INSTRUKCJA POŁĄCZENIA ZE ŚRODOWISKIEM PRODUKCYJNYM API XS2A W CREDIT AGRICOLE

Wersja 1.4

CRÉDIT AGRICOLE

Spis treści

Wymagania wstępne

Przed rozpoczęciem pracy ze środowiskiem produkcyjnym należy:

- Uzyskać certyfikaty kwalifikowane niezbędne do komunikacji z endpointami API XS2A.
- Zapoznać się z dokumentacją techniczną usług zamieszczoną w Serwisie informacyjnym API Portal https://www.credit-agricole.pl/apiportal.

Podczas pracy ze środowiskiem produkcyjnym należy pamiętać, że:

- Połączenie z endpointami realizowane jest z użyciem TLS z dwustronnym uwierzytelnianiem certyfikatami QWAC.
- Poszczególne żądania i odpowiedzi muszą być podpisane za pomocą certyfikatów QSEAL. Zgodnie ze standardem PolishAPI (https://polishapi.org/#docs) wywołując usługę należy w żądaniu umieścić nagłówek X-JWS-SIGNATURE zawierający podpis JWS treści żądania. Szczegóły generowania podpisu JWS opisano w dalszej części dokumentu.
- Zawartość pola `"requestId"` w każdym żądaniu musi być unikalna.
- Zawartość pola `"tppId"` w żądaniach musi być zgodna z wartością pola Podmiot - 2.5.4.97 (organizationIdentifier) w wykorzystywanych certyfikatach.
- Zawartość pola `"client_id"` w żądaniach musi być zgodna z wartością pola `"tppId"`
- Rozwiązanie nie obsługuje dedykowanej usługi onboardingu. Pierwsze, poprawne wywołanie usługi /authorize z wykorzystaniem ważnych certyfikatów jest równoznaczne z rejestracją aplikacji TPP.

Generowanie podpisu JWS

W celu zapewnienia integralności i niezmienności przesyłanych komunikatów każdy z nich musi posiadać nagłówek `X-JWS-SIGNATURE` którego wartość zawiera podpis JWS żądania. Podpis JWS powinien zostać wygenerowany zgodnie ze standardem RFC 7515. Ponadto podpis JWS powinien zostać przygotowany bez załączonego payloadu (detached) oraz wyliczony na podstawie niezakodowanego payloadu (Unencoded Payload Option - RFC 7797).

Nagłówek podpisu JWS powinien zawierać następujące parametry

- `"alg"` – algorytm użyty podczas podpisywania – pole powinno zawierać wartość `"RS256"`
- `"x5c"` – certyfikat lub ścieżka certyfikacji odpowiadająca kluczowi użytemu do wygenerowania podpisu
- `"x5u"` – adres URL do certyfikatu odpowiadającego kluczowi użytemu do wygenerowania podpisu
- `"x5t#S256"` – zakodowany w base64url odcisk palca certyfikatu odpowiadającego kluczowi użytemu do wygenerowania podpisu
- `"b64"` – wskazanie, czy podpis wygenerowano na pozdstawie zakodowanego payloadu – pole powinno zawierać wartość `false`
- `"kid"` – identyfikator klucza użytego do wygenerowania podpisu

Parametry wskazujące na użyty certyfikat `"x5c"` / `"x5u"` należy stosować zamiennie.

Przykład wywołania usługi AIS

Poniżej prezentujemy przebieg przykładowego wywołania usługi AIS getAccount na środowisku produkcyjnym dostępnym pod adresem https://xs2a.credit-agricole.pl/CaPolishAPI/prod/individual. Realizacja odbywa się w trzech krokach, które zostały szczegółowo opisane poniżej.

Autoryzacja

Pierwszym krokiem jest pobranie kodu autoryzacji. W tym celu należy wywołać usługę `/authorize` z poniższym payloadem:

```
{
    "requestHeader": {
        "requestId": "8a740673-c751-4558-86e7-9fab31d91c4e",
        "tppId": " YYYYY-ZZZZ-TPPIdentificator",
        "userAgent": "SOAP-UI accounts.0-ais-getAccount",
        "isCompanyContext": false,
        "ipAddress": "127.0.0.1",
        "sendDate": "2019-09-06T09:36:47.536Z"
    },
    "response_type": "code",
    "client_id": " YYYYY-ZZZZ-TPPIdentificator ",
    "redirect_uri": "http://example.com/",
    "state": "252a5f94-dc2a-4260-9070-af91e3eb3cde",
    "scope": "ais",
    "scope_details": {
        "scopeGroupType": "ais",
        "consentId": "ffd4954c-e2c5-488d-b7e9-eeffad0c64ac",
        "scopeTimeLimit": "2019-10-06T11:28:50.000+02:00",
```

```
        "throttlingPolicy": "psd2Regulatory",
        "privilegeList": [
            {
                "accountNumber": "PL78194000086704648427357299",
                "ais:getAccount": {
                    "scopeUsageLimit": "single"
                }
            }
        ]
    }
}
```

Powyższy payload należy podpisać za pomocą certyfikatu QSEAL i umieścić w nagłówku X-JWS-SIGNATURE żądania do usługi REST. Do poprawnego przetworzenia żądania niezbędne jest umieszczenie w nim następujących nagłówków:

```
Accept-Language: pl
X-REQUEST-ID: {wartość zgodna z polem "requestId" w payloadzie}
Accept: application/json
Accept-Charset: utf-8
Accept-Encoding: deflate
X-JWS-SIGNATURE: {podpis JWS żądania}
```

W opisywanym przykładzie wywołanie usługi REST wygląda następująco:

```
POST https://xs2a.credit-
agricole.pl/CaPolishAPI/prod/individual/v3_0.1/auth/v3_0.1/authorize
HTTP/1.1
Accept-Encoding: gzip,deflate
Accept-Language: pl
X-REQUEST-ID: 8a740673-c751-4558-86e7-9fab31d91c4e
Accept: application/json
Accept-Charset: utf-8
Accept-Encoding: deflate
X-JWS-SIGNATURE: eyJ4NWMiOlsiTUlJRnl6Q0NCTE9nQXdJQkFnSUpBSUU1eGo5RC9CdzdN(...)
Content-Type: application/json;charset=UTF-8
Content-Length: 874
Host: xs2a.credit-agricole.pl
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Oczekiwaną odpowiedzią jest adres URL do serwisu bankowego, który pozwala na autoryzowanie przez klienta zgody na realizację usługi.

Przykładowa odpowiedź zawierająca adres przekierowania:

```
{
    "responseHeader": {
        "requestId": "8a740673-c751-4558-86e7-9fab31d91c4e",
        "sendDate": "2019-09-06T09:36:48.834+02:00",
        "isCallback": false
    },
    "aspspRedirectUri": "https://ca24.credit-
agricole.pl/login?key=bQpG2SInmCEjQjYs&hash=%2B8s4CtTcrEiicV5N5aOxCHkpQ3k%3D"
}
```

Po przejściu pod adres przekazany w polu `"aspspRedirectUri"` docieramy do serwisu bankowego. Po autoryzowaniu zgody w serwisie bankowym, nastąpi przekierowanie pod adres wskazany w żądaniu w polu `"redirect_uri"`. W powyższym przykładzie będzie to `"http://example.com/"`. Adres zostanie rozszerzony o parametry: `"code"` zawierający kod autoryzacji wymagany w kolejnym kroku, oraz `"state"` o wartości przekazanej w żądaniu authorize. W opisywanym przykładzie kompletny adres przekierowania wygląda następująco:

`http://example.com?code=zJAjnsTwTPkTAekm&state=252a5f94-dc2a-4260-9070-af91e3eb3cde`

Uzyskanie tokena

Kolejnym krokiem jest wygenerowanie tokena dostępowego. W tym celu należy wywołać usługę `/token` z payloadem zawierającym w polu `"code"` wartość otrzymanego kodu autoryzacji. W opisywanym przykładzie payload wygląda następująco:

```
{
    "requestHeader": {
        "requestId": "7aa46703-1cac-457e-b616-a50f9f514713",
        "tppId": "YYYYY-ZZZZ-TPPIdentificator",
        "userAgent": "SOAP-UI accounts.0-ais-getAccount",
        "ipAddress": "127.0.0.1",
        "isCompanyContext": true,
        "sendDate": "2019-09-06T09:37:42.632Z"
    },
    "grant_type": "authorization_code",
    "code": "zJAjnsTwTPkTAekm",
    "client_id": "YYYYY-ZZZZ-TPPIdentificator",
    "redirect_uri": "http://example.com/"
}
```

Do poprawnego przetworzenia żądania niezbędne jest umieszczenie w nim następujących nagłówków:

```
Accept-Language: pl
X-REQUEST-ID: {wartość zgodna z polem "requestId" w payloadzie}
Accept-Charset: utf-8
Accept-Encoding: deflate
X-JWS-SIGNATURE: {podpis JWS żądania}
```

Oczekiwaną odpowiedzią jest token dostępu do realizacji usługi. W opisywanym przykładzie odpowiedź wygląda następująco:


```
{
    "responseHeader": {
        "requestId": "7aa46703-1cac-457e-b616-a50f9f514713",
        "sendDate": "2019-09-06T09:37:46.313+02:00",
        "isCallback": false
    },
    "access_token": "O3/D6+rfvDOTnUjMQx2EU6AQxGk=",
    "token_type": "bearer",
    "expires_in": "120",
    "refresh_token": "4oL6qdF86tHj48k2",
    "scope": "psd2-ais",
    "scope_details": {
        "privilegeList": [
            {
                "accountNumber": "PL78194000086704648427357299",
                "ais:getAccount": {
                    "scopeUsageLimit": "single"
                }
            }
        ],
        "consentId": "ffd4954c-e2c5-488d-b7e9-eeffad0c64ac",
        "scopeTimeLimit": "2019-10-06T11:28:50.000+02:00",
        "throttlingPolicy": "psd2Regulatory"
    }
}
```


Odpowiedź zawiera token w polu `"access_token"`. Jeżeli token straci ważność należy wygenerować następny używając tokena przekazanego w polu `"refresh_token"` o ile został wydany.

Realizacja usługi

Ostatnim krokiem jest realizacja usługi biznesowej. W omawianym przykładzie jest to pobranie informacji o koncie. W tym celu należy wywołać usługę `/getAccount` z payloadem zawierającym w polu `"token"` wartość otrzymaną w poprzednim kroku w polu `"access_token"`. Dla omawianego przykładu payload wygląda następująco:


```
{
    "requestHeader": {
        "requestId": "fa884132-d089-11e9-bb65-2a2ae2dbcce4",
        "userAgent": "SOAP-UI accounts.0-ais-getAccount",
        "ipAddress": "127.0.0.1",
        "sendDate": "2019-09-06T09:37:50.583+02:00",
        "tppId": "YYYYY-ZZZZ-TPPIdentificator",
        "token": "O3/D6+rfvDOTnUjMQx2EU6AQxGk=",
        "isDirectPsu": true
    },
```

```
    "accountNumber": "PL78194000086704648427357299"
}
```

Do poprawnego przetworzenia żądania niezbędne jest umieszczenie w nim następujących nagłówków:

```
Accept-Language: pl
Authorization: { wartość zgodna z polem "token" w payloadzie }
X-REQUEST-ID: {wartość zgodna z polem "requestId" w payloadzie}
Accept-Charset: utf-8
Accept-Encoding: deflate
X-JWS-SIGNATURE: {podpis JWS żądania}
```

Oczekiwaną odpowiedzią jest szczegółowa informacja o koncie. W opisywanym przykładzie odpowiedź wygląda następująco:

```
{
    "responseHeader": {
        "requestId": "fa884132-d089-11e9-bb65-2a2ae2dbcce4",
        "sendDate": "2019-09-06T09:37:56.148+02:00",
        "isCallback": false
    },
    "account": {
        "accountNumber": "PL78194000086704648427357299",
        "nameAddress": {
            "value": [
                "ul. Niska 1, 50-000 Wrocław"
            ]
        },
        "accountType": {
            "code": "5555",
            "description": "description"
        },
        "accountTypeName": "Account Type Name",
        "accountHolderType": "individual",
        "accountNameClient": "Client Name",
        "currency": "PLN",
        "availableBalance": "99999",
        "bookingBalance": "99999",
        "bank": {
            "bicOrSwift": "AGRIPLPR",
            "name": "Credit Agricole Bank Polska SA",
            "address": [
                "Credit Agricole Bank Polska SA",
                "Legnicka 48 bud.C-D",
                "54-202 Wrocław"
            ]
        },
        "auxData": {
```

```
            "additionalProp1": "",
            "additionalProp2": "",
            "additionalProp3": ""
        }
    }
}
```

Uzyskanie powyższej odpowiedzi oznacza poprawne wykonanie wywoływanej w tym przykładzie usługi AIS getAccount. Wywołanie pozostałych usług AIS, PIS oraz CAF przebiega analogicznie do przedstawionego powyżej przykładu. Specyfikacja wywołań poszczególnych usług dostępna jest w dokumentacji technicznej zamieszczonej w Serwisie API Portal https://apiportal.credit-agricole.pl.

Zmiany i uszczegółowienia względem standardu PolishAPI

W niektórych aspektach standard PolishAPI jest mało precyzyjny i bywa różnie interpretowany. Poniżej przedstawiamy doprecyzowania wybranych obszarów w kontekście tego jak zostały one zrealizowane po stronie banku. Zakres zgody

W wywołaniu usługi `/authorize` nie są obsługiwane zgody na pobranie statusu płatności, czyli:

- pis:getPayment
- pis:getBundle
- pis:getRecurringPayment

Ponadto przyjęto następujące reguły w zakresie możliwych kombinacji uprawnień:

- uprawnienia z grupy pis mogą być przekazywane w usłudze tylko pojedynczo (nie mogą być łączone z innymi uprawnieniami)
- uprawnienia z grupy ais mogą być łączone ze sobą w dowolnych kombinacjach
- uprawnienia ais-accounts muszą być przekazywane pojedynczo (nie mogą być łączone z innym uprawnieniami)

Pozyskanie uprawnień na pobranie statusu płatności

Zgodnie ze standardem Polish API pozyskanie uprawnień na pobranie statusu płatności, statusu paczki przelewów lub płatności cyklicznej możliwe jest przez wymianę tokena (refreshToken) wydanego dla zgody na płatność, paczkę przelewów lub płatności cyklicznej. Wymiana takiego tokena następuje przy użyciu usługi token w trybie refreshToken po zleceniu przelewu, paczki przelewów, płatności cyklicznej (po zużyciu uprawnienia z oryginalnej zgody), ale przed upłynięciem okresu ważności oryginalnej zgody. . Możliwe są następujące przejścia:

Uprawnienie pierwotne	Uprawnienie pozyskiwane
pis:domestic	pis:getPayment
pis:EEA	pis:getPayment
pis:nonEEA	pis:getPayment
pis:tax	pis:getPayment
pis:bundle	pis:getBundle
pis:recurring	pis:getRecurringPayment

Poniżej prezentujemy przykładowe wywołanie usługi `/token` pozyskujące uprawnienia do wykonania usługi `/getPayment` na podstawie wcześniej uzyskanej zgody na wykonanie usługi `/domestic`

```
{
    "requestHeader": {
        "requestId": "2bc25652-e8d6-11e9-81b4-2a2ae2dbcce4",
        "tppId": "YYYYY-ZZZZ-TPPIdentificator",
        "userAgent": "SOAP-UI refreshToken",
        "isCompanyContext": false,
        "ipAddress": "127.0.0.1",
        "sendDate": "2019-10-04T19:38:22.139Z"
    },
    "grant_type": "refresh_token",
    "refresh_token": "wkqLFectV5WsXPmd",
    "scope_details": {
        "scopeGroupType": "pis",
        "throttlingPolicy": "psd2Regulatory",
        "consentId": "72839b4e-e8ec-11e9-81b4-2a2ae2dbcce4",
        "scopeTimeLimit": "2019-10-16T11:28:50.000+02:00",
        "privilegeList": [
            {
                "pis:getPayment": {
                    "scopeUsageLimit": "multiple",
                    "paymentId": "3243564",
                    "tppTransactionId": "85638563"
                }
            }
        ]
    }
}
```

W powyższym wywołaniu należy pamiętać, że:

- wartość pola `"refresh_token"` musi być równa wartości zwróconej w polu `"refresh_token"` po wywołaniu usługi `/token` pozyskującym token do realizacji usługi pierwotnej – w tym przypadku `/domestic`
- wartość pola `"consentId"` musi być równa wartości przekazanej podczas wywołania usługi `/authorize` dla zgody pierwotnej – w tym przypadku `/domestic`
- wartość pola `"paymentId"`, `"bundleId"` lub `"recurringPaymentId"` musi być równa wartości zwróconej po wywołaniu usługi pierwotnej – w tym przypadku `/domestic`
- wartość pola `"tppTransactionId"`, `"tppBundleId"` lub `"tppRecurringPaymentId"`” musi być równa wartości przekazanej podczas wywołania usługi z pierwotnej zgody – w tym przypadku `/domestic`. Pole jest opcjonalne.
- w polu `"scopeTimeLimit"`” można wskazać nową datę ważności zgody i data ta nie jest walidowana względem daty ważności z pierwotnej zgody (można określić szerszy zakres czasowy dla nowego uprawnienia)

- w polu `"scopeUsageLimit"`” można wskazać inną wartość niż w pierwotnej zgodzie (można pozyskać wielokrotne uprawienie na pobranie statusu przelewu, paczki przelewów lub płatności cyklicznej)

Uszczegółowienie zakresu zgody

Usługa `/token` w trybie `exchangeToken` pozwala na wymianę tokena ze zgodą ais-accounts:getAccounts na token z nowym, uszczegółowionym zakresem zgody AIS dla wybranych rachunków. W trybie tym możliwe jest pozyskanie następujących uprawnień:

- ais:getAccount
- ais:getHolds
- ais:getTransactionsDone
- ais:getTransactionsPending
- ais:getTransactionsRejected
- ais:getTransactionsCancelled
- ais:getTransactionsScheduled
- ais:getTransactionDetail

Przykładowe wywołanie pozyskujące uprawnienie getAccount na podstawie zgody getAccounts wygląda następująco:

```
{
    "requestHeader": {
        "requestId": "8b62b3d0-54cd-404d-8daa-918013599cc9",
        "tppId": "YYYYY-ZZZZ-TPPIdentificator",
        "userAgent": "SOAP-UI exchangeToken",
        "ipAddress": "127.0.0.1",
        "isCompanyContext": false,
        "sendDate": "2019-10-16T19:38:22.139Z"
    },
    "grant_type": "exchange_token",
    "exchange_token": "ImOUbP6_AOSuVjFMz3GD177SQjs=",
    "scope": "ais",
    "scope_details": {
        "scopeGroupType": "ais",
        "throttlingPolicy": "psd2Regulatory",
        "consentId": "e798d37c-c92c-41bf-806e-e0a61fb4811a",
        "scopeTimeLimit": "2019-11-16T11:28:50.000+02:00",
        "privilegeList": [
            {
                "accountNumber": "PL68146000095180629309555036",
                "ais:getAccount": {
                    "scopeUsageLimit": "multiple"
                }
            }
        ]
    }
}
```

W powyższym wywołaniu należy pamiętać, że:

- wartość pola `"exchange_token"` musi być równa wartości zwróconej w polu `"access_token"` po wywołaniu usługi `/token` pozyskującym token do realizacji usługi pierwotnej `/getAccounts`
- numery rachunków wskazane w wywołaniu, muszą zawierać się na liście otrzymanej z wywołania usługi `/getAccounts`
- data ważności zgody przekazana w polu `"scopeTimeLimit"` nie może przekraczać daty ważności zgody pierwotnej
- wartość pola `"consentId"` powinna zawierać nowy identyfikator zgody
- w polu `"scopeUsageLimit"`" można wskazać inną wartość niż w pierwotnej zgodzie

W celu modyfikacji zakresu tak uzyskanej zgody, np. usunięcia jednego z pozyskanych uprawnień lub usunięcia ze zgody jednego z rachunków należy ponownie wywołać usługę `/token` w trybie `exchangeToken` zgodnie z powyższymi wytycznymi z podaniem nowego zakresu zgody AIS. Poprzednio pozyskana zgoda AIS zostanie automatycznie unieważniona.

Należy pamiętać, że usunięcie pierwotnej zgody ais-accounts:getAccounts za pomocą usługi `/deleteConsent` spowoduje również usunięcie wszystkich zgód z uszczegółowionym zakresem wygenerowanych na jej podstawie.

Pobranie informacji o rachunku z wyborem rachunku po stronie ASPSP

W scenariuszu, gdy wybór rachunków dla których zostaną nadane uprawnienia jest dokonywany po stronie banku informacja o wybranych rachunkach przekazywana jest w odpowiedzi na wywołanie usługi `/token` w polu `"accountNumber"` będącym elementem struktury `"privilegeList"`. Przykładowa odpowiedź wygląda następująco:


```
{
    "responseHeader": {
        "requestId": "7aa46703-1cac-457e-b616-a50f9f514713",
        "sendDate": "2019-09-06T09:37:46.313+02:00",
        "isCallback": false
    },
    "access_token": "O3/D6+rfvDOTnUjMQx2EU6AQxGk=",
    "token_type": "bearer",
    "expires_in": "120",
    "refresh_token": "4oL6qdF86tHj48k2",
    "scope": "psd2-ais",
    "scope_details": {
        "privilegeList": [
            {
                "accountNumber": "PL78194000086704648427357299",
                "ais:getAccount": {
                    "scopeUsageLimit": "single"
                }
            }
        ],
        "consentId": "ffd4954c-e2c5-488d-b7e9-eeffad0c64ac",
        "scopeTimeLimit": "2019-10-06T11:28:50.000+02:00",
```

```
        "throttlingPolicy": "psd2Regulatory"
    }
}
```

Tak pozyskany numer rachunku należy później przekazać w wywołaniu docelowej usługi, na którą była wyrażona zgoda.

Odnowienie SCA dla zgody AIS

Możliwość odnowienia SCA dla zgody AIS została zrealizowana za pomocą usługi `/authorize`. Aby dokonać odnowienia zgody należy wywołać usługę `/authorize` podając w polu `"consentId"` identyfikator wcześniej udzielonej zgody bez podawania listy uprawnień. Na tej podstawie system wykrywa, że następuje odnowienie zgody. Poniżej zamieszczamy przykładowe wywołanie w tym trybie:

```
{
    "requestHeader": {
        "requestId": "86340673-b751-4558-8357-9fab21d91c8e",
        "tppId": "YYYYY-ZZZZ-TPPIdentificator",
        "userAgent": "SOAP-UI accounts.0-ais-getAccount-SCA",
        "isCompanyContext": false,
        "ipAddress": "127.0.0.1",
        "sendDate": "2019-10-04T11:16:56.536Z"
    },
    "response_type": "code",
    "client_id": "YYYYY-ZZZZ-TPPIdentificator",
    "redirect_uri": "http://example.com/",
    "state": "267a5f94-d32a-1860-9070-af37e3eb27de",
    "scope": "ais",
    "scope_details": {
        "scopeGroupType": "ais",
        "consentId": "0b5159d2-030f-4ad5-8591-0b489b935ade",
        "scopeTimeLimit": "2019-10-14T11:16:56.536Z",
        "throttlingPolicy": "psd2Regulatory"
    }
}
```

W powyższym wywołaniu należy pamiętać, że:

- wartość pola `"scopeTimeLimit"` nie powinna przekraczać daty podanej przy wyrażaniu zgody (w przypadku krótszej daty ważności zgoda jest aktualizowana)

Ponadto, odnowienie SCA dla zgody AIS którą uszczegółowiono za pomocą za pomocą usługi `/token` w trybie `exchangeToken` nie powoduje automatycznie odnowienia zgody podrzędnej.

Limity żądań

Standard PolishAPI nakłada ograniczenie wywołań usług AIS do max 4 wywołań bez udziału użytkownika w ciągu 24 godzin. Zliczanie zapytań realizowane jest na poziomie pojedynczego uprawnienia w zgodzie.

Dla przykładu, jeżeli zgoda zawiera uprawnienie ais:getTransactionsDone i ais:getAccount pozwoli na odpytanie 4 razy w ciągu 24 godzin o historię i 4 razy w ciągu 24h o szczegóły rachunku.

W związku z tym limity zapytań nie są weryfikowane podczas wydawania tokena, a dopiero podczas wywołania usługi biznesowej. Dlatego też oprócz ustawienia nagłówka `"is_user_session"` w wywołaniu usługi `/token`, kluczowa w tej kwestii jest zawartość nagłówka `"isDirectPsu"` w poszczególnych wywołaniach metod biznesowych.

Ograniczenia ważności zgód

Maksymalna ważność zgód typu PIS na zlecenie przelewu, paczki przelewów lub płatności cyklicznej została ograniczona do 15 minut.