

Редактор бизнес правил на Play и Slick.

Данное веб приложение позволит создавать наборы бизнес правил и обращаться к ним при помощи веб сервисов. В качестве языка для правил был выбран Python, среда разработки – IntelliJ J с плагином для Скалы. Клиентская часть кода написана на CoffeeScript с использованием JQuery. Далее будет описан результат моей работы.


Веб приложение позволяет создавать и редактировать наборы бизнес правил. Веб Сервис для каждого набора бизнес правил генерируется автоматически, по определенным для бизнес правила типам входящих и исходящих данных.

Редактор бизнес правил

admin

ComponentsSettingsMonitoring

English

 **Repository / Rule Set: BasicRuleSet**

Create New Version

Metadata

Id	3	
Api	3c60bcd2984494fb6a3eb54b3ea44042	SOAP Service Definition by sequence
Name	BasicRuleSet	SOAP Service Definition by sequence and api
Sequence	CMP0	SOAP Service Definition by id

Arguments

Name	Type
argument1	string

Results

Name	Type
result1	string

Rules

N	Example
?	argument1 == "OK"
=	result1 = "Success"

На данном скриншоте изображен редактор набора бизнес правил в режиме просмотра. Сверху перечислены свойства, например имя набора правил, версия и т.д. Ниже виден код правил, где каждое правило состоит из названия (оно не обязательно, но упрощает отладку), условия (в случае выполнения которого будет исполнено тело правила) тела правила (где формируется результат) и нескольких ссылок на WSDL описание веб сервиса. Для каждого набора правил доступно описание веб сервиса с XSD схемой типов данных.

Автоматически генерируемый WSDL для набора правил определенных выше:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/soap/" xmlns:tns="http://example.com/rules/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Rules" targetNamespace="http://example.com/rules/">
  <types>
    <xsd:schema targetNamespace="http://example.com/rules/">
      <xsd:element name="rulesRequest">
        <xsd:complexType base="xsd:anyType" minOccurs="1" maxOccurs="1">
          <xsd:sequence base="xsd:anyType" minOccurs="1" maxOccurs="1">
            <xsd:element name="argument1" type="xsd:string" maxOccurs="1" minOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="rulesResponse">
        <xsd:complexType base="xsd:anyType" minOccurs="1" maxOccurs="1">
          <xsd:sequence base="xsd:anyType" minOccurs="1" maxOccurs="1">
            <xsd:element name="rule" maxOccurs="unbounded" minOccurs="0">
              <xsd:complexType base="xsd:anyType" minOccurs="1" maxOccurs="1">
                <xsd:sequence base="xsd:anyType" minOccurs="1" maxOccurs="1">
                  <xsd:element name="result1" type="xsd:string" maxOccurs="1" minOccurs="1"/>
                  <xsd:element name="name" type="xsd:string" maxOccurs="1" minOccurs="1"/>
                  <xsd:element name="condition" type="xsd:boolean" maxOccurs="1" minOccurs="1"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="rulesFault">
        <xsd:complexType base="xsd:anyType" minOccurs="1" maxOccurs="1">
          <xsd:sequence base="xsd:anyType" minOccurs="1" maxOccurs="1">
            <xsd:element name="code" type="xsd:int" maxOccurs="1" minOccurs="1"/>
            <xsd:element name="message" type="xsd:string" maxOccurs="1" minOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="rulesRequest">
    <part element="tns:rulesRequest" name="parameters"/>
  </message>
  <message name="rulesResponse">
    <part element="tns:rulesResponse" name="parameters"/>
  </message>
  <message name="rulesFault">
    <part element="tns:rulesFault" name="parameters"/>
  </message>
  <portType name="Rules">
    <operation name="invoke">
      <input message="tns:rulesRequest" type="tns:rulesRequest"/>
      <output message="tns:rulesResponse" type="tns:rulesResponse"/>
      <fault name="fault" message="tns:rulesFault" type="tns:rulesFault"/>
    </operation>
  </portType>
  <binding name="RulesSOAP" type="tns:Rules">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="invoke">
      <soap:operation soapAction="http://example.com/rules/invoke"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="Rules">
    <documentation>Service for rule set: BasicRuleSet (id:3) => 3</documentation>
    <port binding="tns:RulesSOAP" name="RulesSOAP">
      <soap:address location="https://localhost/interface/rules/id:3/soap"/>
    </port>
  </service>
</definitions>
```



rulesRequest содержит описание аргументов бизнес правил. В данном случае, это “argument1” типа строка.


rulesResponse это список типов rule, которые, в свою очередь, описывают результат выполнения каждого правила. Тип rule содержит возвращаемые данные (result1 типа строка) и атрибуты с именем парвила и результатом выполнения условия.

rulesFault определяет тип ошибки.

Редактор правил в режиме редактирования

Что-бы изменить правила, необходимо создать новую версию правила, где система позволит редактировать код и некоторые свойства.

admin  Components Settings Monitoring English 




Repository / Rule Set: New RuleSet (Uncommitted)

SaveRemove

Metadata


Id	4	
Api	3c60bcd2984494fb6a3eb54b3ea44042	SOAP Service Definition by sequence
Name	<input type="text" value="New RuleSet"/>	SOAP Service Definition by sequence and api
Sequence	<input type="text" value="CMz9"/>	SOAP Service Definition by id

Arguments

Name	Type	
<input type="text" value="argument1"/>	string	

Add


Results

Name	Type	
<input type="text" value="result1"/>	string	

Add

Rules

Add

N	<input type="text" value="Example"/>	
?	<input ok\""="" type="text" value="argument1 == \"/>	
=	<input success\""="" type="text" value="result1 = \"/>	

Add

При редактировании пользователь может сохранять изменения на сервере, которые остаются видны только ему . Хотя непримененные(uncommitted) наборы правил можно вызывать по уникальному идентификатору версии, они еще могут быть удалены и не являются последней версией правил, если обращение происходит не по уникальному имени версии, а по идентификатору правила (которое может иметь несколько версий).

Так как система различает разные версии одного набора правил, их можно выбирать при помощи фильтров, например:

<https://localhost/interface/rules/id:7/soap?wsdl> - Описание правил версии 7.

<https://localhost/interface/rules/seq:CMe3/soap?wsdl> – Описание последней версии правил **CMe3**

<https://localhost/interface/rules/seq:CMe3;api:4a75056b2f7396d1aa2b49343d6281a5/soap?wsdl> – Описание последней версии правил **CMe3** интерфейс которых соответствует коду **4a75056b2f7396d1aa2b49343d6281a5**. Идентификатор интерфейса вычисляется из описания типов аргументов и результата.

Репозиторий

Для версионирования правил имеется Репозиторий, где можно увидеть все доступные в системе наборы правил, а также историю их изменений. Ранее примененные наборы правил не могут быть удалены.

The screenshot displays the 'Repository' management interface. At the top, there is a navigation bar with 'admin', 'Components', 'Settings', and 'Monitoring' links, along with a language dropdown set to 'English'. Below the navigation bar, the 'Repository' title is accompanied by a folder icon. To the right, there is an 'Updates' section with 'Apply' and 'Cancel' buttons. The main content area is titled 'List of Components' and includes 'Create...' and 'Upload' buttons. A table lists the components, with the last row highlighted in green. The table has columns for Version ID, Component ID, Type, Name, Status, and Actions. Pagination controls at the bottom right show '1' and '20'.

Version ID	Component ID	Type	Name	Status	Actions
3	CMPO	rules	BasicRuleSet	Active	Open
2	CMe3	rules	MyRuleSet	Active	Open
1	CMYe	types	MyType	Active	Open
4	CMz9	rules	New RuleSet		Open, Remove

История правила

История одного правила показана на следующем снимке:

The screenshot shows a web interface for a repository. At the top, there is a navigation bar with 'admin' and a link icon, and tabs for 'Components', 'Settings', and 'Monitoring'. A language dropdown menu is set to 'English'. Below the navigation bar, the page title is 'Repository / MyRuleSet' with a folder icon. The main content area is titled 'History of component: CMe3'. It contains a table with the following data:

ID	Status	Type	Name	Commit	Actions
7		rules	MyRuleSet		Open
6	Active	rules	MyRuleSet	admin (19 November 2015 21:16)	Open
2		rules	MyRuleSet	admin (19 November 2015 20:50)	Open

At the bottom right of the table, there are pagination controls showing '1' and '20' with a dropdown arrow.


Типы данных

Помимо правил, можно определять сложные типы данных которые представляют из себя структуры с полями примитивных типов или списки.

Ниже показан редактор одного сложного типа в режиме просмотра и редактирования.

Режим просмотра

[admin](#) [Components](#) [Settings](#) [Monitoring](#) English ▼

 **Repository / Type: MyType**

Create New Version

Metadata


Id:	5
Name:	MyType
Sequence:	CMYe

Fields

Name	Type
field1	string
field2	int

Редактирование

[admin](#) [Components](#) [Settings](#) [Monitoring](#) English ▼

 **Repository / Type: MyType (Uncommitted)**

Save

Metadata

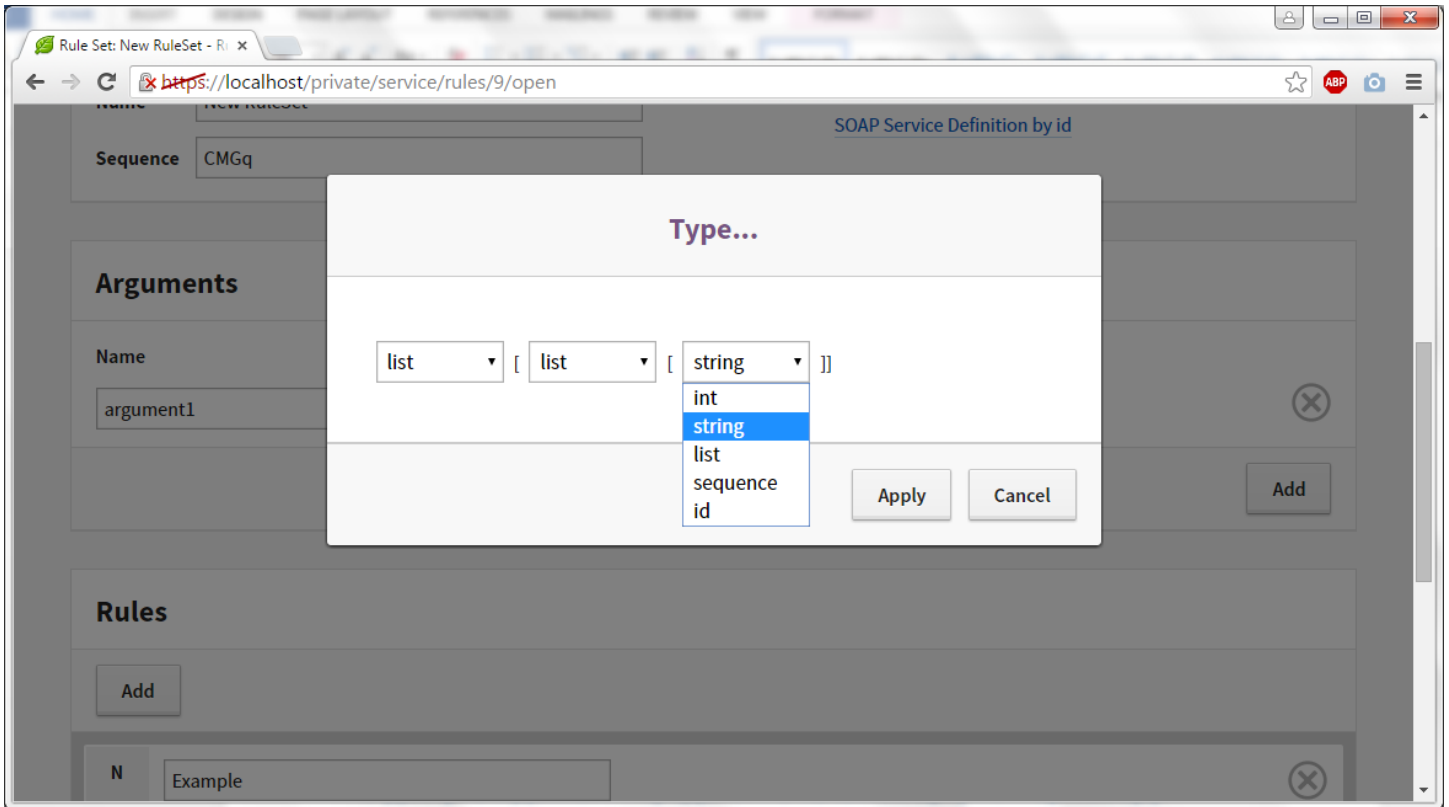
Id:	8
Name:	<input type="text" value="MyType"/>
Sequence:	<input type="text" value="CMYe"/>

Fields

Name	Type	
<input type="text" value="field1"/>	string	<input type="button" value="✕"/>
<input type="text" value="field2"/>	int	<input type="button" value="✕"/>

Add

Типы данных можно выбирать в диалоговом окне:



Аргументы наборов бизнес правил.

Сложным моментом оказалось обеспечить доступ к аргументам из языка Python, не используя сложные структуры. Аргументы доступны как обычные переменные. Например, на следующем снимке в выражении на питоне используется аргумент типа `list[list[list[int]]]` и представляет из себя вложенные списки языка.

Тип результата, это структура с полями `field1` и `field2` которая задается как ассоциативный массив.



Repository / Rule Set: New RuleSet (Uncommitted)

Save

Remove

Metadata

Id 9

Api 69146fb2998f83b89af3c31462afcc3e

Name New RuleSet

Sequence CMGq

SOAP Service Definition by sequence

SOAP Service Definition by sequence and api

[SOAP Service Definition by id](#)

Arguments

Name

Type

argument1

[list\[list\[list\[int\]\]\]](#)

Add

Results

Name

Type

result1

[sequence:CMYe](#)

Add

Rules

Add

N

Example



?

argument1[0][0][0] == 1

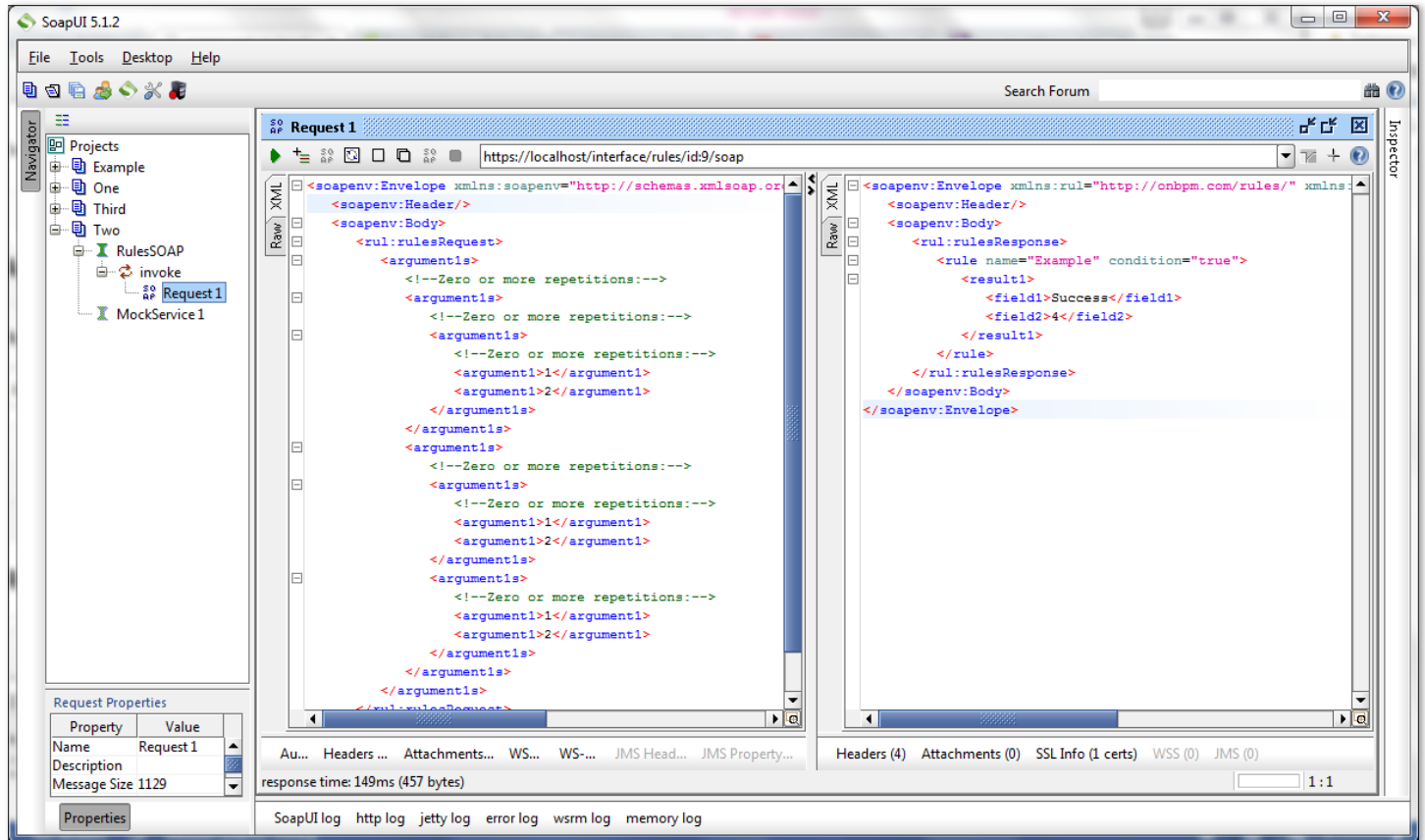
=

```
result1 = {  
  "field1": "Success",  
  "field2": 4  
}
```

Add

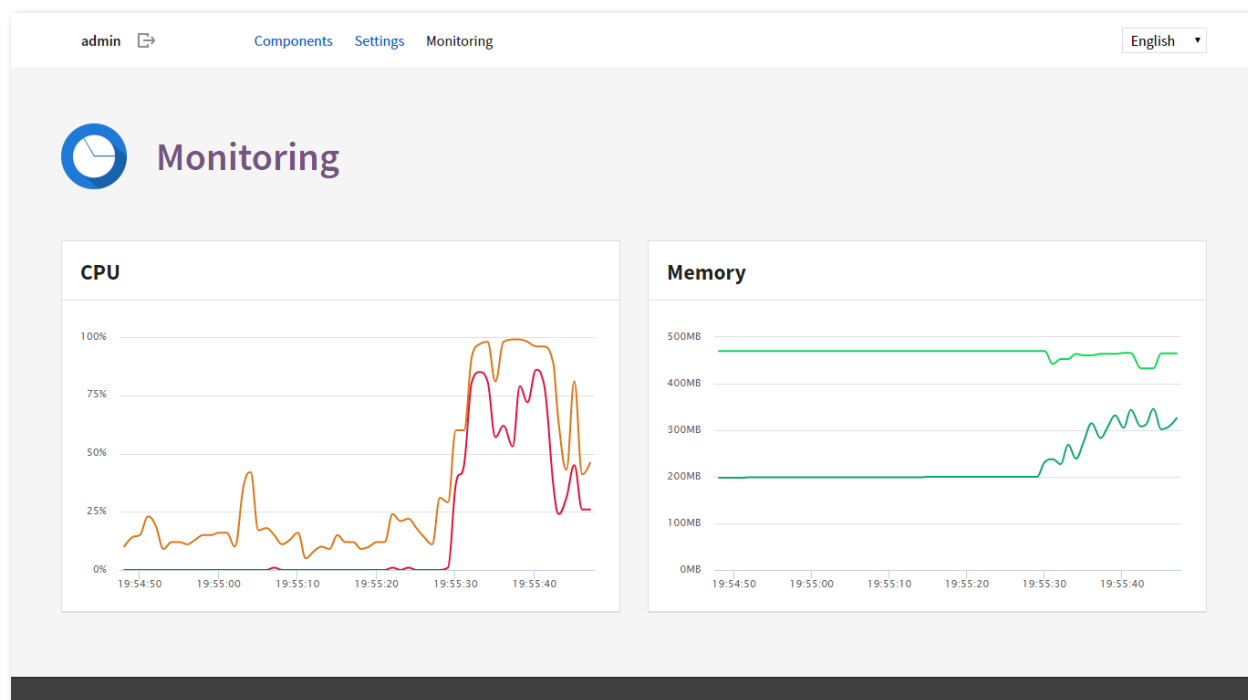
В качестве аргументов можно использовать структуры данных. Для этого, по описанию сложных типов, генерируются классы (при помощи javaassist) которые имеют геттеры соответствующие полям типов. Объекты этих классов передаются питону (Jython версии 2.7), после чего они становятся доступны в коде правила.

На следующем снимке SOAP-UI и пример вызова бизнес правил.



Мониторинг нагрузки

Система способна отображать графики загрузки процессора всей системой и виртуальной машиной, слева. Справа график использованной и доступной памяти.



Для создания графиков был использованы HighCharts (www.HighCharts.com). Информация передается через веб сокет. При открытии страницы мониторинга создается подключение. На сервере, если подключен хотя-бы один клиент, актер мониторинга активизируется и начинает каждую секунду генерировать статистику для всех подписавшихся клиентов. При отключении последнего клиента, актер мониторинга приостанавливает свою работу.

Дополнительная информация

Приложение частично переведено на русский язык. В будущем планируется заменить CoffeeScript на Scala.JS , добавить возможность использовать REST сесвисы вместо WSDL веб сервисов и сделать редактор аккаунтов пользователей.

Автор

Сергей Тарханов (sergey.tarkhanov@gmail.com), 20 Ноября 2015.