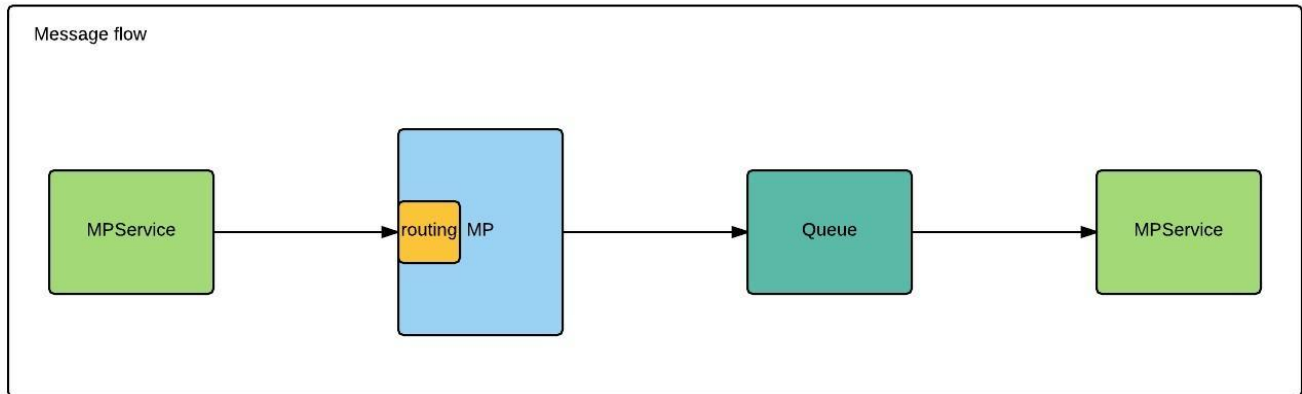


# SR\_Routing

- Wstp
- Omówienie logiki routingu
- Conditions
- Podstawowe informacje o regexach
- Bindowanie usug
- MPConsole
- Dodawanie routingu IN
- Dodawanie routingu OUT
- Testowanie routingu
- Szukanie routingów
- Wyjtki
- Bdy NoRoutingRules

## Wstp

Routing jest mechanizmem MP sycym do kierowania wiadomoci midzy usugami, a docelowymi kolejkami. Teorytycznie rozróniamy dwa rodzaje routingów IN (wyslane od operatorów do docelowych mp-service/partnerów) i OUT (kierunek od mp-service/partnera do operatora). Praktycznie, routing w kadym przypadku dziaa tak samo - kieruje wiadomo wyslan przez usug na inn usug (tak naprawd kolejk tej usugi) na podstawie danych zawartych w wiadomoci.



Kady routing konfigurowany jest za pomoc pliku XML, który po zdeployowaniu przetwarzany jest na odpowiedni obiekt wykorzystywany przez MP.

Aktualn konfiguracj routingu moemy przejrze za pomoc zapytania:

```
select * from MP.ROUTING
```

## Omówienie logiki routingu

XML routingu zawiera warunki bazujcy na zawartoci wiadomoci i ich konsekwencj która oznaczana jest poprzez mpid serwisu na który wiadomo speniajca warunki powinna trafi. Oparty jest o schemat dostpny pod adresem: <http://data.avantis.pl/mp/schema/mp-routing.xsd>

Kady XML skada si z nastpujcego nagówka:

```
<?xml version="1.0" encoding="UTF-8"?>
<routing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://data.avantis.pl/mp/schema/mp-routing.xsd">
```

Następnie wymagany elementem jest jego ID oraz krótki opis.

```
<routing-id>7055</routing-id>
  <routing-description>Routing przychodzący IN dla LA 7055</routing-description>
```

Opcjonalnym elementem który możemy dodać w "nagłówku" routingu jest także globalny warunek, który musi być spełniony aby dalsza jego część mogła być weryfikowana, w poniższym przykładzie odbiorca wiadomości musi mieć wartość 7055, aby dalsza część routingu była weryfikowana

```
<global-condition>
  <condition operator="equals" expression="recipient" value="7055" />
</global-condition>
```

Dalsza część routingu to interesuje nas routing-elementy. Składają się one z kilku kluczowych elementów.

Po pierwsze priorytet - czyli kolejno wg. której sprawdzane są warunki. Jeżeli wiadomość spełni dany warunek, jest przekierowywana na dany serwis i dalsze warunki nie są procesowane. **(Patrz wyżej!)**

Następnie w bloku conditions zawieramy logikę wg. której sprawdzamy czy wiadomość spełnia warunki.

W tagu consequence zawieramy natomiast ID usługi na którą mamy przekierować wiadomość.

```
<routing-element priority="1">
  <!-- conditions -->
  <consequence>
    <service-id>100532</service-id>
  </consequence>
</routing-element>
```

## Conditions

Warunki - czyli to co nas najbardziej interesuje. Jest to coś na kształt "ifów" wg. których sprawdzamy czy dana wiadomość zawiera interesujący nas element. Może być to LA, MSISDN, treść wiadomości, operator etc etc.

Warunki logiczne (operator condition) określamy za pomocą:

```
<xs:enumeration value="or" />
<xs:enumeration value="and" />
<xs:enumeration value="equals" />
<xs:enumeration value="pattern" />
<xs:enumeration value="not" />
<xs:enumeration value="lt" />
<xs:enumeration value="gt" />
```

Umieszczając w tagu condition jedynie znacznik operatora możemy tworzyć bloki logiczne, np:

```

<condition operator="or">
  <!-- (...) -->
</condition>

  <condition operator="and">
    <!-- (...) -->
  </condition>

```

Moemy tam take umieszczac dodatkowe dane, tak aby porównywa treści wiadomości czy te LA:

```

<condition operator="pattern" expression="sms-text" pattern="( ?i )( ?s )( ?m ) \s * B [ \. , \ ; :
] * I [ \. , \ ; : ] * C . * " />
<condition operator="equals" expression="recipient" value="7055" />
<condition expression="master-operator" operator="equals" value="26001" />

```

Atrybut condition może przyjmować wartości:

- value - oznacza wartość
- expression - oznacza który element chcemy porównywać (typ wiadomości, treść, la, msisdn, kierunek itp)
- pattern - oznacza dopasowanie wartości expression do regexach

## Przykłady

Powysze zagadnienia możemy czy tworzyć routing-elementy, np

Przekierowanie wszystkich wiadomości posiadających sendera = 72988 oraz operatora 26001 (Plus) na usługę 2182 (bramka PLUS plus-sms-72988)

```

<routing-element priority="1660">
  <condition operator="and">
    <condition expression="sender" operator="equals" value="72988" />
    <condition expression="master-operator" operator="equals" value="26001" />
  </condition>
  <consequence>
    <service-id>2182</service-id>
  </consequence>
</routing-element>

```

Przekierowanie wszystkich wiadomości zawierających treść AS.\* lub PZU.\* lub KFL.\* (wielko liter, białe znaki i ilość spacji na początku wiadomości bez znaczenia) na usługę 110111 (mps-yellowbox)

```

<routing-element priority="101">
  <condition operator="or">
    <condition operator="pattern" expression="sms-text"
pattern="( ?s )( ?m )( ?i ) \s * AS . * " />
    <condition operator="pattern" expression="sms-text"
pattern="( ?s )( ?m )( ?i ) \s * PZU . * " />
    <condition operator="pattern" expression="sms-text"
pattern="( ?s )( ?m )( ?i ) \s * KFL . * " />
  </condition>
  <consequence>
    <service-id>110111</service-id>
  </consequence>
</routing-element>

```

## Podstawowe informacje o regexach

Powysze regexy budowane s na podstawie skadni javowych regexów. Najlepsz stronk, na które moemy testowa nasze wyraenia to <http://www.regexplanet.com/advanced/java/index.html> .

Tak w skrócie, najbardziej przydatne elementy:

- (?:) - case insensitive, czyli wielko liter jest bez znaczenia
- (?:s) - oznacza e biae znaki s oznaczane jako kropka
- (?:m) - oznacza, e koniec lini i poczek s apane za pomoc ^ oraz \$
- . - oznacza wystpienie (lub nie) dowolnego znaku (tylko jednego!)
- \* oznacza dowolne wystpienie poprzedzajcego go wyraenia, np
  - \* oznacza dowoln ilo wystpie dowolnego znaku
  - [0-9]\* oznacza dowoln ilo wystpie dowolnej cyfry
- \s oznacza spacj

### Negacja

zdarza si, e "not" w pewnych sytuacjach nie zadziaa jak powinien, w takim wypadku moemy zanegowa np prefix w regexie, np:

```
<condition operator="pattern" expression="sms-text" pattern="(?! (EURO\sHELP)) .*" />
```

## Bindowanie usug

Bindowanie suy do okrelenia, która usuga moe korzysta z danego routingu. Tzn, w przypadku wysania wiadomoci przez usug, bdzie ona sprawdzaa tylko te routinge do których jest podbindowana. Jeden routing moe mie podbindowanych wiele routingów, tak samo jak i usuga moe by do wielu routingów podbindowana.

## MPConsole

Znajduje si pod adresem: [mpconsole.digitalvirgo.pl](http://mpconsole.digitalvirgo.pl). Jest to "graficzna nakadka" na JMXa. Pozwala wykonywa metody JMXowe oraz upraszcza wiele z nich.

Mpconsole posiada narzdzie to tworzenia oraz edytowania routingów w zakadce **MP Routing Management**

Home MP Statistics Delayed Delivery **MP Routing Management** Routing Changes Impact New MP-Service

Find by Routing ID  EXECUTE

Find by Service ID  EXECUTE

Find all Routings EXECUTE

DEPLOY NEW ROUTING

Moemy tutaj wyszukiwa routing za pomoc **Routing ID** oraz znale wszystkie routinge które s podbindowane do danej usugi.

Po przejciu do edycji wybranego routingu pojawi si nam okno.

[Go Back](#)

**Naley pamita, aby przy kadej zmianie podawa ID taska w ramach którego zmienia si routing**

## Dodawanie routingu IN

Dodawanie routingu IN polega na kierowaniu ruchu INoweg (od operatorów lub innych agregatorów) na inne usługi. Standardowo ruch IN rozróżniamy po LA i następnie po prefixie.

ID routingu INowego dla polskich numerów jest takie samo jak LA. U operatorów sowackich jest to np poczenie perfixu 400 i LA np 4007774.

Wicej informacji [tutaj](#)

## Dodawanie routingu OUT

Routing OUT to nic innego jak kierowanie ruchu z usług do operatorów (lub agregatorów) z naszych usług. Routing OUT powinien kierować poszczególne LA na odpowiednie bramki na podstawie operatorów zawartych w wiadomości.

## Testowanie routingu

Aby mie pewno, e routing jest prawidłowo skonfigurowany utrzymanie posiada 2 skrypty testujące oraz 2 generujące raport routingu. Wszystkie skrypty najlepiej odpala na hacie scripts z użytkownika mpdevelop. ciekawość to ~/DNIU/utils.

## Testowanie

Do testowania routingu posiadamy dwa skrypty:

**routing\_OUT\_tester.sh**

Parametry potrzebne do jego wykonania:

```
Podales za malo paramerow
1 id_testowanego_routingu
2 odbiorca (MSISDN)
3 operator
4 nadawca (LA)
np. ./routing_OUT_tester.sh 119902 48606932401 26002 7155
```

### routing\_IN\_tester.sh

Parametry potrzebne do jego wykonania:

```
1 odbiorca (LA)
2 tresc
np. ./routing_IN_tester.sh 7155 test
```

Uycie powyszch skryptów zwróci m. in. informacje o tym gdzie dana wiadomo trafia:

```
RAPORT:
26003 100405 SMS Kontakt Deutsche Bank 48519762823 7155 200 test 30245050491 14-APR-16
04.18.09.126000 PM
26003 100405 SMS Kontakt Deutsche Bank 48519762823 7155 200 test 30245051059 14-APR-16
04.18.16.637000 PM
Milego dnia :)
```

### WANE:

Przed konfiguracją nowego prefiksu robimy test czy nie jest on zajęty przez inn usług/partnera.  
Pomoe nam to wykrywa zajęte prefiksy.

## Raport

Flow każdego taska na routing (ITSO) wymaga, aby w jego komentarzach znalazły się informacje nt. zmian przeprowadzonych w routingach. Aby ułatwić sobie życie, zostały utworzone dwa automaty generujące raporty:

### routing\_IN\_raport.sh

Wymagane parametry:

```
1 odbiorca (LA)
np. ./routing_IN_raport.sh 7155
```

### routing\_OUT\_raport.sh

Wymagane parametry:

```
1 id_testowanego_routingu
np. ./routing_OUT_raport.sh 110715
```

następnie wynik całego skryptu należy wkleić do taska

## Szukanie routingów

Czasami jest potrzeba znalezienia wszystkich routingów o konkretnej specyfikacji.

### Routingi IN

Moemy to zrobić na 3 sposoby (przykład dla serwisu: 110070):

- **Dane w WIKI**

W wyszukiwark na stronie <https://wikis.services.avantis.pl> wpisujemy id usługi i wybieramy stron w formacie: [MPID-110070] mps-txtnation .

- **Skrypt na DNIU**

Lokalizacja: /usr/local/mpservices/utls/mprouting-tools/

Skrypt: search-recipient.sh

Przykład wywołania:

```
[mpdevelop@dniu /usr/local/mpservices/utls/mprouting-tools]$  
./search-recipient.sh 110070  
DEFAULT MPCORE URL=nl.mpcore.mp.avantis.pl  
***** SEARCH RESULT *****  
ROUTING-ID ==> 90946  
ROUTING-ID ==> 91225  
ROUTING-ID ==> 91255  
ROUTING-ID ==> 79003  
ROUTING-ID ==> 91955  
ROUTING-ID ==> 7136  
ROUTING-ID ==> 7555  
ROUTING-ID ==> 7925  
ROUTING-ID ==> 7936  
*****  
Exit !
```

- **SQLka**

```
select *  
from MPS.mpsml_services  
where service_id = '110070'  
;
```

### Routingi OUT

Moemy to zrobić na 2 sposoby (przykład dla serwisu: 110070):

- **Dane w WIKI**

W wyszukiwark na stronie <https://wikis.services.avantis.pl> wpisujemy id usługi i wybieramy stron w formacie: [MPID-110070] mps-txtnation .

- **SQLka**

```
select *  
from MP.mpservice_routing  
where SERVICEBEANS_SERVICE_ID = 110070  
;
```

## Wyjtki

eby ycie byo pikne istniej mpsservice które reaguj w specyficzny sposób na wiadomoci. Otó, standardowo usuga na któr wiadomo zostanie skierowana przyjmuje j i dalsze sprawdzanie wiadomoci zostaje zakoczone.

Specyficznymi przypadkami s tutaj m. in. usuga smscodes (100111), smsps (100243) oraz bramki direct Billing (PLAY - 101353, PLUS - 101403)

Kiedy routing skieruje wiadomo na tak usug sprawdza ona czy w swojej konfiguracji posiada dane do przyjcia takiej wiadomoci (np smscodes na podstawie prefixu i sufixu). Jeeli nie, to odrzuca tak wiadomo i ta leci dalej po routingu.

Powysz sytuacj obrazuje wynik testu routingu:

```
26006 200001 default-response-service 48531162757 80460 200 EROBALL 32997268122  
17-AUG-18 10.24.37.871000 AM
```

W takiej sytuacji trzeba sprawdzi czy wiadomoc wpada na usug (jej logi + logi MP). Jeeli widzimy e routing przekierowa wiadomo na prawidow usug, to warto porozmawia z DEV usugi, aby dokonfigurowa prefix na niej.

## Bdy NoRoutingRules

W przypadku, kiedy nie uda si dopasowa wiadomoci do adnej z konsekwencji wystpuje wyitek NoRoutingRules. W takiej sytacji, wiadomo zostaje zrzucana z kolejki.

**Uwaga!** Jest to nieporzdne zachowanie. Kad (**prawidow!**) wiadomo powinniśmy poprawnie przetworzy (chociaby poprzez default-response-service lub no-response-service). W przypadku gdy wykryjemy ten wyitek powinniśmy sprawdzi czy wiadomo jest poprawna oraz skonsultowa ew. z biznesem/dev czy usuga powinna wysya/odbiera takie wiadomoci.