

INTEGRACJA SYSTEMU NASZ WROCŁAW Z SYSTEMEM MENNICY

Koncepcja techniczna

SPIS TREŚCI:

1.1	Założenia techniczne	2
1.2	Model techniczny rozwiązania	2
1.3	Proponowane metody API REST (Nazwa, URI, Metoda, Parametry URI, Zwracane dane)	4
1.3.1	Metod getChanges	4
1.3.2	Metod confirmChanges	7
1.3.3	Metoda getFullState	8
1.4	Przykładowe diagramy przepływów i interakcji	10
1.4.1	Wywołanie inicjalizujące	10
1.4.2	Wywołanie PULL	11
1.5	Zakres integracji pomiędzy UC a NW	12

1.1 ZAŁOŻENIA TECHNICZNE

Integracja Systemu Urbancard (dalej „UC”) z Systemem Nasz Wrocław (dalej „NW”) musi opierać się na rozwiązaniu gwarantującym:

- a. bezpieczeństwo infrastruktury UC,
- b. integralność przekazywanych danych,
- c. braku wysyłania danych użytkowników UC do zewnętrznych podmiotów,
- d. zapewnienie minimalnego lub zerowego opóźnienia procesu sprzedaży produktów,
- e. brak wpływu dostępności zewnętrznych serwisów na dostępność usług UC,
- f. wielopoziomowe i konfigurowalne mechanizmy raportowania stanu synchronizacji danych i automatyczne powiadamianie o wszelkich nieprawidłowościach w kanale komunikacyjnym w ramach infrastruktury UC,

Model techniczny rozwiązania

Na podstawie przyjętych założeń technicznych proponuje się następujący model rozwiązania:

1. W systemie UC przechowywana jest aktualna informacja o statusie wybranego użytkownika z NW. Użytkownik jest identyfikowany przez zahaszowany numer PESEL, za pomocą metody SHA1 z użyciem stałej dodatkowej wartości haszującej oraz dodatkowo przez zahaszowane numer telefonu (opcjonalnie) za pomocą metody SHA1 z użyciem stałej dodatkowej wartości haszującej.
Np. $\text{SHA1}(\text{numer_pesel} + \text{stala_wartosc})$
2. Pole z numerem telefonu jest wypełnione tylko dla użytkowników, którzy podali w NW dane numeru telefonu.
3. System NW wystawia dedykowane API komunikacyjne, poprzez które pobierane są wcześniej nie odebrane (potwierdzone) przez UC zmiany związane z kontami użytkowników (dodanie, modyfikacja, usunięcie). UC komunikuje się z API w określonych interwałach czasowych (np. co 1 minutę). API wymaga zalogowania oraz wykorzystywania zwróconego TOKEN’u w następnych wywołaniach API. Logowanie opiera się na gotowym module autoryzacji dla Partnerów. Dostęp do poniższych danych otrzymują Partnerzy posiadający rolę / uprawnienie „Dostęp do API – status podatnika”.

4. System NW wystawi API do potwierdzania odebrania i poprawnego zapisu informacji o kontaktach z pkt. 3.
5. Dodatkowo System NW wystawi API zwracające aktualny stan WSZYSTKICH kont zarejestrowanych w NW - co zapewni weryfikację integralności posiadanych danych przez partnera. Stan ten może być w formie migawki – aktualizowanej co pewien czas po stronie NW (np. co 1 godzinę) aby nie obciążać znacząco systemu NW oraz aby umożliwić użycie mechanizmów cache. Interwał komunikacji ze strony partnera byłby odpowiednio dłuższy – np. co kilka godzin.
6. Możliwe jest użycie bezpiecznego kanału komunikacji HTTPS.
7. Pierwsze połączenie z systemu partnera powinno zacząć się od pobrania danych z metody zwracającej aktualny stan systemu, zapisać stany w lokalnej bazie danych i potwierdzić rekordy zmian. Dopiero po POPRAWNYM wykonaniu tego kroku powinno się używać odpytywania metody API dla listy zmian w ramach identyfikatora partnera.
8. Kody odpowiedzi dla błędów w API systemu NW powinny opierać się na odpowiednich kodach http innych niż 2xx oraz ustalonej strukturze obiektu JSON zaproponowanej przez NW (cała odpowiedź kodowana w UTF-8).

1.2 PROPONOWANE METODY API REST (NAZWA, URI, METODA, PARAMETRY URI, ZWRACANE DANE)

1.2.1 Metod getChanges

Pole	Opis
Nazwa	getChanges
URI	https://server/getChanges?partner=PID&limit=LIM
Metoda HTTP	GET
Parametry URI	<ul style="list-style-type: none"> PID - Partner ID – identyfikator partnera łączącego się do NW, dopuszczalne zestaw znaków w identyfikatorze to [a-zA-Z0-9], parametr WYMAGANY. LIM – dowolna wartość z zakresu UINT32, parametr OPCJONALNY, domyślnie 100. Parametr określa maksymalną ilość zwracanych rekordów zmian.
Kod odpowiedzi poprawnej	200 - OK
Typ zawartości	application/json; charset=utf-8
Przykładowa odpowiedź	<pre>{ "service": "Nasz Wrocław", "method": "getChanges", "partner_name": "Wrocław Urbancard", "record_limit": 100, "record_count": 3, "has_more_data": false, "data_checksum_md5": "7e0d5df2bdc2823a7ec79ea616631ad1", "data": "WwogICB7CiAgICAgICJpZCI6MTM0Njc2LAogICAgICAIidHlwZSI6Ikk4iLAogICAgICAIUEVTRUwiOiI0NDExMTIyMjMzMyIsCiAgICAgICJtb2JpbGUiOiIiLAogICAgICAIY2hhbmdlX2RhdGVfdGltZSI6IjIwMTgtMDctMzEgMTI6MTU6MDAiCiAgICAgIH0sCiAgICAgIHsKICAgICAgImlkIjoxMzQ5NzIsCiAgICAgICJ0eXB1IjojRCIsCiAgICAgICJQRVNFTCI6IjU1MTEyMjMzIiwKICAgICAgIm1vYm1sZSI6IjIyMzQ1Njc4OSIsCiAgICAgICJjaGFuZ2VfZGF0ZV90aW11IjojMjAxOC0wNy0zMSAxMjoxNjo1MSIKICAgfSwKICAgewogICAgICAIaWQiOjEzNDY3NiwKICAgICAgInR5cGUiOiJNIIiwKICAgICAgIlBFU0VMIjojMDAxMTEyMjMzMiLAogICAgICAgIibW9iaWx1IjojNTY3NTY5MjU5IiwKICAgICAgImNoYW5nZV9kYXR1X3RpbWUiOiIyMDE4LTA4LTI3IDE5OjE1OjAwIgowICB9Cl0="</pre>
Opis	<p>Metoda zwraca listę NIEPOTWIERDZONYCH przez partnera (UC) zmian w ramach kont użytkowników. Uwaga, zwracane rekordy posortowane są ROSNĄCO według daty zmiany. W Przypadku braku danych zwrócona zostanie pusta tablica.</p> <p>service – text - nazwa serwisu obsługującego żądanie</p>

method – text - nazwa metody API
partner_name – text -nazwa partnera skojarzona z parametrem PID
record_limit – int32 - limit rekordów z parametru LIM lub jeżeli nie został podany to wartość domyślna
record_count – int32 - ilość rekordów zmian w polu data
has_more_data – bool – wartość true lub false oznaczająca, iż jest więcej danych do pobrania niż obecnie zwrócona paczka rekordów
data_checksum_md5 – text – suma kontrolna md5 zawartości pola data
data – text – tablica rekordów zmian zakodowana w formacie base64

Przykładowa struktura pola **data** dla braku danych:

```
[  
]
```

Przykładowa struktura pola data dla danych:

```
[  
  {  
    "change_id":394679,  
    "account_id":134676,  
    "type":"N",  
    "PESEL":"b444ac06613fc8d63795be9ad0beaf55011936ac",  
    "mobile":"","  
    "change_date_time":"2018-07-31T12:15:00"  
  },  
  {  
    "change_id":394680,  
    "account_id":664476,  
    "type":"D",  
    "PESEL":"b444ac06613fc8d63795be9ad0beaf55011936ac",  
    "mobile":"","  
    "change_date_time":"2018-07-31T12:16:51"  
  },  
  {  
    "change_id":394681,  
    "account_id":1834676,  
    "type":"M",  
    "PESEL":"b444ac06613fc8d63795be9ad0beaf55011936ac",  
    "mobile":"be8d339f56caad4f493e2a965f2a15edabf34b64",  
    "change_date_time":"2018-08-27T19:15:00"  
  }  
]
```

Gdzie odpowiednio:

- change_id – int32 - unikalny identyfikator zmiany w NW
- account_id – int32 - unikalny identyfikator konta w NW
- type – char(1), gdzie wartości to:
 - N – utworzono nowe konto
 - D – usunięto konto (lub cofnięto uprawnienia do ulgi)

	<ul style="list-style-type: none">○ M – zmieniono dane (numer telefonu lub PESEL)¹● PESEL – char(11) – numer PESEL²● mobile – char(12) – numer telefonu (opcjonalny)³● change_date_time – data zmiany⁴
--	---

¹ Uwaga – zakładamy, że numer PESEL jest unikalny w NW i nie można zmienić tego numeru (zmianie podlega wtedy tylko numer telefonu) lub NW gwarantuje, że po zmianie unikalność nadal jest zachowana

² Zgodnie z <https://www.gov.pl/web/gov/czym-jest-numer-pesel> (SHA1(numer_pesel+stala_wartosc))

³ Zgodnie z formatem +48123456789 (SHA1(numer_pesel+stala_wartosc))

⁴ Zgodnie z formatem YYYY-MM-DDThh:mm:ss

record_count – int32 - ilość rekordów zmian w polu data
compression – bool – opcjonalna kompresja zbioru (przed kodowaniem base64), algorytm kompresji to gzip⁵
data_checksum_md5 – text – suma kontrolna md5 zawartości pola data
data – text – tablica rekordów zmian zakodowana w formacie base64

Przykładowa struktura pola data:

```
[
{
  "change_id":394679,
  "account_id":134676,
  "PESEL":" b444ac06613fc8d63795be9ad0beaf55011936ac ",
  "mobile":"",
  "change_date_time":"2018-07-31T12:15:00"
},
{
  "change_id":394681,
  "account_id":1834676,
  "PESEL":" b444ac06613fc8d63795be9ad0beaf55011936ac ",
  "mobile":"be8d339f56caad4f493e2a965f2a15edabf34b64",
  "change_date_time":"2018-08-27T19:15:00"
}]
```

Gdzie odpowiednio:

- change_id – int32 - unikalny identyfikator zmiany w NW
- account_id – int32 - unikalny identyfikator konta w NW
- PESEL – char(11) – numer PESEL⁶
- mobile – char(12) – numer telefonu (opcjonalny)⁷
- last_change_date_time – data ostatniej zmiany w koncie użytkownika⁸

⁵ Zgodnie z <http://www.faqs.org/rfcs/rfc1952.html>

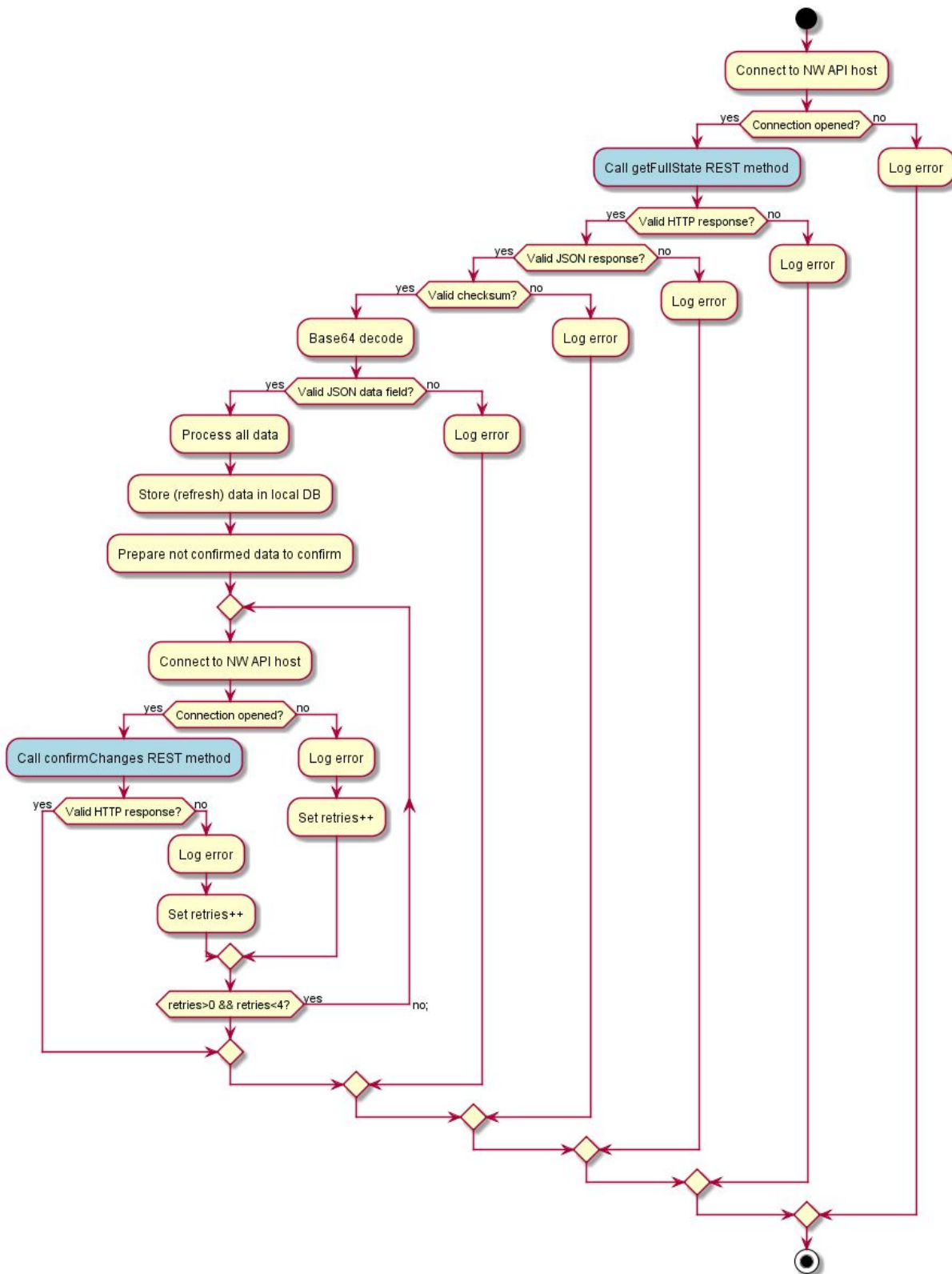
⁶ Zgodnie z <https://www.gov.pl/web/gov/czym-jest-numer-pesel> (SHA1(numer_pesel+stala_wartosc))

⁷ Zgodnie z formatem +48123456789 (SHA1(numer_pesel+stala_wartosc))

⁸ Zgodnie z formatem YYYY-MM-DDThh:mm:ss

1.3 PRZYKŁADOWE DIAGRAMY PRZEPŁYWÓW I INTERAKCJI

1.3.1 Wywołanie inicjalizujące



1.3.2 Wywołanie PULL

