

[DOC] Landing Pages Technical Documentation

Spis Treści

- [Spis Treści](#)
- [Wstęp](#)
- [Architektura](#)
- [Strony Landing Page](#)
 - [Typ 1: kreacje flashowe](#)
 - [Typ 2: kreacje HTML5](#)
- [Afilianci](#)
 - [Powiadamianie afilianta o wejściu użytkownika na stronę](#)
 - [Powiadamianie afilianta o zbilowaniu użytkownika w serwisie MT](#)
- [Funkcjonalności elementów systemu](#)
 - [Landing Page Service](#)
 - [Zapis i odczyt danych dotyczących użytkowników stron LP oraz ich akcji](#)
 - [Zapis i odczyt danych dotyczących konfiguracji poszczególnych stron LP](#)
 - [Obsługa powiadomienia o subskrypcji/bilowaniu użytkowników stron LP w serwisach MT z MTSP](#)
 - [Obsługa subskrypcji webowej MT za pośrednictwem MTSubscriber](#)
 - [Powiadamianie afiliantów o zbilowaniu numeru pozyskanego na LP za ich pośrednictwem](#)
 - [Zarządzanie stanem punktów \(kredytów\) użytkowników oraz historia pobierania contentu z powiązanych z LP stron WAP](#)
 - [Wysyłka wiadomości MT - SMS, WAPPush](#)
 - [Zrzucanie RBR dla wybranych serwisów MT](#)
 - [Wysyłka wiadomości e-mail do użytkowników](#)
 - [landing-page-web](#)
 - [landing-page-admin](#)
 - [Automatyczny deployment stron LP](#)
 - [Weryfikacja hostów wirtualnych](#)
- [Integracja z JMES](#)
- [Automatyczne monitorowanie stron LP](#)
 - [Notyfikacje monitorowania](#)
- [Komunikacja z aplikacjami: AVSS, EDW](#)
- [Zapytania i Baza LP dla DNIU co się z czym je](#)
- [Zalepka na zędat Landing Page](#)
 - [Teoria](#)
 - [Praktyka](#)

Wstęp

Aplikacja Landing Page służy do zarządzania i obsługi logiki i komunikacji ze stronami flash / HTML5, służy do pozyskiwania klientów do serwisów MT / MO (tzw. Landing Pages - LP).

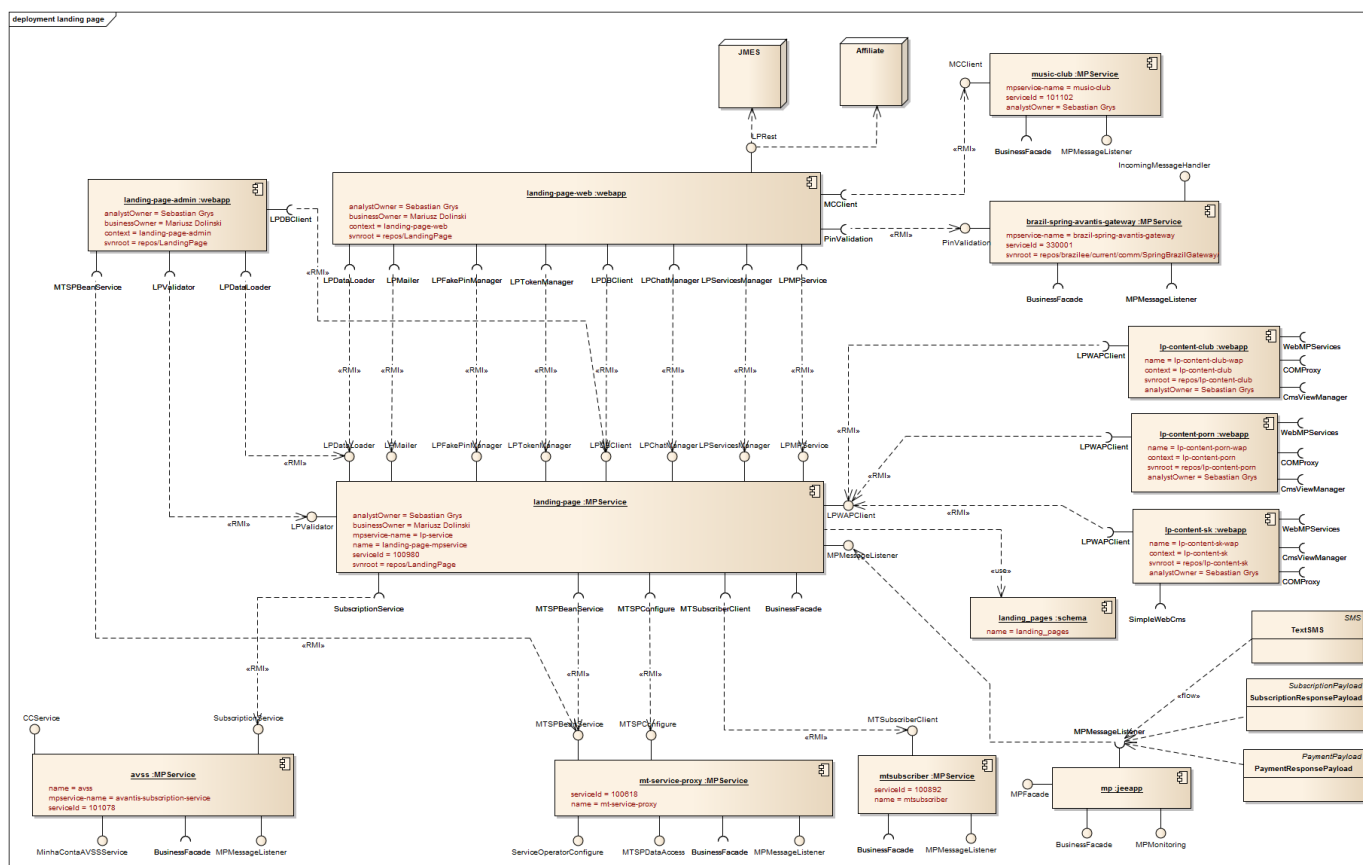
Głównym zadaniem aplikacji jest komunikacja z kreacjami flashowymi, osadzonymi w statycznych stronach HTML. Kreacje takie zazwyczaj wymagają od użytkownika wprowadzenia numeru MSISDN oraz przeprowadzenia kilku akcji (np. odpowiedzi na kilka zamkniętych pytań), by na koniec wyświetlił komendę oraz LA do zapisu do danego serwisu MT. Sam logik tych kreacji można ująć w kilku schematach, co scharakteryzowane będzie w dalszych punktach artykułu. Komunikacja taka polega w ogólności na pobieraniu z flasha danych generowanych przez akcje użytkownika, wykonywania pewnych operacji (np. zapis do bazy, generowanie wiadomości MP) i wysyłaniu odpowiedzi XML ze statusem wykonania dania oraz ew. dodatkowymi parametrami. Odpowiedzi te obsługiwane są z kolei przez flash i na ich podstawie następuje w nim przejście do kolejnego kroku, np. zmiana widoku, wyświetlenie komunikatu o bledzie.

Przykładowe strony LP:

- [Klub Lidera](#)
- [Czy będziesz bogaty za 10 lat](#)
- [Nimfomanka](#)

Dodatkowo, możliwe jest również zarządzanie (dodawanie/usuwanie/edycja) poszczególnymi 'instancjami' stron LP oraz ich konfiguracjami przechowywanymi w bazie, czyli możliwe wygenerowanie i wystawienie pod publicznym adresem URL nowej statycznej strony HTML z osadzonym flashem, gotowej do komunikacji z główną logiką systemu - wszystko z poziomu interfejsu użytkownika.

Architektura



Zasadniczo pod pojęciem strony LP rozumiemy stron HTML, w której osadzona jest kreacja flash, komunikująca się z LPS za pośrednictwem `landing-page-web`. W praktyce stosowane są strony JSP, wraz z zestawem skryptów JavaScript[?], służących do osadzania flasha. Strona jest w technologii JSP, ponieważ flash musi przekazać do reszty systemu poza parametrami generowanymi w obrębie kreacji (a więc np. numer MSISDN, token) również parametry zewnętrzne względem flasha, a więc na przykład numer IP odwiedzającego (w celu zastosowania filtrów antyspamowych) lub pewnych parametrów z linku wywołującego stronę. W tym celu stosowane są `flashvars`, zmienne, których wartości określane są w kodzie strony HTML i które przekazywane są następnie do flasha. Aby zmienne były generowane dynamicznie, również strona HTML musi być generowana dynamicznie - std JSP. Strony generowane i wystawiane są przez aplikację `landing-page-admin` (patrz: [Automatyczny deployment stron LP](#))

Typ 1: kreacje flashowe

Typ 2: kreacje HTML5

Zarówno kreacje flash jak i HTML5 korzystaj z metod landing-page-web wystawionych na interfejsach RESTowych. Do głównych API nale:

- LPMusicClubRest
- LPXmlConfigurationRest

Dokumentacja metod jest w dokumencie:

- [LPRestAPI.pdf](#)

Afilianci

Ruch na stronach LP generowany jest przez afiliantów, czyli firmy, które zajmują się reklamą i prowadzą w sieci kampanie promocyjne poszczególnych LP i na podstawie tego ruchu, a przede wszystkim ilości osób zapisanych do MT za pośrednictwem LP następuje rozliczenie z Avantis.

W celu umożliwienia takiego rozliczenia stosowane są następujące funkcjonalności:

- Powiadamianie afilianta o wejściu użytkownika na stronę
- Powiadamianie afilianta o zbilowaniu użytkownika w serwisie MT

Powiadomianie afilianta o wejściu użytkownika na stronę

W tym celu afiliant dostarcza statyczny kod, który przechowywany jest w bazie w polu `AFFILIATE_CODE` tabeli `LP_CONFIGS` konfiguracji LP. Kod ten wklejany jest w kod strony w trakcie deploymentu LP. Jest to opcja wykluczająca się wzajemnie z opcją poniżej i o jej uyciu decyduje flaga `DISPLAY_CODE` w tej samej tabeli.

Powiadomianie afilianta o zbilowaniu użytkownika w serwisie MT

W tym przypadku afiliant dostarcza adres URL, pod który oczekuje puknięcia `http(s)` GET z odpowiednimi parametrami. Adres najczęściej jest unikalny dla LP. W zależności od afilianta, oczekuje on parametrów generowanych przez aplikację oraz/lub parametrów przekazanych w linku kierującym użytkownika na LP. Wartości tych ostatnich przechowywane są w polu `AFFILIATE_PARAM_VALUES` tabeli `LP_ORDERS` i przechwytywane na podstawie nazw parametrów określonych w polu `PARAMS` tabeli `LP_AFFILIATES` dla danego afilianta. Mechanizm przechwytywania i przekazywania parametrów opisany jest [tutaj](#).

Co do parametrów generowanych dynamicznie, obecnie wyróżniamy 2 typy: unikalny dla użytkownika (`customerID`) i unikalny dla akcji (`orderID`). Ten pierwszy generowany jest funkcją hashując numer `MSISDN` (jako String), za drugi `MSISDN` i datę (jako Stringi oddzielone znakiem '_'). Uycie ich w tworzonej URL określa flagi `USE_CUSTOMER_ID` oraz `USE_ORDER_ID` w tabeli `LP_AFFILIATES`. Sam URL przechowywany jest w polu `AFFILIATE_URL` tabeli `LP_CONFIGS` w formacie `MessageFormat?`, przykładowo: `http://al.s1.system3.pl/1/?s3fp={1}&webID={2}&AdvertiserId=122&OrderId={0}&TotalPrice=02`. Jeżeli mają być użyte `customerID` i `orderID`, będą wstawione w parametry `{0}` i `{1}`, dalej umieszczane będą parametry z `AFFILIATE_PARAM_VALUES`.

Kada próba powiadomienia afilianta odnotowywana jest w tabeli `LP_AFF_NOTIFICATIONS`. Rekord w tej tabeli określa, z jakim `msisdn` i stroną LP jest związane powiadomienie, pod jaki adres wychodziło puknięcie i jak się zakończyło (status HTTP lub wartość `ERROR`). W przypadku powiadomienia zakończonych będem, a więc zawieszanych na krócej co 10 minut wiadomości MP, rekord taki zawiera również liczbę nieudanych powiadomień. Rekordy związane są z rekordami z `LP_ORDERS` poprzez `LP_AFF_NOTIFICATIONS.ORDER_ID = LP_ORDERS.ID`.

Czyli jeżeli chcemy znaleźć rekord dla `msisdn 601001001` wykonujemy zapytanie:

```
select
*
from landing_pages.lp_aff_notifications lpa
join landing_pages.lp_orders lpo on lpa.order_id = lpo.id
where lpo.msisdn like '%601001001%'
```

Funkcjonalności elementów systemu

Landing Page Service

Główną częścią systemu - aplikacja standalone, zawierająca `MPService` o service ID `100980` i nazwie `LandingPage`. Aplikacja uruchomiona jest na serwerze `mike.avantis.pl` i loguje do `/usr/local/logs/mpservices/services/landing-page.log` ze standardową rotacją dzienników. Udostępnia po `RMI` interfejs `LPDBClient` pod adresem `rmi://nl.lp.avantis.pl:5678/LPDBClient`

LPS zawiera większość logiki biznesowej, odpowiada za przetwarzanie danych i kontakt z bazą danych oraz afiliantami i usługami powiązanymi (`MTSP` i `MTSubscriber`).

Funkcjonalności

- Zapis i odczyt danych dotyczących użytkowników stron LP oraz ich akcji
- Zapis i odczyt danych dotyczących konfiguracji poszczególnych stron LP
- Obsługa powiadomienia o subskrypcji/bilowaniu użytkowników stron LP w serwisach MT z MTSP
- Obsługa subskrypcji internetowej MT za pośrednictwem MTSubscriber
- Powiadomianie afiliantów o zbilowaniu numeru pozyskanego na LP za ich pośrednictwem
- Zarządzanie stanem punktów (kredytów) użytkowników oraz historia pobierania contentu z powiązanych z LP stron WAP
- Wysyłka wiadomości MT - SMS, WAPPush
- Zrzucanie RBR dla wybranych serwisów MT
- Wysyłka wiadomości e-mail do użytkowników

Zapis i odczyt danych dotyczących użytkowników stron LP oraz ich akcji

W celu zapewnienia wszystkich pozostałych funkcjonalności, a także w celach raportowych, gromadzone są dane o wszystkich użytkownikach, którzy wyrazili chęć zapisu do serwisu MT poprzez LP. Dane trzymane są w bazie na kluczu Oracle w schemacie `LANDING_PAGES`. Podstawową tabelą, gromadzącą informacje o tym, kto (msisdn) wykonał akcję zapisu na jakim LP i jaki jest jego obecny status w danym serwisie MT, jest tabela `LP_ORDER`. Rekord jest unikalny dla trzech wartości `MSISDN`, `SUBSCRIPTION_ID` oraz `CONFIG_ID` wskazujących na rekord z tabeli `LP_CONFIGS` (o tym *niej*), a więc jeden rekord stanowi odzwierciedlenie akcji danego użytkownika na danym LP.

Dodatkowo, akcje istotne z punktu widzenia głównej funkcjonalności aplikacji są osobno logowane do tabeli `LP_EVENTS`. Logowany jest `msisdn` i ID konfiguracji LP, a także data i typ zdarzenia.

Wyróżniamy następujące typy zdarzeń:

- `GET_OPERATOR` - pobranie operatora dla `msisdn`
- `GET_SUBSCRIPTIONS` - pobranie listy subskrypcji MT dla `msisdn`
- `SAVE_MSISDN` - zapis nowego rekordu w `LP_ORDERS`
- `SEND_FAKE_PIN` - wysyłka fake pin
- `SUBS_REQUEST` - danie rozpoczęcia subskrypcji internetowej
- `SUBS_CONFIRMATION` - potwierdzenie rozpoczęcia subskrypcji
- `BILLED` - zbilowanie
- `UNSUBSCRIPTION` - wypisanie z subskrypcji
- `PIN_VERIFY` - danie weryfikacji pinu do websubskrypcji
- `PIN_OK` - poprawna weryfikacja pinu do websubskrypcji
- `PIN_NOK` - niepoprawna weryfikacja pinu do websubskrypcji
- `AFF_NOTIFICATION` - powiadomienie afilianta

Jeżeli akcja wywołana jest za sprawą LP JMES, odpowiednik tej informacji zapisywany jest również do bazy JMES

Zapis i odczyt danych dotyczących konfiguracji poszczególnych stron LP

Ponieważ poszczególne strony LP mogą być przypisane do różnych afiliantów, różnych serwisów MT, a także zawierać różne kreacje flash, niezbędne jest przechowywanie konfiguracji każdej instancji uruchomionego LP w bazie danych. Ma to miejsce w tabeli `LP_CONFIGS`. Konfiguracje te są edytowane za pomocą panelu administracyjnego w aplikacji internetowej `landing-page-admin`.

Obsługa powiadomienia o subskrypcji/bilowaniu użytkowników stron LP w serwisach MT z MTSP

Aby baza danych zawierała aktualne informacje na temat statusu użytkownika w serwisie MT, a także aby powiadamiać afiliantów, niezbędne jest otrzymywanie powiadomienia o każdej zmianie statusu `msisdn` w serwisie MT. Powiadomienia takie wysyłane są z MTSP za pomocą wiadomości MP i odbierane przez `mpservice LandingPage`. O tym, czy powiadomienia o akcjach w danym serwisie MT będą wysyłane do `mpservice LandingPage` decyduje konfiguracja danego serwisu w MTSP.

Przykład:

Powiadomienie o aktywacji subskrypcji

```
2014-06-17 14:53:08,507 | INFO | [26976937734][120058] |
processor.notifier.CommandSendingSubscriptionNotifier | Subscriber received:
SubscriberDTO{msisdn='48506986496', updateDate=Wed Jun 17 14:53:08 CEST 2014, status=10, operator=26006,
serviceName='PREMIUM_60355', parameter='', source='null', subscriberId='453211613',
promoSessionId='4195604476'} for orderId: 89 850 460
2014-06-17 14:53:08,507 | DEBUG | [26976937734][120058] |
processor.notifier.CommandSendingSubscriptionNotifier | Minutes difference between creation and
subscription date: 0 for order with id: 89 850 460
2014-06-17 14:53:08,507 | INFO | [26976937734][120058] |
processor.notifier.CommandSendingSubscriptionNotifier | Subscribe: Order{id=89850460, sessionId=4195604476,
msisdn=48506986496, operator=POLAND_P4, ip=109.95.93.141, referer=null, creationDate=Wed Jun 17 14:52:33
CEST 2014, updateDate=Wed Jun 17 14:52:33 CEST 2014, subscriptionDate=null, unsubscriptionDate=null,
firstBilled=null, lastBilled=null, billCount=0, regChannel=WWW, affiliateParamValues=3CC6542933D1161,
lpParamValues=null, orderId=7917942, configId=194068, serviceId=10050, subscriptionId=PL_67419275,
token=89966, campaign=null, provider=null, channel=null, deliveryStatus=UNKNOWN, rbrProvisionDate=null,
protocolId=null}
```

Powiadomienie o bilowaniu

```
2014-06-17 14:58:13,352 | DEBUG | [26976967801][120036] | processor.notifier.CommandSendingBillingNotifier
| About to invoke
processMsisdnPack(BilledMsisdnPackDTO)status=[20]subscriptionId=[0][26006][PREMIUM_60355][Tue Jun 17
14:58:12 CEST 2014][[BilledMsisdnDTO[48506986496][169650347957]]], 26976967801)
2014-06-17 14:58:13,352 | INFO | [26976967801][120036] | processor.notifier.CommandSendingBillingNotifier |
Processing billed msisdn pack: BilledMsisdnPackDTO]status=[20]subscriptionId=[0][26006][PREMIUM_60355][Tue
Jun 17 14:58:12 CEST 2014][[BilledMsisdnDTO[48506986496][169650347957]]]
2014-06-17 14:58:13,352 | INFO | [26976967801][120036] | processor.notifier.CommandSendingBillingNotifier |
Processing msisdn: BilledMsisdnDTO[48506986496][169650347957]
2014-06-17 14:58:13,352 | INFO | [26976967801][120036] | processor.notifier.CommandSendingBillingNotifier |
Billing date: 17.06.14 14:58 for msisdn: 48506986496, service: PREMIUM_60355 and operator: POLAND_P4
```

Powiadomienie o wypisaniu z subskrypcji

```
2014-06-17 15:04:09,689 | INFO | [26977004897][120058] |
processor.notifier.CommandSendingSubscriptionNotifier | Subscriber received:
SubscriberDTO{msisdn='48690927667', updateDate=Tue Jun 17 15:04:08 CEST
2014, status=2, operator=26006, serviceName='PREMIUM_60355', parameter='null', source='null',
subscriberId='453207934', promoSessionId='null'} for orderId: 89 847 484
2014-06-17 15:04:09,689 | INFO | [26977004897][120058] |
processor.notifier.CommandSendingSubscriptionNotifier | Unsubscribe: Order{id=89847484,
sessionId=4195085379, msisdn=48690927667, operator=POLAND_P4,
ip=83.27.199.213, referer=null, creationDate=Mon Jun 16 18:18:49 CEST 2014, updateDate=Mon Jun 16 18:19:15
CEST 2014, subscriptionDate=Mon Jun 16 18:19:14 CEST 2014, unsubscriptionDate=null, firstBilled=null
, lastBilled=null, billCount=0, regChannel=WWW, affiliateParamValues=10283271992136701140616,
lpParamValues=null, orderId=204483555, configId=67062, serviceId=10050, subscriptionId=PL_170872728,
token=21866,
campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=null,
protocolId=null}
2014-06-17 15:04:09,689 | DEBUG | [26977004897][120058] | orm.jpa.GenericJPA | object Order{id=89847484,
sessionId=4195085379, msisdn=48690927667, operator=POLAND_P4, ip=83.27.199.213,
referer=null, creationDate=Mon Jun 16 18:18:49 CEST 2014, updateDate=Tue Jun 17 15:04:08 CEST 2014,
subscriptionDate=Mon Jun 16 18:19:14 CEST 2014, unsubscriptionDate=Tue Jun 17 15:04:08 CEST 2014, firstBill
ed=null, lastBilled=null, billCount=0, regChannel=WWW, affiliateParamValues=10283271992136701140616,
lpParamValues=null, orderId=204483555, configId=67062, serviceId=10050, subscriptionId=PL_170872728, token
=21866, campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=null,
protocolId=null} updated.
```

Obsuga subskrypcji webowej MT za pośrednictwem MTSubscriber

Poniewa strony LP wykorzystują mechanizm subskrypcji webowej do zapisu użytkownika do serwisu MT, LPS komunikuje się z usług MTSubscriber za pomocą wystawionego przez interfejsu [RMI](#) MTSubscriberClient? .

Zasadniczo proces subskrypcji webowej wygląda następująco:

1. Użytkownik wprowadza MSISDN na stronie LP
2. Kreacja puka po http do landing-page-web przekazując MSISDN i inne parametry
3. landing-page-web puka po [RMI](#) przekazując dane do LPS
4. LPS puka po RMI do MTSubscriber, w celu wygenerowania unikalnego kodu (web token)
5. MTSubscriber powiadamia operatora o chci rozpoczęcia subskrypcji MT przez dany MSISDN
6. Operator wysyła SMS z wygenerowanym tokenem do MSISDN
7. Użytkownik wpisuje otrzymany token na stronie LP
8. Kreacja puka po http do landing-page-web przekazując MSISDN, wpisany token i inne parametry
9. landing-page-web puka po [RMI](#) przekazując dane do LPS
10. LPS puka po RMI do MTSubscriber, w celu weryfikacji wprowadzonego tokena
11. MTSubscriber weryfikuje token u operatora i zwraca status weryfikacji

Jeżeli wprowadzony token jest prawidłowy, użytkownik rozpoczyna subskrypcję MT. Jeżeli nie, konieczne jest powtórzenie całej akcji.

Przykład:

danie rozpoczęcia websubskrypcji (generowanie tokenu)

MTSUBSCRIBER:

```
2014-06-17 14:52:34,639 | INFO | ManagerMTSubscriberClient | [48506986496][PREMIUM_60355] | order token for
MTSubscriberRequest{ip='109.95.93.141', msisdn='48506986496', partnerName='DigitalAlea_L
P_Polska', serviceName='PREMIUM_60355', operatorCode=26006, requestType=ORDER_TOKEN, token='null',
transactionId='null', completeRegistrationUrl='null', failRegistrationUrl='null',
cancelRegistrationUrl='null', ackMpServiceId=120036, promoSessionId=null, codeRegExp='null'}

2014-06-17 14:52:34,671 | INFO | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | request to
play: RegistrationServiceRequest{msisdn='48506986496', serviceName='PREMIUM', largeAccount='6
0355', pin='null', transactionId='null'}

2014-06-17 14:52:34,861 | INFO | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | response from
play: RegistrationServiceResponse{resultCode='200'{transactionId='0866c2ef4a044e99a3fcfae7
7cdc9b1a'}}

2014-06-17 14:52:34,866 | INFO | ManagerMTSubscriberClient | [48506986496][PREMIUM_60355] | order token
response: MTSubscriberResponse{mtResponseStatus=TOKEN_SENT, transactionId='0866c2ef4a044e99a
3fcfae77cdc9b1a', token='null', authChannel='null', authURL='null'}
```

LPS:

```
2014-06-17 14:52:34,866 | INFO | RMI TCP Connection(61982)-192.168.2.101 | core.impl.LPServicesManagerImpl
| MTSubscriber response received: [TOKEN_SENT,0866c2ef4a044e99a3fcfae77cdc9b1a] for request:
MTSubscriberRequest{ip='109.95.93.141', msisdn='48506986496', partnerName='DigitalAlea_LP_Polska',
serviceName='PREMIUM_60355', operatorCode=26006, requestType=ORDER_TOKEN, token='null',
transactionId='null', completeRegistrationUrl='null', failRegistrationUrl='null',
cancelRegistrationUrl='null', ackMpServiceId=120036, promoSessionId=null, codeRegExp='null'}
```

danie potwierdzenia websubskrypcji (weryfikacja tokenu)

MTSUBSCRIBER:

```

2014-06-17 14:53:08,107 | INFO | ManagerMTSubscriberClient | [48506986496][PREMIUM_60355] | activate
subscription for MTSubscriberRequest{ip='109.95.93.141', msisdn='48506986496',
partnerName='DigitalAlea_LP_Polska', serviceName='PREMIUM_60355', operatorCode=26006, requestType=ACTIVATE,
token='89966', transactionId='0866c2ef4a044e99a3fcfae77cdc9b1a', completeRegistrationUrl='null',
failRegistrationUrl='null', cancelRegistrationUrl='null', ackMpServiceId=null, promoSessionId=null,
codeRegExp='null'}
2014-06-17 14:53:08,107 | DEBUG | PoolableMTSubscriberClient | [48506986496][PREMIUM_60355] | invocation
counter for operator: 26006 before invoke(): 0 , global: 1
2014-06-17 14:53:08,107 | DEBUG | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | ACTIVATE
2014-06-17 14:53:08,107 | DEBUG | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | matching
token '89966' by regexp: [0-9]{4,8}
2014-06-17 14:53:08,107 | DEBUG | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | matching
msisdn by regexp: 48[0-9]{9}
2014-06-17 14:53:08,107 | DEBUG | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | IP:
/109.95.93.141
2014-06-17 14:53:08,112 | DEBUG | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] |
countRequestByMsisdn: 0 after1 date: 17.06.14 13:53
2014-06-17 14:53:08,112 | INFO | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | from mtsp
cache: PlayServiceConfiguration{operatorPrefix=26006, playServiceId='X2S30',
playServiceRegistrationId='F8NBJ', playServiceUnregistrationId='3S607',
playServiceActiveRegistrationId='Q3NO0', provider='2', gatewayServiceName='PREMIUM',
contentCode='Premium_60355', rbssProductId=256119, rbssProductIdReg=269087, rbssChannelId=90548,
rbssLA='60355', billCode='null', operatorLA='60355', wapPushLA='null', cronSendDate='0 0 14 ? *
TUE,THU,SAT', cronAdditionalSendDate='null', wapPushCronSendDate='null', contentUpdatable='true',
importAvContent='true', active='true', welcomeMessageType='BILL_LAST_SUBSCRIPTION', tryAndBuyHours='0',
subscriptionInterval='0', mpServiceGateway='100609', serviceOperatorType='STANDARD', singleBillPrice='3',
currency='PLN', subscriptionDescription='null', monitored='1', contentSenderLA='null',
billRetryInterval='null', billRetryCron='null', customPrice='null', registrationMessageText='null',
unregistrationMessageText='null', failedBillingsLimit=null, alerts=null, registrationMessagesText=[],
welcomeMessageStatus=WelcomeMessageStatusDTO{active=true, type=BILL_LAST_SUBSCRIPTION,
updateDate=2014-06-14 14:03:48.073, operatorCode=26006, statusCode=3, contents=[]}, crmProductId=CRM-614}}
date: 17.06.14 13:55
2014-06-17 14:53:08,112 | INFO | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | request to
play: RegistrationServiceRequest{msisdn='48506986496', serviceName='PREMIUM', largeAccount='60355',
pin='89966', transactionId='0866c2ef4a044e99a3fcfae77cdc9b1a'}
2014-06-17 14:53:08,221 | INFO | PlayMTSubscriberClientImpl | [48506986496][PREMIUM_60355] | response from
play: RegistrationServiceResponse{resultCode='200',{transactionId='0866c2ef4a044e99a3fcfae77cdc9b1a'}}
2014-06-17 14:53:08,222 | DEBUG | GenericJPA | [48506986496][PREMIUM_60355] | object
RequestLog(230 694 221): {ip = "109.95.93.141", token = "89966", msisdn = "48506986496", operatorId =
26006, transactionId = "0866c2ef4a044e99a3fcfae77cdc9b1a", createDate = 2014-06-17 14:53:08, requestType =
ACTIVATE, partnerName = "DigitalAlea_LP_Polska", serviceName = "PREMIUM_60355", completeRegistrationUrl =
null, cancelRegistrationUrl = null, failRegistrationUrl = null, promoSessionId = null, ackMpServiceId =
null} persisted.
2014-06-17 14:53:08,229 | DEBUG | PoolableMTSubscriberClient | [48506986496][PREMIUM_60355] | invocation
counter for operator: 26006 after invoke(): 0 , global: 0
2014-06-17 14:53:08,229 | INFO | ManagerMTSubscriberClient | [48506986496][PREMIUM_60355] | activate
subscription response: MTSubscriberResponse{mtResponseStatus=SUBSCRIPTION_ACTIVATED,
transactionId='0866c2ef4a044e99a3fcfae77cdc9b1a', token='null', authChannel='null', authURL='null'}

```

LPS:

```

2014-06-17 14:53:08,230 | INFO | RMI TCP Connection(61963)-192.168.2.102 |
core.impl.LPServiceManagerUnsynchImpl | MTSubscriber response received:
[SUBSCRIPTION_ACTIVATED,0866c2ef4a044e99a3fcfae77cdc9b1a
] for request: MTSubscriberRequest{ip='109.95.93.141', msisdn='48506986496',
partnerName='DigitalAlea_LP_Polska', serviceName='PREMIUM_60355', operatorCode=26006, requestType=ACTIVATE,
token='89966', transac
tionId='0866c2ef4a044e99a3fcfae77cdc9b1a', completeRegistrationUrl='null', failRegistrationUrl='null',
cancelRegistrationUrl='null', ackMpServiceId=null, promoSessionId=null, codeRegExp='null'}
2014-06-17 14:53:08,230 | DEBUG | RMI TCP Connection(61963)-192.168.2.102 | orm.jpa.GenericJPA | object
Order{id=89850460, sessionId=4195604476, msisdn=48506986496, operator=POLAND_P4,
ip=109.95.93.141, referer=null, creationDate=2014-06-17 14:52:33 CEST 2014, updateDate=2014-06-17 14:52:33
CEST 2014, subscriptionDate=null, unsubscriptionDate=null, firstBilled=null, lastBilled=null, billCo
unt=0, regChannel=WWW, affiliateParamValues=3CC6542933D1161, lpParamValues=null, orderId=7917942,
configId=194068, serviceId=10050, subscriptionId=PL_67419275, token=89966, campaign=null, provider=null,
chan
nel=null, deliveryStatus=UNKNOWN, rbrProvisionDate=null, protocolId=null} updated.

```


Powiadamianie afiliantów o zbilowaniu numeru pozyskanego na LP za ich pośrednictwem

W celu prowadzenia rozlicze z ruchu generowanego na stronach przez afiliantów, wymagany krok jest powiadamianie afiliantów o poprawnym pozyskaniu użytkownika w serwisie MT za pomocą strony LP, na którą skierowany został przez rodki promocji danej kampanii (np. banner, link). Za pozyskanie takie najczęściej uważa się moment pierwszego zbilowania MSISDN użytkownika w serwisie MT. Inne opcje powiadamiania to: po każdym zbilowaniu, po rejestracji oraz po wejściu na stronę. W tym ostatnim przypadku afiliant dostarcza statyczny kod, który wklejany jest w kod strony HTML z zagniecionym flashem. Kod taki przechowywany jest w polu `AFFILIATE_CODE` w tabeli `LP_CONFIGS`.

W przypadku powiadomienia po zbilowaniu/zbilowaniach proces wygląda następująco:

1. MTSP przysyła do `LandingPage` wiadomość MP z informacją o zbilowaniu danego MSISDN w danym serwisie MT
2. LPS sprawdza, czy w tabeli `LP_ORDERS` istnieją rekordy dla danego MSISDN i serwisu MT.
3. Jeśli rekordy istnieją, to następuje ich update - wartości `STATUS`, `FIRST_BILLED` w przypadku pierwszego bilowania i `LAST_BILLED` w przypadku każdego kolejnego, `BILL_COUNT`
4. Następuje dodanie kredytów do konta użytkownika (o tym [niej](#))
5. Następuje wygenerowanie obiektu powiadomienia i wysłanie go wiadomości MP `INTERNAL` - czyli usługa wysyła wiadomość sama do siebie. Ma to na celu zapewnienie, że jeśli z jakich przyczyn powiadomienie nie udało się w danym momencie (błąd sieci, przyczyny zewnętrzne), wyjdzie ono później. W przypadku nieudanego powiadomienia wiadomość opóźniana jest o 1 minutę.
6. `LandingPage` odbiera wiadomość, odczytuje z niej informacje niezbędne do powiadomienia i wykonuje puknięcie http pod adres podany przez afilianta. Adres ten przechowywany jest w polu `AFFILIATE_URL` tabeli `LP_CONFIGS`

Przykład:

```
2014-06-17 15:35:55,506 | DEBUG | [26977213596][120046] | mp.executor.NotifyAffiliateExecutor | About to
invoke execute(NotifyAffiliateCommand{orderId=89678270, billCount=1, priceNet=null,
sendRbrCommand=SendRbrCommand{msisdn=48511357552, operator=POLAND_ORANGE, transactionDate=Tue Jun 17
15:35:54 CEST 2014, transactionId=26977213588, orderId=89678270, serviceName=PREMIUM_60355,
defaultRBR=false, billCode=null, affiliateNotificationId=null, priceNet=null}})

2014-06-17 15:35:55,522 | DEBUG | [26977213596][120046] | orm.jpa.GenericJPA | found 1
AffiliateConfiguration(s)
2014-06-17 15:35:55,522 | DEBUG | [26977213596][120046] | mp.executor.NotifyAffiliateExecutor |
AffiliateConfiguration: AffiliateConfiguration{affiliateId=11050, serviceName=PREMIUM_60355,
saveDate=2013-09-27 16:03:08.254, channelFromAffiliateBu=true, channelId=102392,
productFromAffiliateBu=true, productId=256119, laFromAffiliateBu=true, la=LP0_6ZL,
notificationType=FIRST_BILLING, notificationLimit=100, billCode=null, registrations=0,
registrationNotifications=0}, msisdn: 48511357552, order billCount: 1, command billCount: 1, displayCode:
false, isOrderJMESbyId: false, priceNet: null

2014-06-17 15:35:55,674 | DEBUG | [26977213596][120046] | mp.executor.NotifyAffiliateExecutor | Existing
notification: null
2014-06-17 15:35:55,674 | INFO | [26977213596][120046] | mp.executor.NotifyAffiliateExecutor | Checking
conditions for order: Order{id=89678270, sessionId=4188114257, msisdn=48511357552, operator=POLAND_ORANGE,
ip=46.113.145.65, referer=null, creationDate=Mon May 26 15:01:17 CEST 2014, updateDate=Tue Jun 17 15:35:55
CEST 2014, subscriptionDate=Mon May 26 15:03:10 CEST 2014, unsubscribeDate=null, firstBilled=Tue Jun 17
15:35:55 CEST 2014, lastBilled=null, billCount=1, regChannel=WWW,
affiliateParamValues=102461131456536036140526, lpParamValues=null, orderId=1205758282, configId=217057,
serviceId=10050, subscriptionId=PL_HpHnC0QS3AlsCnUouYkYUrlPOXyTmOy2Ka9XhPoTn192SUqXcx, token=93781,
campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=null,
protocolId=null}
2014-06-17 15:35:55,674 | INFO | [26977213596][120046] | mp.executor.NotifyAffiliateExecutor | Adding
affiliate notification to queue.
2014-06-17 15:35:55,674 | DEBUG | [26977213596][120046] | core.affiliate.AffiliateNotifierProxy | About to
invoke notifyAffiliate(Notification{orderId=89678270, configId=217057, msisdn=48511357552,
affiliateName=SponsorPay, affiliateParamValues=null, sendRbrCommand=SendRbrCommand{msisdn=48511357552,
operator=POLAND_ORANGE, transactionDate=Tue Jun 17 15:35:54 CEST 2014, transactionId=26977213588,
orderId=89678270, serviceName=PREMIUM_60355, defaultRBR=false, billCode=null, affiliateNotificationId=null,
priceNet=null}})

2014-06-17 15:35:55,674 | INFO | [26977213596][120046] | core.affiliate.HttpAffiliateNotifier | Notifying
affiliate with url: http://www2.balao.de/SPE8a?transaction_id=102461131456536036140526
2014-06-17 15:35:55,674 | INFO | [26977213596][120046] | core.affiliate.HttpAffiliateNotifier |
NotificationTimeoutChecker result: CheckerReturn{timeout=false, millisSinceRegistration=1902764859,
millisTimeout=null} for notification: Notification{orderId=89678270, configId=217057, msisdn=48511357552,
affiliateName=SponsorPay, affiliateParamValues=null, sendRbrCommand=SendRbrCommand{msisdn=48511357552,
operator=POLAND_ORANGE, transactionDate=Tue Jun 17 15:35:54 CEST 2014, transactionId=26977213588,
orderId=89678270, serviceName=PREMIUM_60355, defaultRBR=false, billCode=null, affiliateNotificationId=null,
priceNet=null}}
```



```
2014-06-17 15:35:56,052 | INFO | [26977213596][120046] | core.affiliate.HttpAffiliateNotifier | HttpStatus: HTTP/1.1 200 OK, response contents: success=true;
2014-06-17 15:35:56,053 | DEBUG | [26977213596][120046] | core.affiliate.HttpAffiliateNotifier | Received http status: 200 with response body: success=true;
2014-06-17 15:35:56,053 | INFO | [26977213596][120046] | core.affiliate.HttpAffiliateNotifier | Saving LP affiliate notification for Order(89 678 270), msisdn: 48511357552, URL: http://www2.balao.de/SPE8a?transact ion_id=102461131456536036140526, HttpStatus: 200
```

Zarządzanie stanem punktów (kredytów) użytkowników oraz historia pobierania contentu z powiązanych z LP stron WAP

Z kampaniami prowadzonymi za pomocą LP związane są również serwisy WAPowe z pobieralnym contentem. Użytkownik zapisany do MT przez LP po każdym zbilowaniu otrzymuje pewną ilość punktów, zalen od LA, na jakim działa dany serwis MT. Punkty te może następnie wydawać na pobieranie contentu w współpracujących z LPS serwisach WAP. Stan punktów dla MSISDN przechowywany jest w tabeli LP_POINTS_RECORDS. Po pobraniu contentu przez użytkownika tworzony jest rekord w tabeli LP_CONTENT_RECORDS z informacjami o tym, kto, kiedy, pobrał jaki content (id) i ile punktów odjęte mu zostało z konta. Funkcjonalności te umożliwiają interfejs LPWAPClient, dostarczany serwisom WAP za pośrednictwem biblioteki av-avantis-landingpage-api.

Przykład:

Dodawanie punktów do konta użytkownika (po zbilowaniu)

```
2014-06-17 14:58:13,399 | DEBUG | [26976967807][101176] | mp.handler.MoReminderHandler | Payload contents: AddPointsCommand{serviceName=PREMIUM_60355, msisdn=48506986496, points=null, operator=POLAND_P4, priceNet=null}
2014-06-17 14:58:13,399 | INFO | [26976967807][101176] | mp.handler.MoReminderHandler | Handler not responsible for payload, passing to successor: class pl.avantis.lp.core.mp.handler.CommandObjectPayloadHandler
2014-06-17 14:58:13,399 | INFO | [26976967807][101176] | mp.handler.CommandObjectPayloadHandler | Payload contents: AddPointsCommand{serviceName=PREMIUM_60355, msisdn=48506986496, points=null, operator=POLAND_P4, priceNet=null}
2014-06-17 14:58:13,401 | DEBUG | [26976967807][101176] | mp.executor.AddPointsExecutor | About to invoke execute(AddPointsCommand{serviceName=PREMIUM_60355, msisdn=48506986496, points=null, operator=POLAND_P4, priceNet=null})
2014-06-17 14:58:13,401 | INFO | [26976967807][101176] | mp.executor.AddPointsExecutor | About to add points to msisdn: 48506986496 for service name: PREMIUM_60355
2014-06-17 14:58:13,401 | DEBUG | [26976967807][101176] | mp.executor.AddPointsExecutor | Points in command: null
2014-06-17 14:58:13,403 | DEBUG | [26976967807][101176] | orm.jpa.GenericJPA | found 1 Service(s)
2014-06-17 14:58:13,403 | DEBUG | [26976967807][101176] | mp.executor.AddPointsExecutor | Points from Service: 3
2014-06-17 14:58:13,403 | DEBUG | [26976967807][101176] | orm.jpa.GenericJPA | object PointsRecord{id=111296039, msisdn=48506986496, serviceName=PREMIUM_60355, points=3} persisted.
2014-06-17 14:58:13,403 | INFO | [26976967807][101176] | mp.executor.AddPointsExecutor | PointsRecord{id=111296039, msisdn=48506986496, serviceName=PREMIUM_60355, points=3} points added
```

Wysyłka wiadomości MT - SMS, WAPPush

W trakcie subskrypcji użytkownika do serwisu MT za pomocą strony LP stosowana jest weryfikacja wprowadzonego numeru telefonu za pomocą wygenerowanego kodu (tokena). W przypadku websubskrypcji generowanie takiej wiadomości jest niezależne od LPS. W pozostałych przypadkach, tj. operatorów, którzy nie obsługują subskrypcji webowej, wiadomość taka generowana i wysyłana jest przez mpsservice [LandingPage](#). Dalsze kroki dokroczenia subskrypcji odbywają się podobnie jak w subskrypcji webowej, natomiast pomijane jest przy tym pośrednictwo MTSubscriber. Token wysyłany do użytkownika jest równocześnie zapisywany do bazy w tabeli LP_TOKENS i po wprowadzeniu go przez użytkownika na stronie LP porównywany jest z rekordem w bazie.

Wiadomości SMS MT wysyłane są przez mpsservice [LandingPage](#) również w przypadku generacji tzw. fake pinu. W takim wypadku token (fake pin) generowany jest nie losowo, lecz algorytmicznie na podstawie numeru MSISDN i nie jest przechowywany w bazie, lecz tylko wysyłany do użytkownika. Algorytm generacji fake pinu zaimplementowany jest również w kreacji flashowej danego LP i na tej podstawie zachodzi jego weryfikacja. Fake pin wysyłany jest w momencie otrzymania powiadomienia o rozpoczęciu subskrypcji MT przez dany MSISDN. Mechanizm ten wykorzystywany jest przez strony LP. Fake pin ten jest również wykorzystywany przez serwisy contentowe współpracujące z LP. Możliwe jest wówczas wysłanie przypomnienia takiego otrzymanego przy subskrypcji pinu. W takim wypadku akcja użytkownika na stronie powoduje puknięcie w metodę na interfejsie LPWAPClient. Aby zabezpieczyć się przed masowym wysyłaniem wiadomości do danego numeru MSISDN wprowadzony jest limit 2 wiadomości z fake pinem na dzień. Limit ten weryfikowany jest na podstawie rekordu, który tworzony jest w tabeli LP_FAKE_PIN_RECORDS przy pierwszej próbie manualnej wysyłki (przypomnienia) pinu i aktualizowany przy każdej kolejnej. Tabela przechowuje informacje o tym, kto dał przypomnienie (MSISDN), kiedy to miało miejsce (LAST_REQUESTED) i czy tego dnia osiągnięty został dopuszczalny limit ilości wysyłki pinów (DAILY_COUNT).

Parametry wysyłki fake pinów (wysyłający numer LA, domyślna treść wiadomości, długość pinu) konfigurowalne są w panelu admina w zakładkach: USTAWIENIA lub SERWISY.

W ramach niektórych kampanii LP jedną z funkcjonalności jest również wysyłka do zapisanych w danej kampanii użytkowników WAPPusha z linkiem do pobrania contentu. Jest to również realizowane przez mpsservice [LandingPage](#).

Przykad:

Wysyka fake pin

```
2011-05-25 11:54:08,681 | INFO | service.impl.LPFakePinManagerImpl | About to send fake pin: 93446
to msisdn: 48501084778
2011-05-25 11:54:08,681 | INFO | lp.mp.LPMPService | Sending message type: TEXT_SMS
to msisdn: 48501084778 from LA: 7936 with body: Twój kod to 93446
2011-05-25 11:54:08,681 | INFO | mp.service.MPMessageExchangeContextImpl | mp-message persisted : true ,
disableMessagePersistence :false
```

Zrzucanie RBR dla wybranych serwisów MT

Po otrzymaniu powiadomienia o zbilowaniu MSISDN w serwisie MT możliwe jest również zrzucenie RBR przychodowego (SMS_PAYMENT), o ile zostało to ustawione w konfiguracji LP, z którego dany numer został pozyskany.

Druga opcja, również konfigurowalna per LP to zrzucanie RBR kosztowych (PRODUCT_PROVISION) w momencie powiadomienia afilianta. Ma to na celu, podobnie jak samo powiadomienie, ułatwienie rozliczania się z ruchu generowanego przez danego afilianta i jego skuteczności.

Wysyka wiadomości e-mail do użytkowników

Niektóre LP mogą mieć możliwość podania przez użytkownika adresu/adresów e-mail, na który ma zostać wysłana wiadomość z materiałami promocyjnymi. Wysyłka realizowana jest za pośrednictwem serwera i użytkownika konfigurowalnych w pliku `classes/pl/avantis/lp/service/impl/LPMailerImpl.properties`

Przykad:

```
2014-06-18 10:47:47,043 | DEBUG | RMI TCP Connection(73195)-192.168.2.100 | core.impl.LPMailerImpl | Got
config: Config{id=217057, url=www.sp.autumn.premium.superserwisy.pl, affiliateURL=http://www2.balao.de/SPE8a?transaction\_id={0}}, fakeAffiliateURL=false, displayCode=false, affiliateId=11050, variantId=84050,
campaignName=Autumn, variantName=premium, additionalCode=<script type="text/javascript">

/* <![CDATA[ */

var google_conversion_id = 1010775343;

var google_custom_params = window.google_tag_params;

var google_remarketing_only = true;

/* ]]> */

</script>

<script type="text/javascript" src="//www.googleadservices.com/pagead/conversion.js">

</script>

<noscript>

<div style="display:inline;">



</div>

</noscript>, additionalCodeHeader=null, additionalCodeFlash=null, notificationTimeout=null,
googleAnalyticsTrackingId=UA-30964654-6}
2014-06-18 10:47:47,043 | DEBUG | RMI TCP Connection(73195)-192.168.2.100 | core.impl.LPMailerImpl |
Message from campaign: Cze,

Twój znajomy stwierdzi, że może Cię to zainteresować.
Wejd i sprawdź:
www.av.autumn.premium.superserwisy.pl

Serdecznie pozdrawiamy,

Digital Alea Sp. z o.o.
2014-06-18 10:47:47,043 | DEBUG | RMI TCP Connection(73195)-192.168.2.100 | core.impl.LPMailerImpl |
Parametrized message: Cze,
```

Twój znajomy stwierdzi, e moe Ci to zainteresowa.

Wejd i sprawd:

www.av.autumn.premium.superserwisy.pl

Serdecznie pozdrawiamy,

Digital Alea Sp z o.o.

2014-06-18 10:47:47,043 | INFO | RMI TCP Connection(73195)-192.168.2.100 | core.impl.LPMailerImpl | Sending message body: Cze,

Twój znajomy stwierdzi, e moe Ci to zainteresowa.

Wejd i sprawd:

www.av.autumn.premium.superserwisy.pl

Serdecznie pozdrawiamy,

Digital Alea Sp z o.o. with subject: Zaproszenie od znajomego to recipient address: Damil123XD@wp.pl

2014-06-18 10:47:47,045 | INFO | pool-84-thread-1 | service.edw.RestEdwSender | Queue purging thread fired...

2014-06-18 10:47:47,082 | DEBUG | RMI TCP Connection(73195)-192.168.2.100 | core.impl.LPMailerImpl | Successfully sent message: javax.mail.internet.MimeMessage@71651dde

landing-page-web

Aplikacja webowa, dziaajca na clustrze lp-tomcat0.avantis.pl / lp-tomcat1.avantis.pl. Loguje do `${LOG_PATH}/landing-page-web.log` ze standardow rotacj dzienn logów. Korzysta z interfejsu [RMI LPDBClient](#) udostpnionego pod adresem <rmi://nl.lp.avantis.pl:5678/LPDBClient> przez Landing Page Service.

Jej przeznaczeniem jest pośredniczenie w komunikacji pomidzy stronami LP a LPS. W tym celu udostpnia interfejs REST pod adresem <http://lp.klublider.pl/landing-page-web/api/rest/>.

Funkcjonalnoci:

- zapis uytkownika w bazie LPS
- wysanie dania rozpoczcia websubskrypcji (generowanie tokenu)
- wysanie dania potwierdzenia websubskrypcji (potwierdzenie tokenu)
- metody pomocnicze, np. sprawdzanie operatora, sprawdzanie listy serwisów MT, do jakich zapisany jest MSISDN.

Dokumentacja metod interfejsu REST dostpna jest w dokumencie:

- [LPRestAPI.pdf](#)

landing-page-admin

Aplikacja webowa, dziaajca na clustrze lp-tomcat0.avantis.pl / lp-tomcat1.avantis.pl. Loguje do `${LOG_PATH}/landing-page-admin.log` ze standardow rotacj dzienn logów. Korzysta z interfejsu [RMI LPDBClient](#) udostpnionego pod adresem <rmi://nl.lp.avantis.pl:5678/LPDBClient> przez Landing Page Service.

Jest to aplikacja przeznaczona dla pracowników biznesowych, oferuje panel administracyjny konfiguracji LP. Umoliwia dodawanie/usuwanie/edycj konfiguracji poszczególnych instancji stron LP, a take ich deployment pod publicznym URL z zakresu domen ustalonego wczesniej z DNIU, w postaci strony JSP z osadzon kreacj flash. Oferuje równie dodatkowe funkcjonalnoci, jak zarzdzanie afiliandami, czy konfiguracjami LA i serwisów MT z punktu widzenia ich uytecznoci dla LP.

Aplikacja posiada rozszerzony webcheck (<http://lp.klublider.pl/landing-page-admin/webcheck>) - dodana jest funkcjonalno sprawdzenia, czy moliwe jest zalogowanie do aplikacji przez AD. W poprawnym przypadku check ten zwraca:

```
<map>
  <entry>
    <string>RMI</string>
    <StatusDTO>
      <feature>RMI</feature>
      <status>OK</status>
```

```

        <invocationTimeMs>13</invocationTimeMs>
        <message>
            checked bean: &lpMPServiceCreator: checked bean: &avssClient: checked bean:
            lpMPServiceCreator: checked bean: &lpServicesManager: checked bean: skssClient: checked
            bean: lpPixelService: checked bean: &lpValidator: checked bean: lpValidator: OK; checked
            bean: lpDbClient: OK; checked bean: &lpMpService: checked bean: lpDataLoader: OK; checked
            bean: &lpPixelService: checked bean: &lpDataLoader: checked bean: lpRbrManager: OK;
            checked bean: mtspBeanService: OK; checked bean: &mtspBeanService: checked bean:
            lpMonitoringService: OK; checked bean: &lpRbrManager: checked bean: avssClient: checked
            bean: &skssClient: checked bean: lpAdwords: checked bean: &lpAdwords: checked bean: &
            lpDbClient: checked bean: lpMpService: OK; checked bean: lpServicesManager: checked bean:
            pcgamesClient: OK; checked bean: &lpMonitoringService: checked bean: &pcgamesClient:
        </message>
    </StatusDTO>
</entry>
<entry>
    <string>AD</string>
    <StatusDTO>
        <feature>AD</feature>
        <status>OK</status>
        <invocationTimeMs>20</invocationTimeMs>
        <message>Successfully logged in test user</message>
    </StatusDTO>
</entry>
<entry>
    <string>OK</string>
    <StatusDTO>
        <feature>OK</feature>
        <status>OK</status>
        <invocationTimeMs>0</invocationTimeMs>
        <message>this method always works</message>
    </StatusDTO>
</entry>
<entry>
    <string>DS</string>
    <StatusDTO>
        <feature>DS</feature>
        <status>OK</status>
        <invocationTimeMs>0</invocationTimeMs>
        <message/>
    </StatusDTO>
</entry>
</map>

```

Automatyczny deployment stron LP

Aplikacja landing-page-admin umożliwia użytkownikowi zautomatyzowany deployment instancji strony LP. W konfiguracji LP użytkownik może dodać pliki strony (obowiązkowo plik *.swf / *.zip, opcjonalnie *.css oraz inne). Akceptowane typy plików konfigurowalne są w parametrze `edit.deploy.types` w pliku `WEB-INF/classes/pl/avantis/lp/admin/resource/messages.properties`. Nazwy plików użytkownika przechowywane są w bazie LPS (schemat `LANDING_PAGES`) w tabeli `LP_CAMPAGNS_RESOURCES` i `LP_VARIANTS_RESOURCES`.

Proces deploymentu wygląda następująco:

1. Utworzenie kampanii.
2. Dodanie plików do kampanii (zapis w `LP_CAMPAGNS_RESOURCES`)
3. Dodanie wariantu
4. Wybór odpowiednich plików, które zostaną dodane w kampanii (zapis w `LP_VARIANTS_RESOURCES`)
5. Wybór afilianta
6. Automatycznie tworzony jest URL strony LP na podstawie: nazwy afilianta, wariantu, kampanii, wybranej domeny w wariantcie.
7. Po zapisaniu pod wygenerowanym adresem dostępna jest strona LP.

Weryfikacja hostów wirtualnych

W chwili obecnej mechanizm jest wyczerpany, a weryfikację zajmuje skrypt napisany przez Kamila

Ponieważ po restarcie serwera, na którym stoją wszystkie zdeployowane instancje stron LP czasami "gubione" są hosty wirtualne, dodany został w aplikacji landing-page-admin job do weryfikacji, który odpala się co określony czas (domyślnie co 5 minut, edytowalne za pomocą formuły quartzowej w `landing-page-admin-ctx.xml`). Sprawdza on wszystkie konfiguracje LP z tabeli `LP_CONFIGS` pod kątem statusu deploymentu

(warto DEPLOYED 1/0) i dokonuje weryfikacji, czy host dla danej konfiguracji istnieje. Jeżeli konfiguracja jest zdeployowana (DEPLOYED 1), a w liście hostów wirtualnych nie ma jej adresu, jest on dodawany. Analogicznie, jeżeli host istnieje, a konfiguracja jest nieaktywna (DEPLOYED 0), host jest usuwany. Operacja ta wykonywana jest dla serwerów z listy `hostManagerUrls`, czyli w praktyce dla obu wzów klastra.

Przykład:

Sprawdzanie istniejących hostów

```
2011-06-13 13:25:00,018 | INFO | admin.deploy.LPDeployerImpl | Verifying LP host deployment status...
2011-06-13 13:25:00,566 | DEBUG | admin.deploy.HostManagerClientImpl | Invoking GET method for URL:
http://lp-tomcat0.avantis.pl:8080/host-manager/text/list
2011-06-13 13:25:00,570 | DEBUG | admin.deploy.HostManagerClientImpl | HTTP GET method executed. ResultCode
= 200
2011-06-13 13:25:00,570 | DEBUG | admin.deploy.LPDeployerImpl | Got deployed hosts: [OK - Listed hosts,
www.s3.bogaty.superserwis.pl:s3.bogaty.superserwis.pl, jmes2.pl:www.jmes2.pl, www.jmes.es:jmes.es,
landpages.avantis.pl:www.landpages.avantis.pl, localhost:,
www.fb.bogaty.superserwis.pl:fb.bogaty.superserwis.pl,
www.chlopaki.superserwis.pl:chlopaki.superserwis.pl, www.lp2.ns.klublider.pl:lp2.ns.klublider.pl,
www.nimfomanka.superserwis.pl:nimfomanka.superserwis.pl,
www.singiel2.superserwis.pl:singiel2.superserwis.pl,
www.ns.bogaty.superserwis.pl:ns.bogaty.superserwis.pl,
www.nimfomanka2.superserwis.pl:nimfomanka2.superserwis.pl,
www.bogaty.superserwis.pl:bogaty.superserwis.pl, www.gg.bogaty.superserwis.pl:gg.bogaty.superserwis.pl,
www.singiel.superserwis.pl:singiel.superserwis.pl, www.jmes2.es:jmes2.es] for host manager:
http://lp-tomcat0.avantis.pl:8080/host-manager
```

Dodawanie brakującego hosta

```
2011-06-13 13:25:00,570 | INFO | admin.deploy.LPDeployerImpl | Host www.2lp2.ns.klublider.pl should be
deployed, but is not, adding host...
2011-06-13 13:25:00,570 | DEBUG | admin.deploy.HostManagerClientImpl | Invoking POST method for URL:
http://lp-tomcat0.avantis.pl:8080/host-manager/html/add, name: www.2lp2.ns.klublider.pl, aliases:
2lp2.ns.klublider.pl, appBase:/usr/local/webapps/www.2lp2.ns.klublider.pl
2011-06-13 13:25:00,605 | DEBUG | admin.deploy.HostManagerClientImpl | HTTP POST method executed.
ResultCode = 200
2011-06-13 13:25:00,606 | DEBUG | admin.deploy.HostManagerClientImpl | Invoking POST method for URL:
http://lp-tomcat1.avantis.pl:8080/host-manager/html/add, name: www.2lp2.ns.klublider.pl, aliases:
2lp2.ns.klublider.pl, appBase:/usr/local/webapps/www.2lp2.ns.klublider.pl
2011-06-13 13:25:01,026 | DEBUG | admin.deploy.HostManagerClientImpl | HTTP POST method executed.
ResultCode = 200
```

Usuwanie nadmiarowego hosta

```
2011-06-13 12:20:03,649 | INFO | admin.deploy.LPDeployerImpl | Host www.wakacje.superserwis.pl is
deployed, but shouldn't be, removing host...
2011-06-13 12:20:03,649 | DEBUG | admin.deploy.HostManagerClientImpl | Invoking POST method for URL:
http://lp-tomcat0.avantis.pl:8080/host-manager/html/remove, name: www.wakacje.superserwis.pl, aliases: ,
appBase:
2011-06-13 12:20:04,474 | DEBUG | admin.deploy.HostManagerClientImpl | HTTP POST method executed.
ResultCode = 200
2011-06-13 12:20:04,474 | DEBUG | admin.deploy.HostManagerClientImpl | Invoking POST method for URL:
http://lp-tomcat1.avantis.pl:8080/host-manager/html/remove, name: www.wakacje.superserwis.pl, aliases: ,
appBase:
2011-06-13 12:20:04,491 | DEBUG | admin.deploy.HostManagerClientImpl | HTTP POST method executed.
ResultCode = 200
```

Integracja z JMES

W ramach zwiększenia liczby afiliantów, współpracujemy z Jet Multimedia w Hiszpanii. JMES zapewnia obsługę dostarczanych przez siebie afiliantów na stronach LP wystawianych i generowanych przez siebie i komunikuje się z naszym LPS poprzez interfejs REST w `landing-page-web`, korzystając z metod używanych również przez wystawiane przez nas LP. Różnica polega na tym, że rekordy generowane przez LP JMES posiadają dodatkowy parametr `JMES_ID`, który wykorzystywany jest przez JMES w logice po ich stronie.

Dodatkowo, JMES udostępnia nam API HTTP, umożliwiające zapis i update rekordów w ich bazie z pewnymi danymi, dotyczącymi statusu subskrypcji poszczególnych użytkowników. Na podstawie tych danych JMES dokonuje rozliczeń z ruchem z poszczególnymi afiliantami.

Automatyczne monitorowanie stron LP

Na serwerze nagios w cronie użytkownika dyzur od poniedziałku do piątku co 15 minut jest uruchamiany skrypt:

```
*/15 * * * 1-5 /home/dyzur/DNIU/monitoring_LP/lp.sh
```

który:

1. Wyciąga bieżące konfiguracje aktywnych stron LP WWW za pomocą zapytania:

```
Select lpc.url  
from Landing_Pages.lp_configs lpc  
join Landing_Pages.lp_variants lpv on lpc.variant_id = lpv.id  
Join Landing_Pages.Lp_Campaigns Lpcamp On Lpv.Campaign_Id = Lpcamp.Id  
Where Lpcamp.Name Not Like 'Migration%'  
And Lpc.Deployed=1  
And Lpc.Url not In('www.jmes.test.ero.es', 'www.jmes.test.ok.es')
```
2. Wyciąga bieżące konfiguracje aktywnych stron mobilnych LP

```
Select lpc.url  
from Landing_Pages.lp_configs lpc  
Join Landing_Pages.Lp_Variants Lpv On Lpc.Variant_Id = Lpv.Id  
Where Exists (Select Key From Landing_Pages.Lp_Variants_Resources Where Variant = Lpv.Id And Key =  
'zip')  
And Lpc.Deployed=1
```
3. Tworzy w locie nową konfigurację nagiosa wykorzystując nagiosowy check_http_string sprawdzający frazę main, która jest na każdej stronie LP (strony WWW);
4. Tworzy w locie nową konfigurację nagiosa dla stron mobilnych wykorzystując podszywając się w nagłówku jako telefon komórkowy (przykładowe wywołanie ./check_http -H www.av.paliwo.premium.superserwis.pl -A "SAMSUNG-GT-S8600/S8600XXLC3" -s "main" -f follow);
5. Porównuje nowy plik konfiguracyjny ze obecnym (https://svn.avantis.pl/repos/DNIU/nagios.pi/newcfg_app/virtual.lp/services.cfg)
6. W sytuacji gdy nowa wersja pliku konfiguracyjnego jest inna niż obecny plik comituje nową konfigurację oraz reloaduje nagiosa.

Reakcja na alerty stron LP:

1. Sprawdzenie poprawności działania wszystkich elementów architektury Landing Pages
2. Powodów problemu może być wiele począwszy od apacha lp-webproxy, tomcatów na lp-tomcat0 i lp-tomcat1 po samą aplikację landing pages więc w zależności od sytuacji zwracamy się o pomoc do Macieja Turo (aplikacja) bądź Kamila Lachtarego (tomcat, apache)

Notyfikacje monitorowania

Skrypt wysyła notyfikację (sms + mail) do biznesu w przypadku gdy któraś ze stron LP nie działa dłużej niż 20 minut.

```
*/20 8-22 * * 1-5 /home/mpdevelop/svn/DNIU/monitoring_LP/lp_notifier.pl
```

Komunikacja z aplikacjami: AVSS, EDW

Diagram sekwencji - Zapisanie do subskrypcji, flow SMS, SK



Logi:

2014-07-10 07:18:04,094 | INFO | RMI TCP Connection(18369)-192.168.2.100 | core.impl.LPDBClientImpl | Saving Order: OrderDTO{id=null, sessionId=null, msisdn=421908891123, operator=SLOVAK_REPUBLIC_TELEFONICA_O2, ip=95.102.23.161, referer=null, creationDate=Thu Jul 10 07:18:04 CEST 2014, updateDate=null, subscriptionDate=null, unsubscriptionDate=null, firstBilled=null, lastBilled=null, billCount=0, regChannel=null, affiliateParamValues=2a424c81-3368-4d8d-bf3e-418a96c78114, lpParamValues=null, subscriptionId=PL_hRFI96d8mQemQvwlT89IqvcEY7fcOnTXM2r9QnN4oHWlfhs9aA, token=null, tokenVerDate=null, campaign=null, provider=null, channel=null, configId=224052, service=null, serviceSource=null, extCode=null, lpDeliveryStatus=null, rbrProvisionDate=null, protocolId=null, partnerParams=null}

2014-07-10 07:18:04,102 | INFO | RMI TCP Connection(18369)-192.168.2.100 | core.impl.LPDBClientImpl | Saving Event SAVE_MSISDN for msisdn: 421908891123, configId: 224 052 with subscriptionId: PL_hRFI96d8mQemQvwlT89IqvcEY7fcOnTXM2r9QnN4oHWlfhs9aA

2014-07-10 07:18:04,155 | INFO | RMI TCP Connection(18369)-192.168.2.100 | core.impl.LPEdwManagerImpl | Got promoEvent: PromoEvent{promoSession=PromoSession{sessionId=4210279785, ip='95.102.23.161', msisdn='421908891123', subscriptionId='PL_hRFI96d8mQemQvwlT89IqvcEY7fcOnTXM2r9QnN4oHWlfhs9aA', regChannel='WWW', operatorId=23106, campaign=PromoCampaign{campaignId=68050, name='null', description='null', test=false, countryCode='null'}, affiliate=pl.avantis.edw.api.promo.PromoAffiliate@83557e0, service=pl.avantis.edw.api.promo.PromoService@661c3755, variant=pl.avantis.edw.api.promo.PromoVariant@4674e1d2, website=pl.avantis.edw.api.promo.PromoWebsite@2395c621}, eventId=194288791, url='null', response='null', eventDate=Thu Jul 10 07:18:04 CEST 2014, params='null', eventType=PromoEventType{type='SAVE_MSISDN', affNotification=false, description='Wpisanie numeru telefonu przez uytownika na kreacji LP, zapisanie Order'u'}, eventStatus=null, operatorId=23106, transactionId=null}

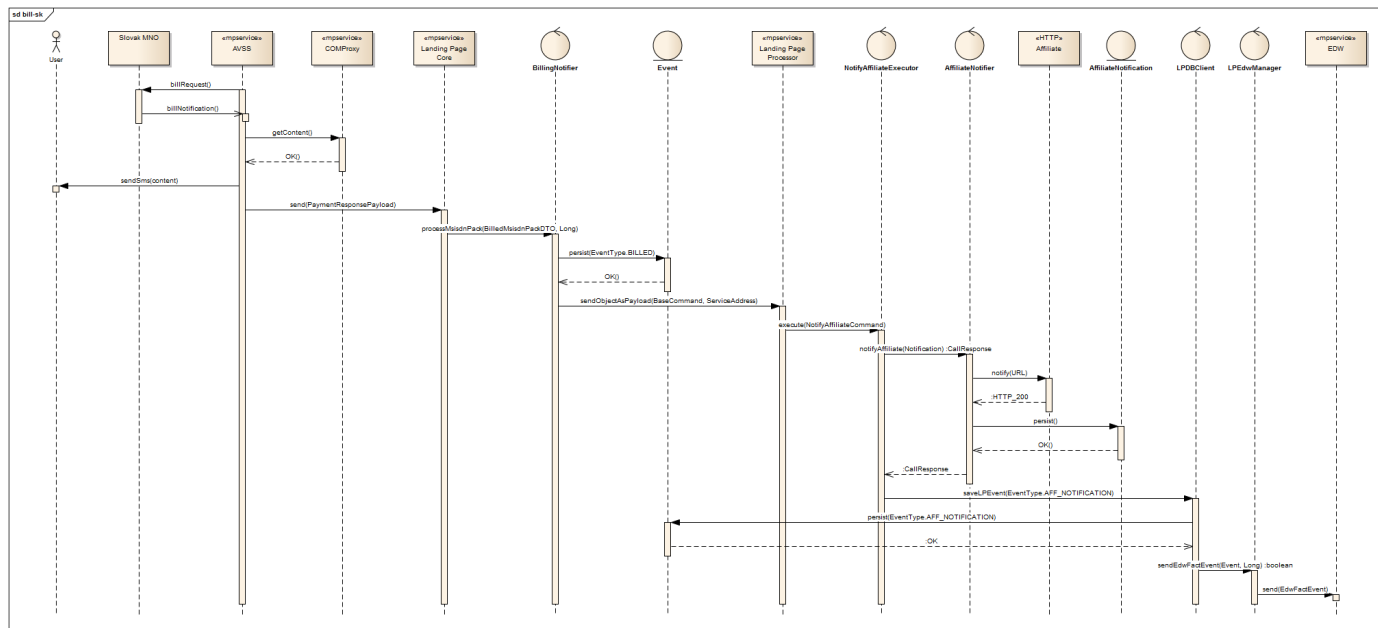
2014-07-10 07:19:27,220 | INFO | [27102997361][120007] | mp.handler.SubscriberHandler | SubscriberDTO: SubscriberDTO{msisdn='421908891123', updateDate=Thu Jul 10 07:19:26 CEST 2014, status=10, operator=2310602, serviceName='WIN_7770', parameter='', source='null', subscriberId='27102997351', promoSessionId='null'}

2014-07-10 07:19:27,227 | INFO | [27102997361][120007] | processor.notifier.CommandSendingSubscriptionNotifier | Subscriber received: SubscriberDTO{msisdn='421908891123', updateDate=Thu Jul 10 07:19:26 CEST 2014, status=10, operator=2310602, serviceName='WIN_7770', parameter='', source='null', subscriberId='27102997351', promoSessionId='null'} for orderId: 90 170 817

2014-07-10 07:19:27,246 | INFO | [27102997361][120007] | core.impl.LPDBClientImpl | Saving Event SUBS_CONFIRMATION for msisdn: 421908891123, configId: 224 052 with params: null, priceNet: null, affiliateNotificationId: null

2014-07-10 07:19:27,316 | DEBUG | [27102997361][120007] | mp.handler.SubscriberHandler | Queued commands: {SendAddedMessageCommand{orderId=90170817}=[id=101176,landing-page-product-center], SendTriggeredMessageCommand{orderId=90170817}=[id=101176,landing-page-product-center], SendFakePinCommand{msisdn=421908891123, serviceName=WIN_7770, orderId=90170817}=[id=101176,landing-page-product-center], SendIplayTrackCommand{msisdn=421908891123, orderId=90170817, trackId=null}=[id=101176,landing-page-product-center], SendGadgetCommand{msisdn=421908891123, orderId=90170817}=[id=101176,landing-page-product-center], NotifyTrack4SemCommand{regChannel=WWW, leadId=null, orderId=90170817, operator=SLOVAK_REPUBLIC_TELEFONICA_O2}=[id=120007,lp-ALeadPay-120007], NotifyDveCommand{orderId=90170817}=[id=120007,lp-ALeadPay-120007], NotifyAffiliateCommand{orderId=90170817, billCount=0, priceNet=null, sendRbrCommand=SendRbrCommand{msisdn=421908891123, operator=SLOVAK_REPUBLIC_TELEFONICA_O2, transactionDate=Thu Jul 10 07:19:26 CEST 2014, transactionId=27102997361, orderId=90170817, serviceName=WIN_7770, defaultRBR=false, billCode=null, affiliateNotificationId=null, priceNet=null}}=[id=120007,lp-ALeadPay-120007]}

Diagram sekwencji - bilowanie:



Logi:

2014-07-10 07:19:36,329 | INFO | [27102998107][100980] | core.mp.CoreMPService | Message delivered, tid=[27102998107], sender=[id=101078,avantis-subscription-service]], payload=pl.avantis.mp.payload.payment.PaymentResponsePayload [supportsAccounting=true, supportsAcknowledge=true, sender=421908891123, recipient=WIN_7770, sendDate=Thu Jul 10 07:19:36 CEST 2014, externalSendDate=null, deliveryDate=null, validityDate=null, payloadDirection=INTERNAL, payloadType=PAYMENT_RESPONSE, extendedType=null, operator=SLOVAK_REPUBLIC_O2_NONBRAND, masterOperator=null, mobileTerminated=false, direction=UNDEFINED, isMobileTerminated=false, billCode='null', serviceCategory='null', sourcePayload=null, paymentType=null, commChannel=null, properties={}, request=pl.avantis.mp.payload.payment.PaymentRequestPayload [supportsAccounting=true, supportsAcknowledge=true, sender=WIN_7770, recipient=421908891123, sendDate=Thu Jul 10 07:19:36 CEST 2014, externalSendDate=null, deliveryDate=null, validityDate=null, payloadDirection=INTERNAL, payloadType=PAYMENT_REQUEST, extendedType=null, operator=SLOVAK_REPUBLIC_O2_NONBRAND, masterOperator=null, mobileTerminated=false, direction=UNDEFINED, isMobileTerminated=false, billCode='BILL', serviceCategory='WIN_7770', sourcePayload=null, paymentType=PaymentType{type=PERIODIC, period=86400000}, commChannel=SMS, properties={TRANSACTION_ID=AVSS1418667665, PAYMENT_ID=1418667665}}, status=PAYMENT_COMPLETE]

2014-07-10 07:19:36,417 | INFO | [27102998125][120007] | processor.notifier.CommandSendingBillingNotifier | Processing billed msisdn pack: BilledMsisdnPackDTO[status=[null]subscriptionId=[null][2310602][WIN_7770][Thu Jul 10 07:19:36 CEST 2014][BilledMsisdnDTO[421908891123][1]]]

2014-07-10 07:19:36,417 | INFO | [27102998125][120007] | processor.notifier.CommandSendingBillingNotifier | Processing msisdn: BilledMsisdnDTO[421908891123][1]

2014-07-10 07:19:36,417 | INFO | [27102998125][120007] | processor.notifier.CommandSendingBillingNotifier | Billing date: 10.07.14 07:19 for msisdn: 421908891123, service: WIN_7770 and operator: SLOVAK_REPUBLIC_O2_NONBRAND

2014-07-10 07:19:36,426 | DEBUG | [27102998125][120007] | core.impl.LPDBClientImpl | About to invoke saveLPEvent('421908891123', SLOVAK_REPUBLIC_TELEFONICA_O2, 224052, 90170817, BILLED, Thu Jul 10 07:19:36 CEST 2014, null, null)

2014-07-10 07:19:36,426 | INFO | [27102998125][120007] | core.impl.LPDBClientImpl | Saving Event BILLED for msisdn: 421908891123, configId: 224 052 with params: null, priceNet: null, affiliateNotificationId: null

2014-07-10 07:19:36,426 | DEBUG | [27102998125][120007] | orm.jpa.GenericJPA | object Event{id=194288861, msisdn=421908891123, configId=224052, eventDate=Thu Jul 10 07:19:36 CEST 2014, type=BILLED, params=null, priceNet=null} persisted.

2014-07-10 07:19:36,736 | DEBUG | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | About to invoke notifyAffiliate(Notification{orderId=90170817, configId=224052, msisdn=421908891123, affiliateName=ALeadPay, affiliateParamValues=null, sendRbrCommand=SendRbrCommand{msisdn=421908891123, operator=SLOVAK_REPUBLIC_O2_NONBRAND, transactionDate=Thu Jul 10 07:19:36 CEST 2014, transactionId=27102998125, orderId=90170817, serviceName=WIN_7770, defaultRBR=false, billCode=null, affiliateNotificationId=null, priceNet=null}})

2014-07-10 07:19:36,736 | DEBUG | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | CustomerID and OrderID not generated

2014-07-10 07:19:36,736 | INFO | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | Config: Config{id=224052, url=www.alp.superhra2.win.premiumclub.sk, affiliateURL=http://tracking1.aleadpay.com/Pixel/Server/1789/?cid={0}, fakeAffiliateURL=false, displayCode=false, affiliateId=16051, variantId=87050, campaignName=Superhra2, variantName=win, additionalCode=null, additionalCodeHeader=null, additionalCodeFlash=null, notificationTimeout=null, googleAnalyticsTrackingId=UA-30947875-7}

2014-07-10 07:19:36,736 | INFO | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | Notifying affiliate with url: http://tracking1.aleadpay.com/Pixel/Server/1789/?cid=2a424c81-3368-4d8d-bf3e-418a96c78114

2014-07-10 07:19:36,736 | INFO | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | NotificationTimeoutChecker result: CheckerReturn{timeout=false, millisSinceRegistration=9741, millisTimeout=null} for notification: Notification{orderId=90170817, configId=224052, msisdn=421908891123, affiliateName=ALeadPay, affiliateParamValues=null, sendRbrCommand=SendRbrCommand{msisdn=421908891123, operator=SLOVAK_REPUBLIC_O2_NONBRAND, transactionDate=Thu Jul 10 07:19:36 CEST 2014, transactionId=27102998125, orderId=90170817, serviceName=WIN_7770, defaultRBR=false, billCode=null, affiliateNotificationId=null, priceNet=null}}

2014-07-10 07:19:39,619 | INFO | [27102998129][120007] | core.affiliate.HttpAffiliateNotifier | Saving LP affiliate notification for Order(90 170 817), msisdn: 421908891123, URL: http://tracking1.aleadpay.com/Pixel/Server/1789/?cid=2a424c81-3368-4d8d-bf3e-418a96c78114, httpStatus: 200

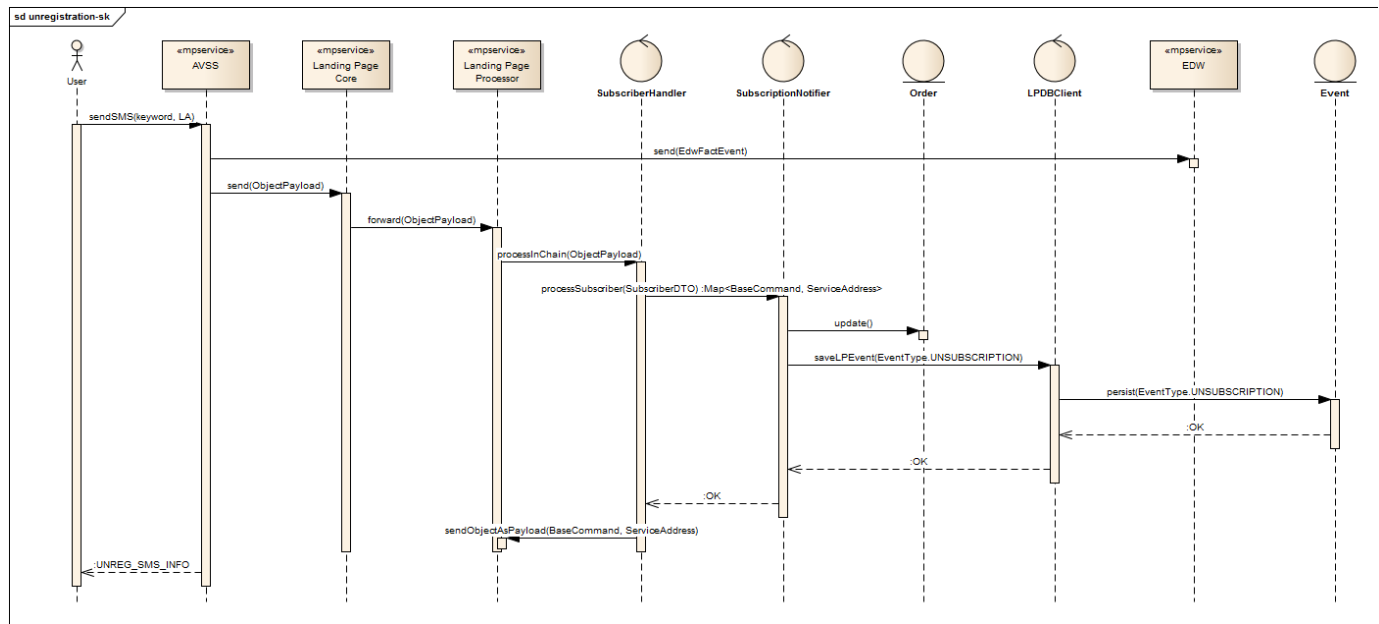
2014-07-10 07:19:39,755 | DEBUG | [27102998129][120007] | core.impl.LPDBClientImpl | About to invoke saveLPEvent('421908891123', SLOVAK_REPUBLIC_TELEFONICA_O2, 224052, 90170817, AFF_NOTIFICATION, null, null, 10770051)

2014-07-10 07:19:39,755 | INFO | [27102998129][120007] | core.impl.LPDBClientImpl | Saving Event AFF_NOTIFICATION for msisdn: 421908891123, configId: 224 052 with params: null, priceNet: null, affiliateNotificationId: 10 770 051

2014-07-10 07:19:39,755 | DEBUG | [27102998129][120007] | orm.jpa.GenericJPA | object Event{id=194288868, msisdn=421908891123, configId=224052, eventDate=Thu Jul 10 07:19:39 CEST 2014, type=AFF_NOTIFICATION, params=null, priceNet=null} persisted.

2014-07-10 07:19:39,758 | DEBUG | [27102998129][120007] | core.impl.LPEdwManagerImpl | About to invoke sendEdwFactEvent(Event{id=194288868, msisdn=421908891123, configId=224052, eventDate=Thu Jul 10 07:19:39 CEST 2014, type=AFF_NOTIFICATION, params=null, priceNet=null}, 10770051)

Diagram sekwencji - wypisanie z subskrypcji:



Logi:

```

2014-07-10 07:20:26,473 | INFO | [27103001608][120007] | mp.handler.SubscriberHandler | SubscriberDTO:
SubscriberDTO{msisdn='421908891123', updateDate=Thu Jul 10 07:20:26 CEST 2014, status=2, operator=2310602,
serviceName='WIN_7770', parameter='null', source='null', subscriberId='27102997351', promoSessionId='null'}
2014-07-10 07:20:26,520 | INFO | [27103001608][120007] |
processor.notifier.CommandSendingSubscriptionNotifier | Unsubscribe: Order{id=90170817,
sessionId=4210279785, msisdn=421908891123, operator=SLOVAK_REPUBLIC_TELEFONICA_O2, ip=95.102.23.161,
referer=null, creationDate=Thu Jul 10 07:14:58 CEST 2014, updateDate=Thu Jul 10 07:19:36 CEST 2014,
subscriptionDate=Thu Jul 10 07:19:26 CEST 2014, unsubscriptionDate=null, firstBilled=Thu Jul 10 07:19:36
CEST 2014, lastBilled=null, billCount=1, regChannel=WWW,
affiliateParamValues=2a424c81-3368-4d8d-bf3e-418a96c78114, lpParamValues=null, orderId=1620893687,
configId=224052, serviceId=30050, subscriptionId=PL_hRFI96d8mQemQvw1T89IqvcEY7fcOnTXM2r9QnN4oHW1fhs9aA,
token=null, campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=Thu Jul
10 07:19:39 CEST 2014, protocolId=null}
2014-07-10 07:20:26,520 | DEBUG | [27103001608][120007] | orm.jpa.GenericJPA | object Order{id=90170817,
sessionId=4210279785, msisdn=421908891123, operator=SLOVAK_REPUBLIC_TELEFONICA_O2, ip=95.102.23.161,
referer=null, creationDate=Thu Jul 10 07:14:58 CEST 2014, updateDate=Thu Jul 10 07:20:26 CEST 2014,
subscriptionDate=Thu Jul 10 07:19:26 CEST 2014, unsubscriptionDate=Thu Jul 10 07:20:26 CEST 2014,
firstBilled=Thu Jul 10 07:19:36 CEST 2014, lastBilled=null, billCount=1, regChannel=WWW,
affiliateParamValues=2a424c81-3368-4d8d-bf3e-418a96c78114, lpParamValues=null, orderId=1620893687,
configId=224052, serviceId=30050, subscriptionId=PL_hRFI96d8mQemQvw1T89IqvcEY7fcOnTXM2r9QnN4oHW1fhs9aA,
token=null, campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=Thu Jul
10 07:19:39 CEST 2014, protocolId=null} updated.
2014-07-10 07:20:26,520 | INFO | [27103001608][120007] |
processor.notifier.CommandSendingSubscriptionNotifier | Unsubscribe after update: Order{id=90170817,
sessionId=4210279785, msisdn=421908891123, operator=SLOVAK_REPUBLIC_TELEFONICA_O2, ip=95.102.23.161,
referer=null, creationDate=Thu Jul 10 07:14:58 CEST 2014, updateDate=Thu Jul 10 07:20:26 CEST 2014,
subscriptionDate=Thu Jul 10 07:19:26 CEST 2014, unsubscriptionDate=Thu Jul 10 07:20:26 CEST 2014,
firstBilled=Thu Jul 10 07:19:36 CEST 2014, lastBilled=null, billCount=1, regChannel=WWW,
affiliateParamValues=2a424c81-3368-4d8d-bf3e-418a96c78114, lpParamValues=null, orderId=1620893687,
configId=224052, serviceId=30050, subscriptionId=PL_hRFI96d8mQemQvw1T89IqvcEY7fcOnTXM2r9QnN4oHW1fhs9aA,
token=null, campaign=null, provider=null, channel=null, deliveryStatus=SUBSCRIBED, rbrProvisionDate=Thu Jul
10 07:19:39 CEST 2014, protocolId=null}
2014-07-10 07:20:26,520 | DEBUG | [27103001608][120007] | core.impl.LPDBClientImpl | About to invoke
saveLPEvent('421908891123', SLOVAK_REPUBLIC_TELEFONICA_O2, 224052, 90170817, UNSUBSCRIPTION, Thu Jul 10
07:20:26 CEST 2014)
2014-07-10 07:20:26,521 | INFO | [27103001608][120007] | core.impl.LPDBClientImpl | Saving Event
UNSUBSCRIPTION for msisdn: 421908891123, configId: 224 052 with params: null, priceNet: null,
affiliateNotificationId: null
2014-07-10 07:20:26,521 | DEBUG | [27103001608][120007] | orm.jpa.GenericJPA | object Event{id=194288892,
msisdn=421908891123, configId=224052, eventDate=Thu Jul 10 07:20:26 CEST 2014, type=UNSUBSCRIPTION,
params=null, priceNet=null} persisted.
2014-07-10 07:20:26,528 | DEBUG | [27103001608][120007] | core.impl.LPDBClientImpl | Event saved under id:
194 288 892

```

Zapytania i Baza LP dla DNIU co sie z czym je

Opis bazy i paru najwazniejszych tabel aby dyurni mogli ogarna temat bazy LP i nie musieli zawraca girtary Makowi za kadym razem jak LP zrobi 'boom'. Ogolnie jest tak, kluczowymi dla nas tabelami bd zwykle lp_events oraz lp_orders, to w nich jest zapisywana caa aktywno LP. Najwazniejsze dla nas tabele to : *lp_variants *lp_configs *lp_orders *lp_services *lp_events *lp_affiliates *lp_aff_notifications

I tak zeby wyczescac bilowania w LP dla jakiegos serwisu:

```

select
count(*)
from lp_orders o, lp_services s, lp_events e
where s.service_name = 'PREMIUM'
and o.service_id = s.id
and e.order_id = o.id
and e.type = 'BILLED'
and event_date >= to_date('2014-06-10 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
and event_date < to_date('2014-06-17 00:00:00', 'yyyy-mm-dd hh24:mi:ss');

```

Aby wiazc i znalezc ile bylo 'pukni' do danego partnera musimy zajrze do tabeli LP_AFF_NOTIFICATION która czy si z LP_ORDERS, jeli znamy id affianta moemy spreparowa zapytanie i bez problemu sprawdzci ile notyfikacji wyszo do affianta. Jeli tego nie wiemy moemy sprawdzci w LP_AFFILIATES gdzie mamy affiantów po nazwie. Jeli szukamy konkretnego seriwsu znjdziemy go w LP_SERVICES razem z partner_name która pomoe, w tej tabeli mamy tez np. dane, które serwisy znajdujce si w LP maj swe pocztki w MTSP.

```
select name
from lp_services
where service_source = 'MTSP';
```

Wycigniecie notyfikacji jakie wyszy do afilianta: ALeadPay.

```
select
    count(*)
from lp_aff_notifications n, lp_orders o, lp_affiliates a, lp_configs c
where a.name = 'ALeadPay'
and a.id = c.affiliate_id
and c.id = o.config_id
and o.id = n.order_id
and n.last_tried >= to_date('2014-06-10 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
and n.last_tried < to_date('2014-06-17 00:00:00', 'yyyy-mm-dd hh24:mi:ss');
```

Lista orderów dla danego configu + info o rejestracji i powiadomieniu afilianta:

```
select
    o.id as order_id,
    o.msisdn as msisdn,
    o.subscription_date as subscription_date,
    n.last_tried as affiliate_notification_date
from
(
    select o.id, o.msisdn, o.subscription_date
    from lp_orders o
    where o.creation_date > to_date('2014-06-01', 'yyyy-MM-dd')
    and o.creation_date < to_date('2014-06-18', 'yyyy-MM-dd')
    and o.config_id = (select id from lp_configs where url = 'www.aw.gdatask.scan2.mobilum.pl')
) o
left join
(
    select o.id, o.msisdn, o.subscription_date, n.last_tried, n.order_id
    from lp_orders o, lp_aff_notifications n
    where o.creation_date > to_date('2014-06-01', 'yyyy-MM-dd')
    and o.creation_date < to_date('2014-06-18', 'yyyy-MM-dd')
    and o.config_id = (select id from lp_configs where url = 'www.aw.gdatask.scan2.mobilum.pl')
    and o.id = n.order_id
) n
on o.id = n.id
order by o.id;
```

Zalepka na zend do Landing Page

Teoria

Kamil L. skonfigurowa LP jako tak, e cz ruchu bez udziau VirtualHosts? Apache na webproxy i na zend trafia do zend do wyznaczonych (gdzie? jak? przez kogo?)podkatalogów i jest tam obsugiwana przez zalepki php.

na dzi tych zalepek jest trzy:

- paliwo
- bony
- singiel

Praktyka

task: prosz o zdeployowanie <https://svn.avantis.pl/repos/m-lp-singiel/trunk/> na serwerze n1.c1.zend.avantis.pl w katalogu /singiel/

analogicznie jak w tasku: <https://projekty.avantis.pl/view.php?id=653899>

1. Typ: Wdrozenie / Aktualizacja bony i paliwo
2. Lokalizacja w SVN:

- <https://svn.avantis.pl/repos/m-lp-singiel/trunk/>
- <https://svn.avantis.pl/repos/m-lp-bony/trunk/>
- <https://svn.avantis.pl/repos/m-lp-paliwo/trunk/>
- <https://svn.avantis.pl/repos/m-lp-application/trunk/>

3. System plików
4. Project Manager: ukasz Bujak
5. Developer: Patryk Raczyski
6. Analityk: Jakub Gontarek
7. Opis usugi (2-3 zdania; co usuga robi)
 - LP mobilne dla Francji - php
8. Uwagi:
9. Sposób testowania:
 - Naley wej z kom na strony:
 - www.dve.singiel.premium.superserwisy.pl Jeli strony wywietlaj si poprawnie jest ok