Prazan red označava kraj zaglavlja
- Linije 6-9: Naziv paketa i import sekcija (unutar ove sekcije definišu se i globali, type declarations i funkcije)
- Linija 10: Ključna reč template označava početak template-a. Jedan dokument može sadržati više template-a koji moraju imati jedinstvene nazive
- Linije 12-18: Telo template-a
- Linija 20: Ključna reč end template koja označava kraj template-a
- Linija 12: *@{row.rowNumber}*, je ugrađeni izraz koji kreira jedinstveni ID pravila.
- Linija 14-17: MVEL izrazi koji se zamenjuju template parametrima u run-time

Array data source

- Jedan od načina da se template-ima proslede podaci jeste korišćenjem dvodimenzionalnog niza stringova.
- Prva dimenzija predstavlja red, a druga kolonu.
- Podaci se enkapsuliraju unutar *ArrayDataProvider* klase koja implementira *DataProvider* interfejs.
- Konekcija između podataka i template-a se ostvaruje putem *DataProviderComplier* klase.

Array data source

```
InputStream template = RuleTemplatesTest.class.getResourceAsStream(name: "/template-dtable/  
customer-classification-simple.drt");
```

```
DataProvider dataProvider = new ArrayDataProvider(new String[][]{  
    new String[]{"18", "21", "NA", "NA"},  
    new String[]{"22", "30", "NA", "BRONZE"},  
    new String[]{"31", "40", "NA", "SILVER"},  
    new String[]{"41", "150", "NA", "GOLD"},  
});
```

```
DataProviderCompiler converter = new DataProviderCompiler();  
String drl = converter.compile(dataProvider, template);
```

Objects data source

- Kolekcije objekata se takođe mogu koristiti kao način predstavljanja podataka rule template-a
- U ovom slučaju, nazivi promenljivih unutar template-a moraju da se poklapaju sa nazivima polja u objektima
- Konekcija između podataka i template-a se ostvaruje putem *ObjectDataComplier* klase

```
public class ClassificationTemplateModel {  
    private int minAge;  
    private int maxAge;  
    private Customer.Category previousCategory;  
    private Customer.Category newCategory;  
}
```

Objects data source

```
InputStream template = RuleTemplatesTest.class.getResourceAsStream(name: "/template-dtable/customer-classification-simple.drt");
```

```
List<ClassificationTemplateModel> data = new ArrayList<ClassificationTemplateModel>();
```

```
data.add(new ClassificationTemplateModel(minAge: 18, maxAge: 21, Customer.Category.NA, Customer.Category.NA));  
data.add(new ClassificationTemplateModel(minAge: 22, maxAge: 30, Customer.Category.NA, Customer.Category.BRONZE));  
data.add(new ClassificationTemplateModel(minAge: 31, maxAge: 40, Customer.Category.NA, Customer.Category.SILVER));  
data.add(new ClassificationTemplateModel(minAge: 41, maxAge: 150, Customer.Category.NA, Customer.Category.GOLD));
```

```
ObjectDataCompiler converter = new ObjectDataCompiler();  
String drl = converter.compile(data, template);
```


Spreadsheet data source

- Korišćenjem spreadsheet-a je moguće proslediti podatke template-u
- Konekcija između podataka i template-a se ostvaruje putem *ExternalSpreadsheetCompiler* klase (potrebno dodati dependency *drools-decisiontables*)
- *compile* metodi se prosleđuju podaci, template, broj reda prvog podatka, broj kolone prvog podatka

	A	B	C	D	E
1					
2		More than	Less or equal to	Without previous Category	Set Category
3		18	21	NA	NA
4		22	30	NA	BRONZE
5		31	40	NA	SILVER
6		41	150	NA	GOLD

Spreadsheet data source

```
InputStream template = RuleTemplatesTest.class.getResourceAsStream(name: "/templetable/customer-classification-simple.drt");  
InputStream data = RuleTemplatesTest.class.getResourceAsStream(name: "/templetable/template-data.xls");  
  
ExternalSpreadsheetCompiler converter = new ExternalSpreadsheetCompiler();  
String drl = converter.compile(data, template, 3, 2);
```

Generisana pravila

```
1 package chapter07.ruletemplate.simple;
2
3 import com.sample.Customer;
4
5 rule "Categorize Customers_3"
6 no-loop true
7 when
8     $c: Customer(age >= 41, age <= 150, category == Customer.Category.NA)
9 then
10     modify ($c){ setCategory(Customer.Category.GOLD)};
11 end
12
13 rule "Categorize Customers_2"
14 no-loop true
15 when
16     $c: Customer(age >= 31, age <= 40, category == Customer.Category.NA)
17 then
18     modify ($c){ setCategory(Customer.Category.SILVER)};
19 end
20
21 rule "Categorize Customers_1"
22 no-loop true
23 when
24     $c: Customer(age >= 22, age <= 30, category == Customer.Category.NA)
25 then
26     modify ($c){ setCategory(Customer.Category.BRONZE)};
27 end
28
29 rule "Categorize Customers_0"
30 no-loop true
31 when
32     $c: Customer(age >= 18, age <= 21, category == Customer.Category.NA)
33 then
34     modify ($c){ setCategory(Customer.Category.NA)};
35 end
```