



[DOC] mQube - dokumentacja techniczna

Created by Michał MM. Mewald, last modified by Jakub JŻ. Żaczek on Dec 14, 2018

mQube (mQube, mQube-manager)

- mQube (mQube, mQube-manager)
 - MPService
 - WEB
 - Opis wdrożenia.
 - NAGIOS.
 - mqube:webapp
 - Zgłoszenia WWW:
 - PHP user interface:
 - Weryfikacja działania usługi:
 - mqube:service - błędy
 - mqube:service - ostrzeżenia
 - mqube:service - informacje
 - mqube:webapp
- TVP api
- Mqube.pl - frontend
 - Monitoring
 - Check HTTPS_mqube.avantis.pl
 - Check [0]Av SSRV mQube / main page login
 - Check [0]Av SSRV mQube / main page
 - Opis serwerów , usług i komponentów
 - Dostępne komponenty
 - Account
 - Opis ogólny
 - Wykorzystywane metody REST
 - LA/verbatimfixes
 - Opis ogólny
 - Wykorzystywane metody REST
 - Services
 - Opis ogólny
 - Wykorzystywane metody REST
 - Reports
 - Opis ogólny
 - Wykorzystywane metody REST
 - Reminders
 - Opis ogólny
 - Wykorzystywane metody REST

MPService

1. Typ: Wdrożenie
2. Lokalizacja w SVN:
 - binarki: <https://svn.avantis.pl/repos/dev/avantis/release/Services/mqube/bin/service>
 - źródła: <https://svn.avantis.pl/repos/dev/avantis/current/Services/mcube>
3. Project Manager: Anna Cygler
4. Developer: Ireneusz Koślacz
5. Analityk: Rafał Jaczyński
6. Opis usługi: Serwis mqube jest odpowiedzialny za przetwarzanie wiadomości. Tworzy mpserwis **mqube-manager (150000)** oraz dynamicznie inne mpserwisy w zależności od konfiguracji: jeden mpserwis dla grupy i la. Mają one nazwy w formacie

mqube-number-1

, gdzie 0 to id grupy i 1 to LA oraz opis w formacie

mqube-0-1

, gdzie 0 to nazwa grupy i 1 to LA. Na razie będzie tylko jedna grupa, jeden serwis i dwa la, czyli w sumie 3 mpserwisy łącznie z managerem. Jest on także backendem udostępniającym interfejs RMI dla www.

WEB

1. Typ: Wdrożenie
2. Lokalizacja w SVN:
 - binarki: <https://svn.avantis.pl/repos/dev/avantis/release/Services/mqube/bin/webapp>
 - źródła: <https://svn.avantis.pl/repos/dev/avantis/current/Services/mcube>
3. Adres URL: nazwa kontekstu mqube, dostępny tylko z wewnątrz
4. Sposób testowania: <http://secure.avantis.pl/mqube/login.xhtml> - strona z logowaniem
5. Project Manager: Anna Cygler
6. Developer: Ireneusz Koślacz
7. Analityk: Rafał Jaczyński
8. Czas działania: do odwołania
9. Opis usługi: Tymczasowy interfejs www do zarządzania usługą mqube. Dodatkowo udostępnia on interfejs REST dla www w PHP. Łączy się po RMI do serwisu mqube korzystając z adresu **n1.mqube.avantis.pl**.

Opis wdrożenia.

1. Utworzenie adresu do połączenia po RMI do serwisu: n1.mqube.avantis.pl (powinna wskazywać na serwer, gdzie zostanie uruchomiony serwis z pkt. 2)
2. Wdrożenie serwisu z svn (UWAGA - NIE URUCHAMIAĆ):
<https://svn.avantis.pl/repos/dev/avantis/release/Services/mqube/bin/service>
3. Wdrożenie aplikacji web z svn (TEŻ NIE URUCHAMIAĆ):
<https://svn.avantis.pl/repos/dev/avantis/release/Services/mqube/bin/webapp>
 - aplikacja powinna być dostępna tylko z wewnątrz.
4. Utworzenie routingu wychodzącego dla mqube. Będzie to jeden ogólny routing wychodzący dla mqube, modyfikowany w przypadku udostępnienia nowych numerów LA dla usługi mqube.
 - aktualnie routing ten powinien umożliwiać wysyłanie z LA: 7200 i 8061
5. Podłączenie utworzonego routingu do serwisu 150000 (mqube-manager)
6. Wyszukanie i poprawienie w pliku serwisu classes/mcube-service-ctx.xml property outRoutingId na id utworzonego routingu.
7. Uruchomienie serwisu
8. Uruchomienie aplikacji web

Task wdrożeniowy: #462869.

NAGIOS.

1. **mQube_jobs**
2. Sprawdzanie logów aplikacji mQube: /usr/local/logs/mpservices/services/mqube.log.
3. Sprawdzanie czy w przeciągu ostatnich 10 minut wystąpiły stringi:
 - a. Process send reminders task started
 - b. Process send inactivity reminders task started
 - c. Process IVR task started
 - d. Process cron task started##Generate results task started
4. Ważne: sprawdzanie powinno odbywać się dla każdego z tasków (w/w stringów) osobno.
5. Reakcja:
 - a. restart aplikacji
 - b. jeżeli nie pomogło - próba znalezienia błędu w logach
 - c. eskalacja do DRTU: Irek Koślacz

check_mqube_coty

- Check sprawdza dostępność stron:

<https://secure.avantis.pl/mqube/rest/drawingapp/get-drawing-tickets/44226250/2015-03-21T00:00:00+01/2015-03-21T23:59:00+01/false>

<https://secure.avantis.pl/mqube/rest/drawingapp/get-players/44211150/2015-03-23T00:00:00+01/2015-03-23T12:59:00+01/false/20>
- Sposób działania
Check sprawdza czy na powyższych strona otrzymamy kod "200", oraz czy wielkość ciała strony jest większa niż 0 bytes.
- Reakcja
Informacja do Karola Kordzińskiego

zypomnieniami, grupami, verbatimfiksami, użytkownikami i ich uprawnieniami, konfiguracją systemu, itp. Moduł ten jest także odpowiedzialny za integracje z MP w zakresie definiowania nowych mpserwisów i ich konfiguracji oraz modyfikacji routingów.

Moduł ten udostępnia interfejs pod adresem 5678/McubeClient. Podstawowym jego komponentem jest klasa **McubeClientImpl**, która implementuje interfejs **McubeClient**.

Jest to backend dla aplikacji web mqube.

Korzysta z CMSa do walidacji konfiguracji serwisu oraz tworzenia kanałów i produktów (10 i 11).

1. processing

Moduł odpowiedzialny za przetwarzania dostarczonych wiadomości z MP (1), potwierdzeń dostarczenia wiadomości wychodzących do bramki (2), potwierdzeń dostarczenia do użytkownika (3) oraz wysyłanie odpowiedzi zwrotnych zgodnie z logiką serwisu (12).

Mqube uruchamia mpserwisy, do których dostarczane są wiadomości MO i potwierdzenia dostarczenia wiadomości MT. Mpserwis jest uruchamiany dla grupy oraz LA. Mpserwis ma identyfikator z zakresu od 150001 do 159999, nazwę i opis odpowiednio w formacie

mqube-0-2

i

mqube-1-2

, gdzie 0 to id grupy w serwisie mqube, 1 nazwa grupy i 2 numer LA.

Mqube modyfikuje routingu przychodzące kierując ruch na odpowiednie mpserwisy. Element routingu utworzony automatycznie przez serwis mqube jest oznaczony komentarzem "MCUBE managed routing".

Na potrzeby serwisu mqube został utworzony routing wychodzący o id 150000. Mpserwisy są automatycznie bindowane z tym routingiem, ale w celu udostępnienia nowego numeru LA w mqube wymagana jest ręczna modyfikacja tego routingu.

Poszczególne serwisy w ramach mqube mogą działać na różnych numerach LA, które zostały skonfigurowane i przydzielone użytkownikom.

Mpserwis 150000 mqube-manager nie bierze udziału w przetwarzaniu wiadomości MO i jest on wykorzystywany do alertowania błędów oraz wysyłania przypomnień.

1. ivr

Moduł odpowiedzialny za integrację z IVR.

Niektóre serwisy uruchomione w ramach mqube wymagają integracji z IVR. Komunikacja z IVR zrealizowana jest za pośrednictwem bazy danych, tzn. wybierane i przetwarzane są dane z tabel **ivr.biling** i **ivr_serwisy.ivr_odpowiedzi**.

Przetwarzane są odpowiedzi na pytnia oraz informacja o zakończeniu połączenia. Sposób przetwarzania jest zależny od logiki serwisu.

Za integrację z IVR odpowiedzialny jest komponent **ProcessIvrTask**, uruchamiany jako oddzielny proces z konfigurowalnym interwałem (domyślnie 30 sek.) przez komponent **IvrController**.

Błędy alertowane są ze stanem **PROCESS_IVR_TASK**.

Logika serwisu IVR jest niezależna od mqube i odpowiedzialny za nią jest developer usług IVR.

1. reminder

Moduł odpowiedzialny za obsługę przypominaczy i naganiaczy (przypomnień po nieaktywności).

Za obsługę przypominaczy odpowiedzialne są komponenty: **StartReminderTask**, który wyciąga i wstawia bazę odbiorców przypomnienia, uruchamiany jako oddzielny wątek z konfigurowalnym interwałem oraz **SendRemindersTask**, który realizuje wysyłkę, uruchamiany jako oddzielny wątek z konfigurowalnym interwałem. Wysyłka odbywa się w sposób podobny do PBM, czyli badane są kolejki na brankach i proces wysyłający dopełnia kolejkę do określonego poziomu.

Moduł ten i komponent **SendRemindersTask** odpowiada również za wysyłkę naganiaczy.

Przypominacze wysyłane są z mpserwisu 150000 - mqube-manager.

Błędy alertowane są ze stanem **START_REMINDERS_TASK** i **SEND_REMINDERS_TASK** odpowiednio dla startowania i wysyłania przypominaczy.

1. iplay

Moduł odpowiedzialny za integrację z IPLAY. Umożliwia rejestrację i przedzielanie kodów do rejestracji w iplay.pl. Główne komponenty wchodzące w skład tego modułu to **IplayRegistrationManager**, **IplayClient** oraz **RetryUnprocessedTask**, który jest uruchamiany cyklicznie z konfigurowalnym interwałem (domyślnie 15 minut) i ponawia nieudane próby rejestracji uczestników.

Błędy tego procesu alertowane są ze stanem **IPLAY_RETRY_TASK**.

Do komunikacji z IPLAY wykorzystywany jest protokół HTTP (4).

Komunikacja z iplay może zostać zapoczątkowana w wyniku różnych zdarzeń, ale typowo w wyniku przetworzenia określonej wiadomości sms lub przez cykliczny proces ponawiający nieudane próby.

1. mtsp

Moduł odpowiedzialny za integrację z mtsp. Umożliwia on serwisom otrzymywanie notyfikacji o rejestracjach do subskrypcji mt i obciążaniach, a także definiowanie subskrypcji.

Komunikacja z mtsp odbywa się za pomocą RMI (notyfikacje 6 i definiowanie subskrypcji 5).

1. pii (Personally Identifiable Information)

Moduł odpowiedzialny za zbieranie danych osobowych od zwyczajców serwisów. Wysyła on prośby o przesłanie danych osobowych, przypomnienia oraz przetwarza smsy z danymi osobowymi zgodnie z konfiguracją. Główne elementy tego modułu to **PiiProcessor** i **ProcessPiiTask**. Sprawdzanie, czy są jakieś serwisy dla których należy wysłać prośby lub przypomnienia odbywa się cyklicznie z konfigurowalnym interwałem (domyślnie kilka minut).

mqube:webapp

Aplikacja znajduje się pod adresem <https://secure.avantis.pl/mqube/>.

Korzysta z backendu, do którego łączy się po rmi: 5678/MqubeClient (7)

Logi aplikacji w pliku

```
`${LOG_PATH}/mqube.log
```

Pełni ona kilka funkcji:

- jest tymczasowym interfejsem użytkownika do zarządzania serwisem mqube (docelowym jest interfejs PHP)
- udostępnia interfejs REST pod adresem <https://secure.avantis.pl/mqube/rest>, który także służy do zarządzania serwisem i jest przeznaczony dla interfejsu PHP.

Korzysta z webcmsa do pobierania wyświetlanych na stronie komunikatów i treści (9).

Zgłoszenia WWW:

Mqube udostępnia interfejs REST przy pomocy którego uczestnicy mogą brać udział w zdefiniowanych w mqube serwisach.

Interfejs wymaga autoryzacji i mogą z niego korzystać partnerzy, którzy mają dostęp do mqube oraz posiadają uprawnienie WEB_REG.

Autoryzacja jest typu Basic Authorization.

Metoda do web rejestracji znajduje się pod adresem

`mqube/rest/webreg/webreg/{serviceld}/`, gdzie parametr `serviceld` jest identyfikatorem serwisu mqube. Metoda ta przyjmuje wywołania typu POST zawierające xml w następującym formacie:

```
<pl.avantis.mcube.api.WebRegistrationDTO

  <msisdn></msisdn>

  <operatorId></operatorId>

  <email></email>

  <score></score>

  <sessionId></sessionId>

  <properties

    <entry>

      <string></string>

      <string></string>

    </entry>

  </properties>

</pl.avantis.mcube.api.WebRegistrationDTO>
```

`msisdn` - numer telefonu uczestnika w formacie z prefiksem kraju

`operatorId` - opcjonalny identyfikator operatora

`score` - ilość punktów przyznawana uczestnikowi za zgłoszenie

`sessionId` - opcjonalny, unikalny identyfikator transakcji systemu partnera

`properties` - mapa parametrów typu klucz wartość do przekazania specyficznych dla serwisu parametrów np. kod, hasło konkursowe, itp.

W przypadku poprawnego żądania metoda odpowiada statusem 200 i dokumentem xml w formacie:

```
<pl.avantis.mcube.api.WebRegistrationResponseDTO  
  
  <responseCode></responseCode>  
  
  <responseMessage></responseMessage>  
  
  <processed></processed>  
  
  <transactionId></transactionId>  
  
</pl.avantis.mcube.api.WebRegistrationResponseDTO>
```

responseCode - kod opisujący sposób przetworzenia zgłoszenia, specyficzny dla serwisu

responseMessage - treść do wyświetlenia uczestnikowi, jeżeli zarządzanie komunikacją jest po stronie mqube

processed - true, gdy żądanie zostało przetworzone przez serwis lub false, gdy serwis odrzucił żądanie zgłoszenia (np. przed lub po konkursie, użytkownik nie uprawniony)

transactionId - unikalny identyfikator pod jakim zostało zarejestrowane zgłoszenie

W przypadku błędnego żądania metoda odpowiada statusem 500, np.

wewnętrzny błąd, błąd autoryzacji, brak uprawnień, niepoprawny format zgłoszenia, np. nieistniejący serwis, błędny format xml

Przykładowe zgłoszenie do serwisu typu B2bCodes:

Request:

POST http://beta2:9080/mqube/rest/webreg/webreg/43857550/

POST data:

```
<pl.avantis.mcube.api.WebRegistrationDTO  
  
  <msisdn>48600000001</msisdn>  
  
  <operatorId>26002</operatorId>  
  
  <email>ireneusz.koslacz@avantis.pl</email>  
  
  <score>10</score>  
  
  <sessionId>123abc</sessionId>  
  
  <properties  
  
    <entry>  
  
      <string>ADD_MESSAGE</string>  
  
      <string>Test web registration</string>  
  
    </entry>  
  
    <entry>  
  
      <string>CODE</string>  
  
      <string>TEST001</string>  
  
    </entry>  
  
  </properties>  
  
</pl.avantis.mcube.api.WebRegistrationDTO>
```

[no cookies]

Request Headers:

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Authorization: Basic aatvc2xhY3oaVWNuNbQxOTI=

Content-Type: application/xml

Content-Length: 547

Host: beta2:9080

User-Agent: Apache-HttpClient/4.2.6 (java 1.5)

Response:

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 264

```
<pl.avantis.mcube.api.WebRegistrationResponseDTO id="1">

  <responseCode>CORRECT</responseCode>

  <responseMessage>CORRECT</responseMessage>

  <processed>true</processed>

  <transactionId>973853000</transactionId>

</pl.avantis.mcube.api.WebRegistrationResponseDTO>
```

PHP user interface:

Interfejs użytkownika do zarządzania serwisem w PHP. Korzysta z interfejsu REST pod adresem <https://secure.avantis.pl/mqube/rest> (8) Developerem tego interfejsu jest Marcin Federowicz.

Weryfikacja działania usługi:

UWAGA: dodatkowo w stosunku do poniższego opisu włączone może być logowanie zapytań sql na poziomie DEBUG z klasą `org.hibernate.SQL` (w chwili pisania tego opisu jest).

mqube:service - błędy

Jeżeli wystąpi błąd w przetwarzaniu wiadomości, cyklicznie uruchamianym zadaniu lub wywołaniu metody interfejsu **MqubeClient** jest on logowany z poziomem ERROR. Informacje o rodzaju błędu, jego przyczynie i miejscu wystąpienia niesie komunikat błędu oraz stack trace.

mqube:service - ostrzeżenia

Logowane w obsłudze wywołań metod interfejsu **MqubeClient**, gdy metodę próbuje wykonać użytkownik bez wymaganych uprawnień **AccessDeniedException** lub gdy metoda została wywołana z błędnymi parametrami **InvalidParameterException**. Komunikat wyjątku niesie dodatkowe informacje.

```
2010-11-26 14:09:46,380 | WARN | service.client.McubeClientImpl | InvalidParameterExcep
```

mqube:service - informacje

Żeby umożliwić sprawdzenie, czy wiadomości są dostarczane i przetwarzane, czy uruchamiane są cykliczne procesy oraz czy wywoływane są metody interfejsu **MqubeClient**, na poziomie INFO logowanie są następujące informacje:

- Przetwarzanie wiadomości

Przebieg procesu przetwarzania wiadomości od jego rozpoczęcia (Message delivered - processing) do zakończenia (Total processing time: 104 ms).

Z istotnych informacji widać także jaka jest przetwarzana wiadomość (Processing message: ...) oraz jaki serwis tą wiadomość przetwarza (Processing with service: 10 050 (BJBMW3))

```

2010-11-26 11:48:49,039 | [13542131648] | INFO | mcube.mp.McubeMPService | Messa
2010-11-26 11:48:49,039 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Proc
2010-11-26 11:48:49,039 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Searc
2010-11-26 11:48:49,040 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Found
2010-11-26 11:48:49,040 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Filte
2010-11-26 11:48:49,040 | [13542131648] | DEBUG | service.core.ServiceDispatcherImpl | Match
2010-11-26 11:48:49,042 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | 1 ser
2010-11-26 11:48:49,042 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Sorti
2010-11-26 11:48:49,042 | [13542131648] | DEBUG | service.core.ServiceDispatcherImpl | Servi
2010-11-26 11:48:49,042 | [13542131648] | INFO | service.core.ServiceProcessorImpl | Proce
2010-11-26 11:48:49,071 | [13542131648] | INFO | service.core.ServiceProcessorImpl | Messa
2010-11-26 11:48:49,075 | [13542131648] | INFO | mp.service.MPMessageExchangeContextImpl | mp-me
2010-11-26 11:48:49,143 | [13542131648] | INFO | service.core.ServiceDispatcherImpl | Total

```

- Przetwarzanie potwierdzenia dostarczenia

Rozpoczęcie i zakończenie przetwarzania.

```

2010-11-26 12:20:39,558 | INFO | mcube.mp.McubeMPService | Message acknowledged
2010-11-26 12:20:39,576 | INFO | service.core.ServiceDispatcherImpl | Total processing time

```

- Przetwarzanie potwierdzenia dostarczenia do użytkownika

Rozpoczęcie i zakończenie przetwarzania.

```

2011-04-20 14:53:21,249 | [15647713185] | INFO | mcube.mp.McubeMPService | User
2011-04-20 14:53:21,422 | [15647713185] | INFO | mcube.mp.McubeMPService | User

```

- Startowanie przypomnaczy

Uruchomienie i zakończenie zadania. Informacja jakie przypomnaczy zostają wystartowane.

```

2010-11-26 12:42:23,650 | INFO | service.reminder.StartRemindersTask | Process start reminde
2010-11-26 12:42:23,684 | INFO | service.reminder.ReminderStarterImpl | Starting reminder: Re
2010-11-26 12:42:23,754 | INFO | service.reminder.StartRemindersTask | Process start reminde

```

- Wysyłanie przypomnaczy

Uruchomienie i zakończenie zadania. Przebieg procesu wysłania: pobieranie kolejki na usłudze, pobieranie paczki odbiorców, tworzenie wiadomości, wysyłanie wiadomości.

```

2010-11-26 11:48:35,862 | INFO | service.reminder.SendRemindersTask | Process send reminder
2010-11-26 12:42:24,417 | DEBUG | service.reminder.SendRemindersTask | Getting JMS queue siz
2010-11-26 12:42:24,726 | DEBUG | service.reminder.SendRemindersTask | Could not get JMS que
2010-11-26 12:42:24,726 | DEBUG | service.reminder.SendRemindersTask | Getting JMS queue siz
2010-11-26 12:42:24,730 | DEBUG | service.reminder.ReminderSenderImpl | Getting reminder reci
2010-11-26 12:42:24,859 | DEBUG | service.reminder.ReminderSenderImpl | Creating reminder mes
2010-11-26 12:42:24,903 | DEBUG | service.reminder.ReminderSenderImpl | Sending reminder mess
2010-11-26 12:42:26,394 | DEBUG | service.reminder.SendRemindersTask | Waiting for send remi
2010-11-26 11:48:35,868 | INFO | service.reminder.SendRemindersTask | Process send reminder

```

- Wysyłanie naganiaczy (przypomnień po nieaktywności)

Uruchomienie i zakończenie zadania. Informacja które przypomnienie jest wysyłane oraz do jakiej liczby uczestników.


```

2010-11-26 12:35:08,938 | INFO | service.reminder.SendRemindersTask | Process send inactivi
2010-11-26 12:35:08,943 | DEBUG | service.reminder.SendRemindersTask | Waiting for send inac
2010-11-26 12:35:08,949 | INFO | service.reminder.ReminderSenderImpl | Sending 60 min. inacti
2010-11-26 12:35:09,109 | INFO | service.reminder.ReminderSenderImpl | Sending 5 min. inacti
2010-11-26 12:35:09,212 | INFO | service.reminder.SendRemindersTask | Process send inactivi

```

- Obsługa połączeń i odpowiedzi IVR

Uruchomienie i zakończenie zadania.

```

2010-11-26 14:05:05,510 | INFO | service.ivr.ProcessIvrTask | Process IVR task star
2010-11-26 14:05:05,547 | INFO | service.ivr.ProcessIvrTask | Process IVR task succ

```

- Obsługa integracji z MTSP

Notyfikacja o subskrypcji, w której jest informacja o numerze msisdn, statusie subskrypcji, nazwie serwisu.

```

2011-04-20 15:23:17,189 | [RMI TCP Connection(766)-192.168.2.1] | INFO | service.client.McubeMTCL
2011-04-20 15:23:17,189 | [RMI TCP Connection(766)-192.168.2.1] | INFO | service.mt.MtServiceDisp
2011-04-20 15:23:17,194 | [RMI TCP Connection(766)-192.168.2.1] | INFO | service.mt.MtServiceProc
2011-04-20 15:23:17,319 | [RMI TCP Connection(766)-192.168.2.1] | INFO | service.mt.MtServiceProc

```

Notyfikacja o obciążeniu, zawierająca te same informacje.

```

2011-04-20 15:24:17,346 | [RMI TCP Connection(769)-192.168.2.1] | INFO | service.mt.MtServiceDisp
2011-04-20 15:24:17,346 | [RMI TCP Connection(769)-192.168.2.1] | INFO | service.mt.MtServiceDisp
2011-04-20 15:24:17,352 | [RMI TCP Connection(769)-192.168.2.1] | INFO | service.mt.MtServiceProc
2011-04-20 15:24:18,541 | [pool-19-thread-1] | INFO | service.ivr.ProcessIvrTask |
2011-04-20 15:24:20,132 | [RMI TCP Connection(769)-192.168.2.1] | INFO | type.lider.LiderService
2011-04-20 15:24:20,141 | [RMI TCP Connection(769)-192.168.2.1] | INFO | service.mt.MtServiceProc

```

- Przetwarzanie wywołań interfejsu McubeClient

Wywołanie metody interfejsu oraz jego zakończenie. Log wywołania ma format invoke NAZWA_METODY start, natomiast zakończenia

```
invoke [NAZWA_METODY]end in {0} ms
```

, gdzie NAZWA_METODY to nazwa wywoływanej metody. Dodatkowo w większości przypadków logowane są parametry wywołania oraz zwracane rezultaty. Jednym z parametrów metod jest login użytkownika mqube.

```

2011-04-20 15:08:18,202 | [RMI TCP Connection(721)-192.168.2.1] | INFO | mcube.logging.LoggingInt
2011-04-20 15:08:18,647 | [RMI TCP Connection(721)-192.168.2.1] | INFO | mcube.logging.LoggingInt

```

- Rejestracje w iplay

Wysłanie żądania przydzielenia kodu do music startera dla numeru msisdn:

```
2011-04-20 14:34:17,919 | [pool-2-thread-1] | INFO | service.ipplay.IplayClientImpl | I
```

Wysłanie żądania rejestracji numeru msisdn:

```
2011-04-20 14:34:17,919 | [pool-2-thread-1] | INFO | service.ipplay.IplayClientImpl | I
```

- Cron

Wyszukiwanie serwisów do wykonania zadań typu cron oraz ich uruchamianie:

```
2011-04-20 15:45:51,440 | [pool-20-thread-1] | INFO | service.core.ProcessCronTask |
2011-04-20 15:45:51,510 | [pool-20-thread-1] | INFO | service.core.ProcessCronTask |
...
ewentualne logi wykonywania zadania, jeżeli logika serwisu danego typu jakieś zapisuje
...
2011-04-20 15:45:51,615 | [pool-20-thread-1] | INFO | service.core.ProcessCronTask |
```

- pii (Personally Identifiable Information)

Przetwarzanie danych osobowych:

```
2011-05-20 14:36:00,625 | [10000000628266176] | INFO | mcube.mp.McubeMPService |
2011-05-20 14:36:00,628 | [10000000628266176] | INFO | service.core.ServiceDispatcherImpl |
2011-05-20 14:36:00,654 | [10000000628266176] | INFO | service.core.ServiceDispatcherImpl |
2011-05-20 14:36:00,697 | [10000000628266176] | INFO | service.core.ServiceDispatcherImpl |
2011-05-20 14:36:00,702 | [10000000628266176] | INFO | service.core.ServiceDispatcherImpl |
2011-05-20 14:36:00,702 | [10000000628266176] | INFO | service.core.ServiceDispatcherImpl |
```

Wysyłka prób o przysłanie danych osobowych - przykładowo dla czterech zwycięzców:

```
2011-06-03 15:44:38,047 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
2011-06-03 15:44:38,116 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:44:38,129 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:44:38,142 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:44:38,143 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:44:38,143 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:44:38,400 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
```

Automatyczna kontynuacja zbierania danych od osób rezerwowych w przypadku, gdy zwycięzcy nie przysłali danych w wyznaczonym terminie:

```
2011-06-03 15:48:38,632 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
2011-06-03 15:48:38,697 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:48:38,698 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:48:38,698 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 15:48:38,698 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
...
wysyłanie prób do rezerwowych jak w przykładzie wyżej
...
2011-06-03 15:48:38,739 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
```

Wysyłka przypomnień o przysłanie danych osobowych - przykładowo dla dwóch zwycięzców:

```
2011-06-03 16:03:16,096 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
2011-06-03 16:03:16,123 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 16:03:16,125 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 16:03:16,126 | [pool-49-thread-1] | INFO | service.pii.PiiProcessorImpl |
2011-06-03 16:03:16,229 | [pool-49-thread-1] | INFO | service.pii.ProcessPiiTask |
```

mqube:webapp

Ponieważ aplikacja web jest wykorzystywana tylko w celu wystawienia interfejsu REST dla aplikacji php nie loguje ona w zasadzie nic pożytecznego w trybie INFO. Widać tam natomiast ewentualne problemy z połączeniem RMI do backendu:

```
2011-05-18 19:56:20,030 | ERROR | common.filter.CatchLogFilter | *** ERROR *** WRAPPER  
org.jboss.resteasy.spi.UnhandledException: org.springframework.remoting.RemoteLookupFailureExcepti
```

TVP api

Api stworzone wyłącznie na potrzeby obsługi serwisów dedykowanych telewizji TVP. Dla TVP jest również stworzony osobny dedykowany panel. Dostęp do panelu wymaga konta i uprawnień w AD jak i Mqube. Możliwe jest też wykorzystanie przez panel dedykowanego konta tvp2018.

Dokładniejszy opis serwisów: TVP serwisy SMS [CRM-1551]

Adres produkcyjny do api: <http://tvp-api.mqube.pl/rest/tvp/>

Adres produkcyjny do panelu TVP: <https://panel-tvp.digitalvirgo.pl/>

Mqube.pl - frontend

Monitoring

Check HTTPS_mqube.avantis.pl

Sprawdza czy strona www.mqube.pl zwraca frazę "mqube.pl"

Reakcja:

1. Sprawdzenie działania strony <https://mqube.pl>
2. Sprawdzenie działania strony interfejsu rest <https://secure.avantis.pl/rest/login>
3. Sprawdzenie działania apacha na n1.c1.zend.
4. W przypadku gdy apache na n1.c1.zend działa poprawnie, a aplikacja nie działa esklować do Marcina Federowicza (poza godzinami pracy do Wojka Koszuta)

Check [0]Av SSRV mQube / main page login

Check dotcom monitora - sprawdza czy może się zalogować do strony mqube.pl

Reakcja:

1. Sprawdzenie działania strony <https://mqube.pl>
2. Sprawdzenie działania strony interfejsu rest <https://secure.avantis.pl/rest/login>
3. Sprawdzenie działania apacha na n1.c1.zend.
4. W przypadku gdy apache na n1.c1.zend działa poprawnie, a aplikacja nie działa esklować do Marcina Federowicza (poza godzinami pracy do Wojka Koszuta)

Check [0]Av SSRV mQube / main page

Check dotcom monitora - sprawdza czy wyświetla się strona główna serwisu

Reakcja:

1. Sprawdzenie działania strony <https://mqube.pl>
2. Sprawdzenie działania strony interfejsu rest <https://secure.avantis.pl/rest/login>
3. Sprawdzenie działania apacha na n1.c1.zend.
4. W przypadku gdy apache na n1.c1.zend działa poprawnie, a aplikacja nie działa esklować do Marcina Federowicza (poza godzinami pracy do Wojka Koszuta)

Opis serwerów , usług i komponentów

Lokalizacja na serwerze testowym.

Ścieżka do aplikacji: /var/www/html/testphp/mqube/app_repo

Zasoby wykorzystywane przez aplikacje.

serwer:beta2

Interfejs rest url: http://beta2:9080/mqube/rest

url: http://mqube-test.avantis.pl

git: ssh://git@git.services.avantis.pl:7999/mq/mqube-frontend.git

logi: /var/www/html/testphp/mqube/app_repo/application/logs/application.log - logi aplikacji , mqu

Lokalizacja na serwerze produkcyjnym.

Ścieżka do aplikacji: /var/www/html/mqube-ui

Zasoby wykorzystywane przez aplikacje.

serwer:n1.c1.zend

Interfejs rest url: https://secure.avantis.pl/mqube/rest

url: https://mqube.pl

svn:https://svn.avantis.pl/repos/dev/avantis/release/WebApps/mqube-ui

logi: /var/www/html/mqube-ui/application/logs/application.log - logi aplikacji , mqubeldap.log - p

Dostępne komponenty

Aplikacja frontendowa dla mqube.pl napisana jest w php z użyciem zend framework. Konstrukcja aplikacji oparta jest o controllery. W zależności od wybranej sekcji dostępne są dedykowane controllery i tak np dla Reminderów są oddzielne controllery , a dla funkcjonalności związanych z tworzeniem i edycją serwisów są inne controllery. Dzięki takiej strukturze łatwiej znaleźć interesujący nas kawałek kody i można np wdrażać poprawki do danej funkcjonalności niezależnie od innych controllerów. Wszystkie controllery dziedziczą z jednego nadrzędnego controllera jakim jest AVcontroller, dzięki czemu można dodać do aplikacji pewne metody w jednym miejscu, a dostępne będą one w pozostałych.

Account

Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- konfiguracji grup - lista dostępnych grup, dodawanie, edycja
- konfiguracji użytkowników - lista dostępnych użytkowników, dodawanie, edycja, deaktywacja

Wykorzystywane metody REST

Grupy:

Lista grup - GET

http://beta2:9080/mqube/rest/group/

Dodawanie grupy - POST

http://beta2:9080/mqube/rest/group

Modyfikacja grupy - PUT

http://beta2:9080/mqube/rest/group

Użytkownicy:

Lista userów - GET

http://beta2:9080/mqube/rest/user/

Dodawanie usera - POST

http://beta2:9080/mqube/rest/user

Modyfikacja usera - PUT

http://beta2:9080/mqube/rest/user

Pobieranie wybranego usera - GET

http://beta2:9080/mqube/rest/user/{id}

Deaktywacja wybranego usera - DELETE

http://beta2:9080/mqube/rest/user/{id}

LA/verbatimfixes

Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- konfiguracji numerów LA - lista dostępnych numerów, dodawanie, edycja
- konfiguracji verbatimfixów - lista dostępnych verbatimfixów z informacją czy dany verbatimfix jest przypisany do wybranej grupy, dodawanie verbatimfixów do wybranej grupy, edycja, usuwanie verbatimfixów

Wykorzystywane metody REST

Dodawanie LA - POST

`http://beta2:9080/mqube/rest/la`

Edycja LA - PUT

`http://beta2:9080/mqube/rest/la`

Dodawanie verbatimfixu - POST

`http://beta2:9080/mqube/rest/verbatimfix`

Usunięcie verbatimfixu - DELETE

`http://beta2:9080/mqube/rest/verbatimfix/{id}`

Services

Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy serwisów dostępnych dla użytkownika
- * tworzenie nowego serwisu od podstaw lub też kopiowanie już istniejącego

Wykorzystywane metody REST

Lista serwisów - GET

`http://beta2:9080/mqube/rest/service/{grouplist}/-?from=2011-11-05T00:00:00+01:00&to=2013-07-05T00`

Parametry:

- * {grouplist} - lista id grup do których należą serwisy które chcemy pobrać, poprzedzielane znakiem
- * {from} - przedział czasowy od
- * {to} - przedział czasowy do

Pobieranie konfiguracji wybranego serwisu - GET

`http://beta2:9080/mqube/rest/service/{id}`

Parametry:

- * {id} - id serwisu

Tworzenie nowego serwisu - POST

`http://beta2:9080/mqube/rest/service-new/{service-type}`

Parametry:

* {service-type} - typ serwisu jaki chcemy utworzyc, dostępne typy to: REGISTER, BJBW, HIT, SALAR

Tworzenie nowego serwisu na podstawie już istniejącego - kopiowanie - POST

http://beta2:9080/mqube/rest/service/copy/{id}/{skipstages}

Parametry:

- * {id} - id serwisu na podstawie którego chcemy utworzyć nową kopię serwisu
- * {skipstages} - wartość **boolean** - **true**, **false** - oznacza czy chcemy kopiować etapy do nowego serwisu

Reports

Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- listy serwisów dla których dostępne są raporty - przypisane użytkownikowi - możliwość filtrowania po grupie
- wybrane raporty - przeglądanie raportu, verbatimizacja danych w formie tabeli oraz wykresów kołowych i słupkowych w zależności od konfiguracji
- możliwość exportu danych z raportu w postaci plików csv

Wykorzystywane metody REST

Pobranie listy raportów dostępnych dla danego typu serwisu - GET

http://beta2:9080/mqube/rest/report-types/{service-type}

Parametry:

* {service-type} - typ serwisu jaki chcemy utworzyc, dostępne typy to: REGISTER, BJBW, HIT, SALAR

Reminders

Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- listy reminderów - możliwość filtrowania po typie remindera: czy do white listy czy live
- tworzenie reminderów
- edycja reminderów
- export reminderów

Wykorzystywane metody REST

Pobieranie listy reminderów - GET

http://beta2:9080/mqube/rest/reminder/{id}/{datefrom}/{dateto}

Parametry:

- {id} - id serwisu dla którego chcemy pobrać listę reminderów np: 199851
- {datefrom} - data od np : 2011-08-30T00:00:00+02:00
- {dateto} - data do np : 2012-09-13T00:00:00+02:00

Pobieranie remindera - GET

http://beta2:9080/mqube/rest/reminder/{id}

Parametry:

- {id} - id remindera którego konfiguracje chcemy pobrać np 2550316019600

Modyfikacja remindera - PUT

<http://beta2:9080/mqube/rest/reminder>

Tworzenie remindera - POST

<http://beta2:9080/mqube/rest/reminder>

---++++ Automatic Reminders

---+++++ Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy automatic reminderow - standardowych możliwość filtrowania po przedziale czasowym
- * listy automatic reminderow - zaawansowanych możliwość filtrowania po przedziale czasowym
- * tworzenie automatic reminderow - standardowych
- * tworzenie automatic reminderow - zaawansowanych
- * edycja automatic reminderów standardowych
- * export automatic reminderów zaawansowanych

---+++++ Wykorzystywane metody REST

Pobieranie listy automatic reminderów - standardowych , odbywa się poprzez przetworzenie xmla

<http://beta2:9080/mqube/rest/service/{id}>

Parametry:

- * {id} - id serwisu którego konfiguracje chcemy pobrać np 199851

Pobierany jest fragment który znajduje się w tagu : inactivityReminders np:

```
300
C1
false
false
2550316002600
600
CODE1
false
false
2550315986200
```

Pobieranie listy automatic reminderów zaawansowanych - GET:

<http://beta2:9080/mqube/rest/advanced-inactive-reminders/{id}/{datefrom}/{dateto}>

Parametry:

- * {id} - id serwisu którego konfiguracje chcemy pobrać np 199851
- {datefrom}
 - data od - ustawianie przedziału czasowego
- {dateto} - data do - ustawianie przedziału czasowego

Tworzenie automatic remindera standardowego - POST

`http://beta2:9080/mqube/rest/inactivity-reminder`

Tworzenie automatic remindera zaawansowanego - POST

`http://beta2:9080/mqube/rest/reminder`

Wysyłamy podobny obiekt xml co w przypadku Normalnego Remindera , z różnicą dwóch flag:

```
advancedInactive = "true";  
inactivityTimeSec = 123;
```

Pobieranie automatic remindera standardowego - pobieramy konfigurację wybranego serwisu w którym jest on zdefiniowany.
Następnie przesyłamy gotowy obiekt xml do metody:- PUT

`http://beta2:9080/mqube/rest/inactivity-reminder`

Pobieranie automatic remindera zaawansowanego - GET

`http://beta2:9080/mqube/rest/reminder/{id}`

Parametry:

* {id} - id automatic remindera zaawansowanego którego konfigurację chcemy pobrać np 2550316019801

Modyfikacja automatic remindera zaawansowanego - PUT

`http://beta2:9080/mqube/rest/reminder`

Export automatic reminderów zaawansowanych - polega na wyciągnięciu listy automatic reminderów

----- Inquiry Reminders
----- Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy inquiry reminderów
- * tworzenie, edycja, usuwanie inquiry reminderów

----- Wykorzystywane metody REST

Pobieranie listy inquiry reminderów - GET

`http://beta2:9080/mqube/rest/inquiry-reminders/{id}`

Parametry:

* {id} - id serwisu dla którego chcemy pobrać listę inquiry reminderów np 199851

Tworzenie nowych inquiry reminderów - POST

`http://beta2:9080/mqube/rest/inquiry-reminder/{id}`

Parametry:

* {id} - id serwisu dla którego chcemy utworzyć inquiry reminder np 199851

Jako body wysyłamy xml z przykładową konfiguracją inquiry reminder:

```
CITY
SHARED
false
true
false
7555
Europe/Warsaw
false
ADD_SCORE
GUESS_WRONG
city:
```

Pobieranie wybranego inquiry reminder - GET

`http://beta2:9080/mqube/rest/inquiry-reminder/{id}`

Parametry:

* {id} - id inquiry reminder np 5174052

Modyfikacja wybranego inquiry reminder - PUT

`http://beta2:9080/mqube/rest/inquiry-reminder/{id}`

Parametry:

* {id} - id serwisu w którym znajduje się modyfikowany przez nas inquiry reminder

Usuwanie inquiry reminder - DELETE

`http://beta2:9080/mqube/rest/inquiry-reminder/{serviceld}/{id}`

Parametry:

- * {serviceld} - id serwisu w którym znajduje się usuwany przez nas inquiry reminder
- {id}
 - id usuwanego przez nas inquiry reminder

---++++ Bonuses

---+++++ Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy bonusów - możliwość ustawienia przedziału czasowego
- * tworzenie, edycja bonusów

---+++++ Wykorzystywane metody REST

<http://beta2:9080/mqube/rest/bonus/{id}/{datefrom}/{dateto}>

Parametry:

- * {id} - id serwisu dla którego chcemy pobrać listę bonusów np 199851
- {datefrom} - data od - ustawianie przedziału czasowego
- {dateto} - data do - ustawianie przedziału czasowego

Tworzenie nowych bonusów - POST

<http://beta2:9080/mqube/rest/bonus>

Jako body wysyłamy xml z przykładową konfiguracją bonusu:

```
SALARY
199851
2012-09-18 10:52:00 CEST
2012-09-18 11:52:00 CEST
1
abc
abc
```

Pobieranie wybranego bonusu - GET

<http://beta2:9080/mqube/rest/bonus/{id}>

Parametry:

- * {id} - id bonus np 2550315986951

Modyfikacja wybranego bonus - PUT

<http://beta2:9080/mqube/rest/bonus>

Jako body podajemy zmodyfikowaną konfigurację bonusu.

```
---+++++ Mechanics
---+++++ Opis ogólny
```

Z poziomu tej zakładki mamy dostęp do:

- * listy treści wykorzystywanych w mechanice
- * tworzenie, edycja, usuwanie treści w mechanice

---+++++ Wykorzystywane metody REST

Pobieranie listy treści dostępnych w mechanice serwisu - GET

Aby pobrać listę treści w mechanice wybranego serwisu należy pobrać zawartość listy messages

```
GUESS_WRONG
GUESS_WRONG
false
0
0
true
0
0
2550316021700
```

2012-07-31 09:55:20.0
2550316021650
true
500
199851

http://beta2:9080/mqube/rest/service/{id}

Parametry:

* {id} - id serwisu dla którego chcemy pobrać konfigurację np 199851

Tworzenie nowych treści mechaniki - POST

http://beta2:9080/mqube/rest/message

Jako body wysyłamy xml z przykładową konfiguracją treści do mechaniki:

GUESS_WRONG
xxx
199851
false
false
2012-09-17 11:17:21.0 CEST
0
0

Pobieranie wybranej treści mechaniki - GET

http://beta2:9080/mqube/rest/service/{id}

Parametry:

* {id} - id serwisu dla którego chcemy pobrać konfigurację

Wykonujemy pętlę po wszystkich treściach w konfiguracji i wybieramy sobie właściwą która nas

Modyfikacja wybranej treści mechaniki - PUT

http://beta2:9080/mqube/rest/message

Jako body podajemy zmodyfikowaną konfigurację mechaniki.

GUESS_WRONG
xxx
199851
false
false
0
0

Usuwanie wybranej treści mechaniki - PUT

Aby usunąć wybraną treść z mechaniki należy pobrać konfigurację serwisu usunąć z konfiguracji

<http://beta2:9080/mqube/rest/service>

Jako body podajemy całą konfigurację w postaci xml

```
----- Whitelist
----- Opis ogólny
```

Z poziomu tej zakładki mamy dostęp do:

- * listy numerów msisdn dostępnych w whitelist
- * tworzenie, usuwanie numerów msisdn na whitelist
- * wyszukiwanie numerów dostępnych na whitelist

----- Wykorzystywane metody REST

Pobieranie listy dostępnych numerów msisdn na whitelist

W tym celu pobieramy konfigurację serwisu, z niej wyciągamy zawartość tagu whitelist, który z

```
2011-05-25 02:00:00.0
2011-08-25 02:00:00.0
2012-08-31 11:30:00.0
2011-08-26 02:00:00.0
2012-08-30 11:30:00.0
false
true
48509756730
48695713086
KD
48506047148
```

Usuwanie numeru z whitelist

W tym celu pobieramy całą listę robimy iterację po wszystkich elementach jeśli znajdziemy ten

```
----- Blacklist
----- Opis ogólny
```

Z poziomu tej zakładki mamy dostęp do:

- * listy numerów msisdn dostępnych w blacklist
- * tworzenie, usuwanie numerów msisdn na blacklist
- * wyszukiwanie numerów dostępnych na blacklist

----- Wykorzystywane metody REST

Pobieranie licznika elementów listy dostępnych numerów msisdn na blacklist - POST

Z blacklistą jest troszkę bardziej złożona sprawa, ponieważ na liście tej przeważnie znajduje

<http://beta2:9080/mqube/rest/blacklist-count/{id}>

Parametry:

* {id} - id serwisu dla którego chcemy pobrać licznik elementów blacklist

W efekcie otrzymamy informację o ilości elementów na liście która pozwoli nam na stworzenie p

21

Pobieranie fragmentu listy dostępnych numerów msisdn na blacklist - GET

http://beta2:9080/mqube/rest/blacklist/{serviceId}/{from}/{to}

Parametry:

- * {id} - id serwisu dla którego będziemy pobierać fragment listy numerów blacklist
- {from} - pozycja od której mają być wyświetlane numery z listy
- {to} - pozycja do której mają być wyświetlane numery z listy

W efekcie otrzymujemy listę obiektów BlacklistedDTO:

48506123568
SMS
2011-09-30 16:57:59.998
1921952

Usuwanie numeru msisdn z blacklist - DELETE

http://beta2:9080/mqube/rest/blacklist/{msisdndId}

Parametry:

- * {msisdndId} - id obiektu blacklistedDTO w którym znajdują się informacje na temat usuwanego numeru msisdn

Wyszukiwanie na blacklist po fragmencie numeru msisdn - GET

http://beta2:9080/mqube/rest/blacklist/{serviceId}/{string}

Parametry:

- * {serviceId} - id serwisu w którego blackliście będziemy szukać numeru
- {text} - fragment numeru msisdn który chcemy znaleźć

---++++ Players
----- Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * **import** informacji na temat danego gracza
- * **export** informacji na temat graczy z wybranego przedziału czasowego
- * **export** unikalnej listy numerów msisdn z wybranego przedziału czasowego (lista zarejestro

---++++ Wykorzystywane metody REST

Import informacji na temat danego gracza - PUT

http://beta2:9080/mqube/rest/blacklist/{serviceId}/{from}/{to}

Parametry:

* {id} - id serwisu dla którego importować dane o graczach

w body wysyłamy listę obiektów które chcemy zaimportować:

```
00000000
MONTEZUMA
1
19
SAN JOSE
MEN
```

Oczywiście w ramach pojedynczego requesta możemy przesłać większą ilość obiektów SimplePlayer

Export informacji na temat graczy - GET

http://beta2:9080/mqube/rest/simple-player/{id}/{dateFrom}/{dateTo}

Parametry:

- {id} - id serwisu dla którego importować dane o graczach
- {datefrom}
 - data od - ustawianie przedziału czasowego
- {dateto} - data do - ustawianie przedziału czasowego

Export listy unikalnych numerów msisdn - GET

http://beta2:9080/mqube/rest/registered-players/{id}/{dateFrom}/{dateTo}

Parametry:

- {id} - id serwisu dla którego importować dane o graczach
- {datefrom}
 - data od - ustawianie przedziału czasowego
- {dateto} - data do - ustawianie przedziału czasowego

----- Stages/Results

----- Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy etapów dostępnych w wybranym przez nas serwisie
- * tworzenie, edycja, usuwanie etapów w serwisie
- * generowanie wyników - (dla etapów w których jest ustawione ręczne generowanie wyników)
- * podgląd wyników dla etapów i serwisu

----- Wykorzystywane metody REST

Pobieranie listy dostępnych stage w wybranym serwisie - GET

W tym celu pobieramy konfigurację serwisu, z niej wyciągamy zawartość tagu stageList, który z

```
2011-05-25 02:00:00.0
2011-08-25 02:00:00.0
2012-12-31 11:30:00.0
2011-08-26 02:00:00.0
2012-12-31 11:30:00.0
false
true
48509756730
48695713086
KD
48506047148
2011-05-25 02:00:00.0
2012-06-29 11:30:00.0
```

DRAWING
STAGE
false
true
JACKPOT
1
2
false
true
DAILY
2550316007900
true
true
false
false
[mailto:
anna.sauvignon@avantis.pl]
false
true
1
2
true
2011-06-10 19:11:00.0
false
false
false
false
199851

Tworzenie nowego stage - POST

<http://beta2:9080/mqube/rest/stage>

Jako body wysyłamy xml z przykładową konfiguracją stage:

2012-10-18 12:49:00 CEST
2012-10-19 12:49:00 CEST
false
false
Banan
false
STAGE
NONE
0
false
199851

Edycja stage - PUT

<http://beta2:9080/mqube/rest/stage>

Jako body podajemy zmodyfikowaną konfigurację stage.

2012-10-18 12:49:00 CEST
2012-10-19 12:49:00 CEST
false
false
Banan
false
STAGE
NONE
0
false
199851

Usuwanie stage - DELETE

<http://beta2:9080/mqube/rest/stage/{serviceId}/{id}>

Parametry:

<https://wikis.services.avantis.pl/display/BPP/%5BDOC%5D+mQube+-+dokumentacja+techniczna>

- {serviceId} - id serwisu w którym znajduje się usuwany stage
- {id} - Id usuwanego stage

Generowanie wyników - (dla etapów w których jest ustawione ręczne generowanie wyników) - POST

<http://beta2:9080/mqube/rest/generate-stage-result/{id}>

Parametry:

- {id} - Id stage dla którego chcemy wygenerować wyniki

Podgląd wyników dla etapów - GET

<http://beta2:9080/mqube/rest/stage-result/{id}>

Parametry:

- {id} - Id stage dla którego chcemy pobrać wyniki

Podgląd wyników dla serwisu - GET

<http://beta2:9080/mqube/rest/service-result/{id}>

Parametry:

- {id} - Id serwisu dla którego chcemy pobrać wyniki

----- Bulks

----- Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy bulków dostępnych w wybranym przez nas serwisie
- * tworzenie, edycja, stopowanie bulków w serwisie

----- Wykorzystywane metody REST

lista bulków w wybranym serwisie

<http://beta2:9080/mqube/rest/bulk/{id}/{datefrom}/{dateto}>

Parametry:

- {id} - id serwisu dla którego chcemy pobrać listę bulków np: 199851
- {datefrom} - data od np : 2011-08-30T00:00:00+02:00
- {dateto} - data do np : 2012-09-13T00:00:00+02:00

Tworzenie bulka - POST

<http://beta2:9080/mqube/rest/bulk>

Jako body podajemy obiekt nowego bulka:

false

2012-10-18 13:48:00.0 CEST

false

xxxd

2012-10-19 13:48:00.0 CEST

false

SMS

1

2012-10-19 13:48:00.0 CEST

false

SALARY

199851
STANDARD
false
false
default

Pobieranie bulka - GET

<http://beta2:9080/mqube/rest/bulk/{id}>

Parametry:

- {id} - id bulka którego konfiguracje chcemy pobrać np 2550316038650

Modyfikacja bulka - PUT

<http://beta2:9080/mqube/rest/bulk>

Jako body podajemy obiekt zmodyfikowanego bulka:

false
2012-10-18 13:48:00.0 CEST
false
xxxxd
2012-10-19 13:48:00.0 CEST
false
SMS
1
2012-10-19 13:48:00.0 CEST
false
SALARY
199851
STANDARD
false
false
default

Stopowanie bulka - PUT

<http://beta2:9080/mqube/rest/bulk-recipients/{id}>

Parametry:

* {id} - id bulka którego konfiguracje chcemy pobrać np 2550316038650

---++++ Questions
---++++ Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy pytań dostępnych w wybranym przez nas serwisie
- * tworzenie, edycja, usuwanie pytań w serwisie

---++++ Wykorzystywane metody REST

Pobieranie listy dostępnych pytań w wybranym serwisie - GET

W tym celu pobieramy konfigurację serwisu, z niej wyciągamy zawartość tagu questions, który znajduje

pytanie (odp a)
A
1
true

2550316026150
30
27
2550316026100
true
CONSECUTIVELY
false
false
STANDARD
199851

Tworzenie nowych pytań - POST

`http://beta2:9080/mqube/rest/question`

Jako body podajemy obiekt nowego pytania:

ab
\$2
199851
2
true

Edycja pytań - PUT

`http://beta2:9080/mqube/rest/question`

Jako body podajemy obiekt pytania:

ab
\$2
199851
2
true

Usuwanie pytań - PUT

`http://beta2:9080/mqube/rest/service`

W celu usunięcia pytania z serwisu usuwamy wybrane pytanie z konfiguracji serwisu następnie zmodyfikujemy

---++++ Answer codes

---+++++ Opis ogólny

Z poziomu tej zakładki mamy dostęp do:

- * listy kodów odpowiedzi dla pytań dostępnych w wybranym przez nas serwisie
- * tworzenie, edycja, usuwanie kodów odpowiedzi w serwisie

---+++++ Wykorzystywane metody REST

Pobieranie listy kodów odpowiedzi w wybranym serwisie - GET

W tym celu pobieramy konfigurację serwisu, z niej wyciągamy zawartość tagu answerCodes, który znajduje

A
2550316016550
B
2550316016551
C
2550316016552
D

12/17/2018

[DOC] mQube - dokumentacja techniczna - [1] PORTFOLIO PRODUCTS - Digital Virgo Confluence

2550316016553

E

2550316016554

F


2550316016555

G

2550316016556

YYYcodeYYY

Tworzenie, edycja, oraz usuwanie kodów odpowiedzi w serwisie odbywa się poprzez modyfikację listy answerCodes i jej elementów

 Like Be the first to like this