



TEL-STER Sp. z o.o.
ul. Stefana Stefańskiego 23
62-002 Suchy Las
Tel. +48 61 628 97 50
Fax. +48 61 639 37 11



TelCOMM 5.0



TEL-STER Sp. z o.o.
ul. Stefana Stefańskiego 23
62-002 Suchy Las
Tel. +48 61 628 97 50
Fax. +48 61 639 37 11

Historia zmian dokumentu:

| Data | Wersja dokumentu | Autor | Opis zmiany |
|-------------|-------------------------|--------------------|-----------------------------------|
| 2019.04.17 | 4.0 | Michał Siatkowski | Wersja 4.0 |
| 2019.05.30 | 4.0 | Elżbieta Sobkowiak | Dostosowanie formy do wymogów ISO |
| 2019.12.18 | 4.2 | Michał Siatkowski | Dodatkowe funkcjonalności |
| 2020.07.07 | 4.5 | Michał Siatkowski | Jeden P-Mode dla obu kierunków |
| 2020.12.14 | 4.6 | Michał Siatkowski | Aktualizacje bezpieczeństwa |
| 2022.05.17 | 5.0 | Michał Siatkowski | Implementacja AS2 |

Spis treści

| | |
|--|----|
| Specyfikacja | 4 |
| Wprowadzenie | 6 |
| Architektura | 6 |
| Aplikacja interfejsowa | 6 |
| EKTRAN „WYŚLIJ” | 8 |
| EKTRAN „POBIERZ” | 9 |
| EKTRAN „ODEBRANE” | 11 |
| EKTRAN „WYSŁANE” | 12 |
| EKTRAN „EDIGAS” | 12 |
| EKTRAN „LOGI” | 13 |
| EKTRAN „PARTNERZY” | 15 |
| EKTRAN „[P-MODES]” | 15 |
| EKTRAN „UŻYTKOWNICY” | 20 |
| EKTRAN „BAZA DANYCH” | 21 |
| EKTRAN „OPCJE” | 21 |
| Automatyczna aktualizacja certyfikatów | 23 |
| Interfejs do współpracy z aplikacją zewnętrzną..... | 25 |
| Usługa umożliwiająca odbiór dokumentów od partnera | 28 |
| Udostępnianie danych | 29 |
| Załączniki..... | 30 |
| Materiały źródłowe..... | 30 |

Specyfikacja

- Protokoły: AS4, AS2 w wersji 1.1
- Wzorce komunikacji (MEP) AS4: One-Way/Push, Two-Way/Push-Pull, One-Way/Pull jako partner inicjujący
- Algorytmy AS4:
 - Funkcje skrótu dla podpisu (hash): sha256, sha384, sha512
 - Podpis cyfrowy:
 - certyfikat RSA: rsa-sha256, rsa-sha384, rsa-sha512
 - certyfikat ECC: ecdsa-sha256, ecdsa-sha384, ecdsa-sha512
 - Szyfrowanie danych:
 - aes128-cbc, aes192-cbc, aes256-cbc
 - aes128-gcm, aes192-gcm, aes256-gcm
 - Szyfrowanie klucza:
 - certyfikat RSA: rsa-oaep-mgf1p, rsa-oaep
 - MGF: mgf1sha1, mgf1sha256, mgf1sha384, mgf1sha512
 - Funkcje skrótu (hash): sha1, sha256, sha384, sha512
 - certyfikat ECC: ECDH-ES
 - KDF: ConcatKDF
 - KW: kw-aes128, kw-aes192, kw-aes256
 - HMAC: sha1, sha256, sha384, sha512
 - Kompresja: gzip
- Algorytmy AS2:
 - Podpis cyfrowy:
 - sha256, sha384, sha512
 - sha256-rsassa-pss, sha384-rsassa-pss, sha512-rsassa-pss
 - Szyfrowanie: aes128-cbc, aes192-cbc, aes256-cbc
 - Kompresja: zlib
- [ReplyPattern]: Response (synchroniczność), Callback (asynchroniczność, dla AS4)
- [SecurityTokenReference] (AS4): BinarySecurityToken X509v3, BinarySecurityToken X509PKIPathv1, IssuerAndSerialNumber



- Automatyczna aktualizacja certyfikatów między partnerami: tak, dla AS4
- Wiele załączników w komunikacie: tak, dla AS4
- Jeden [P-Mode] dla obu kierunków (gdy partnerzy używają tego samego certyfikatu do podpisu i szyfrowania): tak
- Walidacja certyfikatów: OCSP/CRL (opcjonalnie)
- Certyfikat TLS serwera i klienta: tak (opcjonalnie)
- Architektura: Microsoft Windows Server, IIS, .NET Framework 4.8, HTTPS, TLS 1.2 i 1.3
- Baza danych: SQLite, Oracle
- Interakcja z aplikacją: przeglądarka internetowa dla użytkowników, usługa internetowa (web service) dla aplikacji zewnętrznych
- Język w aplikacji: polski, angielski

Wprowadzenie

Oprogramowanie TelCOMM jest narzędziem umożliwiającym wymianę dokumentów typu B2B za pomocą protokołów AS4 i AS2 w wersji 1.1.

Protokół AS4 (Applicability Statement 4) to standard opisujący bezpieczne i niezawodne przesyłanie plików przez publiczną sieć Internet. Protokół ten bazuje na powszechnie znanych i sprawdzonych rozwiązaniach, takich jak protokoły HTTP, TLS, SOAP oraz usługach sieciowych (web service). Reprezentuje otwarty standard wymiany danych typu B2B opisany w specyfikacji OASIS ebMS 3.0. Elementami odpowiedzialnymi za bezpieczeństwo i wiarygodność przesyłanych danych są podpisy cyfrowe oraz mechanizmy szyfrujące (WS-security).

Protokół AS2 (Applicability Statement 2) powstał wcześniej niż AS4, bazuje na protokołach HTTP i TLS, funkcjonalnie podobny do AS4, ale w praktycznym użyciu został w większości przez niego zastąpiony i obecnie używany jest raczej w celach kompatybilności ze starszymi systemami.

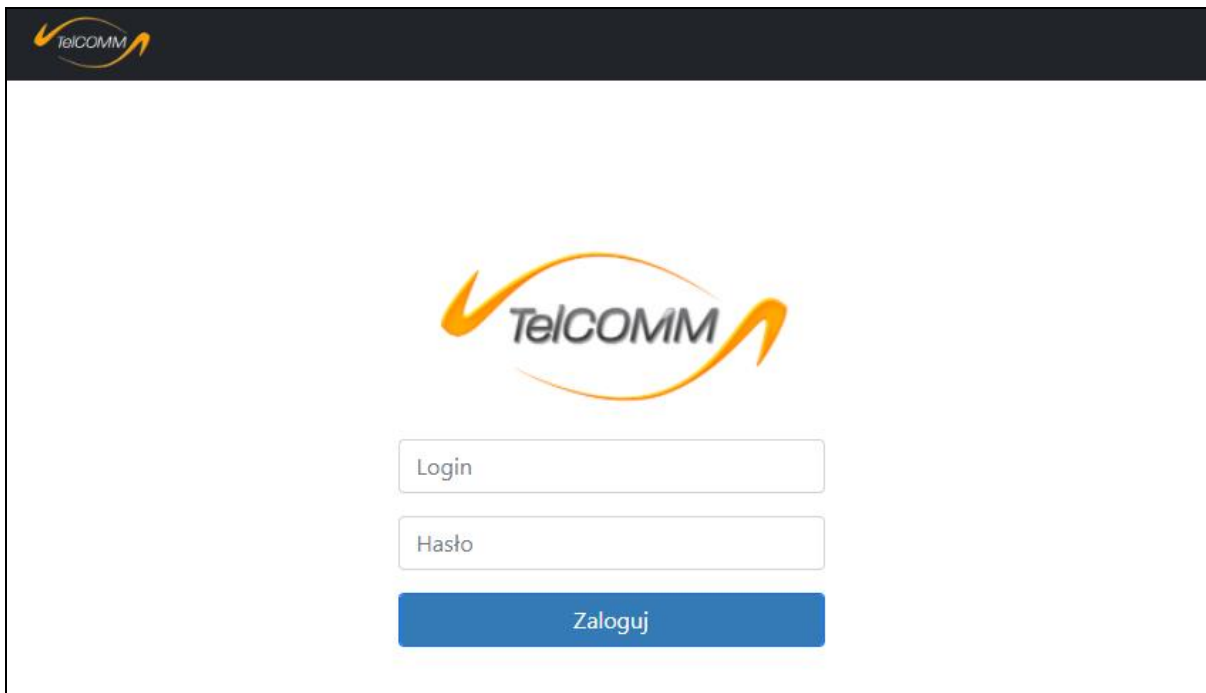
Architektura

Oprogramowanie TelCOMM jest rozwiązaniem adresowanym dla środowiska MS Windows Server, bazującym na podsystemie IIS oraz .NET Framework w wersji co najmniej 4.8. Aplikacja może być udostępniana jedynie w oparciu o protokół HTTPS, a wersja używanego protokołu TLS podczas wysyłania komunikatów to 1.2 lub 1.3. Korzysta ona z wbudowanej bazy danych **SQLite** lub ma możliwość korzystania z bazy danych **Oracle**. Program składa się z trzech komponentów:

- aplikacji interfejsowej dostępnej z poziomu przeglądarki internetowej, umożliwiającej konfigurację i interaktywną wymianę dokumentów,
- usługi internetowej /WebServices/Gateway.asmx umożliwiającej wymianę dokumentów z poziomu aplikacji zewnętrznej,
- usługi internetowej /MSH.asmx/Receive umożliwiającej odbiór dokumentów od partnerów.

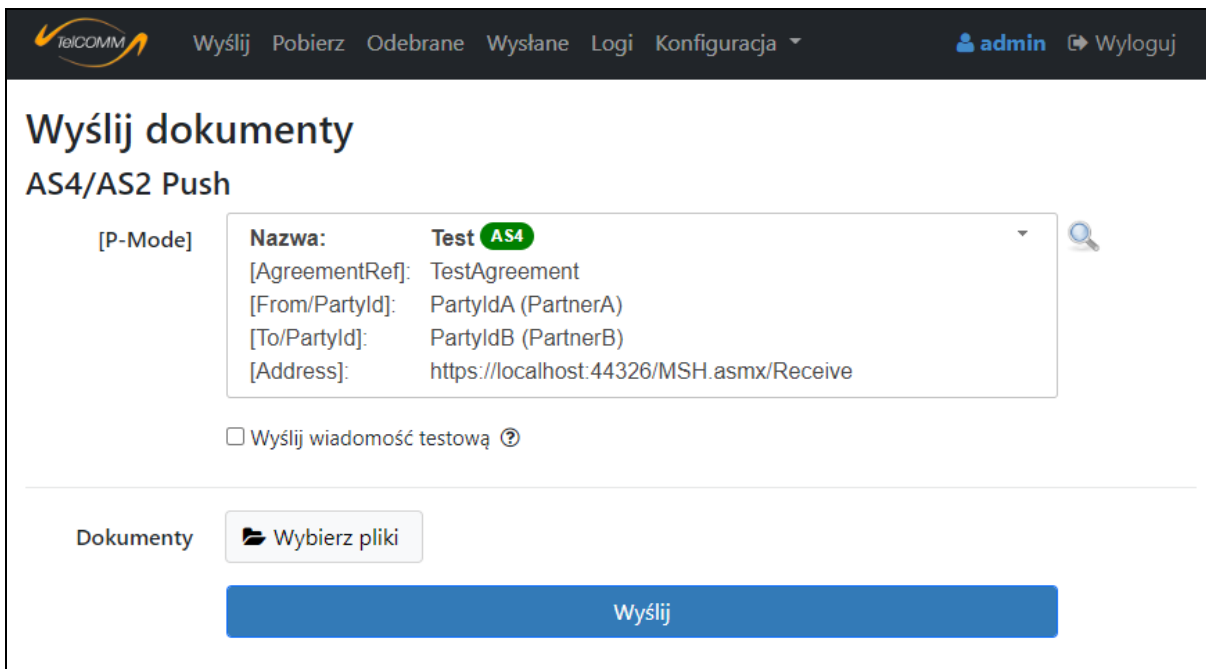
Aplikacja interfejsowa

Pierwszym ekranem aplikacji TelCOMM jest ekran logowania. Aby móc z niej korzystać należy się zalogować. W aplikacji istnieje nieusuwalny użytkownik, dla którego początkowy login i hasło to „admin”. Kliknięcie w ikonę TelCOMM w lewym górnym rogu aplikacji spowoduje wyświetlenie numeru wersji.



Rysunek 1. Ekran logowania

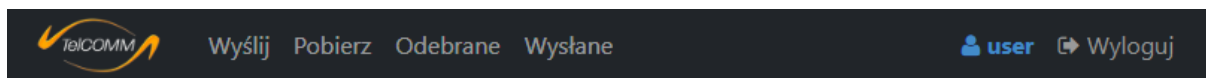
Po zalogowaniu pojawia się główny ekran aplikacji „Wyślij”.



Rysunek 2. Wygląd aplikacji po zalogowaniu

U góry ekranu znajduje się pasek nawigacyjny aplikacji wspólny dla ekranów po zalogowaniu. Po ikonie aplikacji znajduje się hiperłącze do ekranu „Wyślij” oraz „Pobierz”, a następnie do pozostałych ekranów aplikacji. Na końcu znajduje się nazwa zalogowanego użytkownika wraz z opcją wylogowania. Pasek ten różni się w zależności od tego czy zalogowany użytkownik jest

administratorem czy nie. Zwykły użytkownik ma dostęp do ekranów „Wyślij”, „Pobierz”, „Odebrane” i „Wysłane”. Poniżej zrzut ekranu paska nawigacyjnego dla zwykłego użytkownika.



Rysunek 3. Pasek nawigacyjny użytkownika

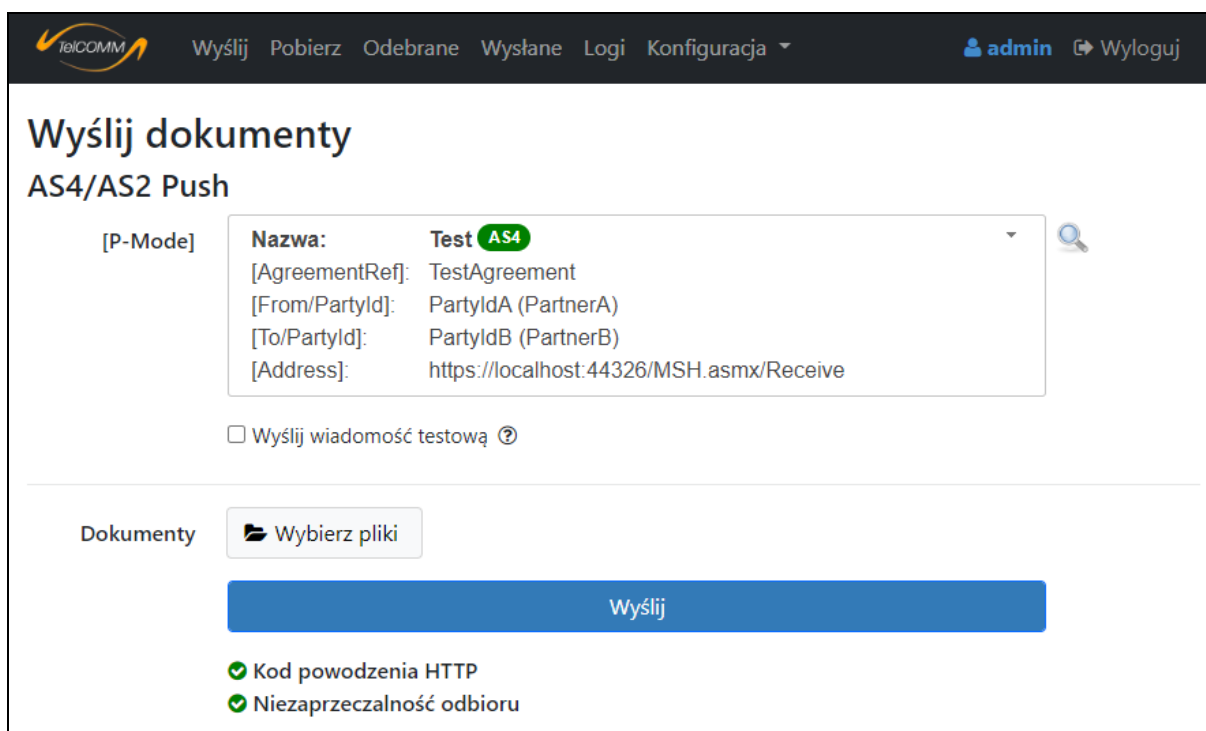
EKRAN „WYŚLIJ”

Służy do wysyłania plików za pomocą wzorca komunikacji **One-Way/Push**. Polega on na jednostronnym wysłaniu wiadomości od jednego partnera do drugiego. W celu wysłania komunikatu AS4/AS2 należy:

- wybrać wcześniej zdefiniowany [P-Mode] ([Processing mode]), w którym to zawarty jest komplet informacji na temat połączenia komunikacyjnego między partnerami,
- w polu „Dokumenty” wybrać plik lub pliki, które mają zostać wysłane.

Kliknięcie przycisku „Wyślij” inicjuje komunikację - dokumenty zostają wysłane zgodnie z protokołem komunikacyjnym AS4/AS2 oraz następuje analiza odpowiedzi od odbiorcy pod kątem:

- braku błędów wysłania wiadomości – kod powodzenia HTTP,
- niezaprzeczalności odbioru (Non Repudiation of Receipt) lub potwierdzenia odbioru (Reception Awareness), gdy odpowiedź przesyłana jest synchronicznie (Response) – możliwe jest również asynchroniczne odebranie odpowiedzi w osobnym komunikacie (Callback, dla AS4).



Rysunek 4. Ekran „Wyślij” – po analizie wysyłki

W przypadku otrzymania w odpowiedzi błędów komunikacji AS4/AS2 zostaną one wyświetlone na dole ekranu i zapisane w celu późniejszego podglądu na ekranie „Logi”.

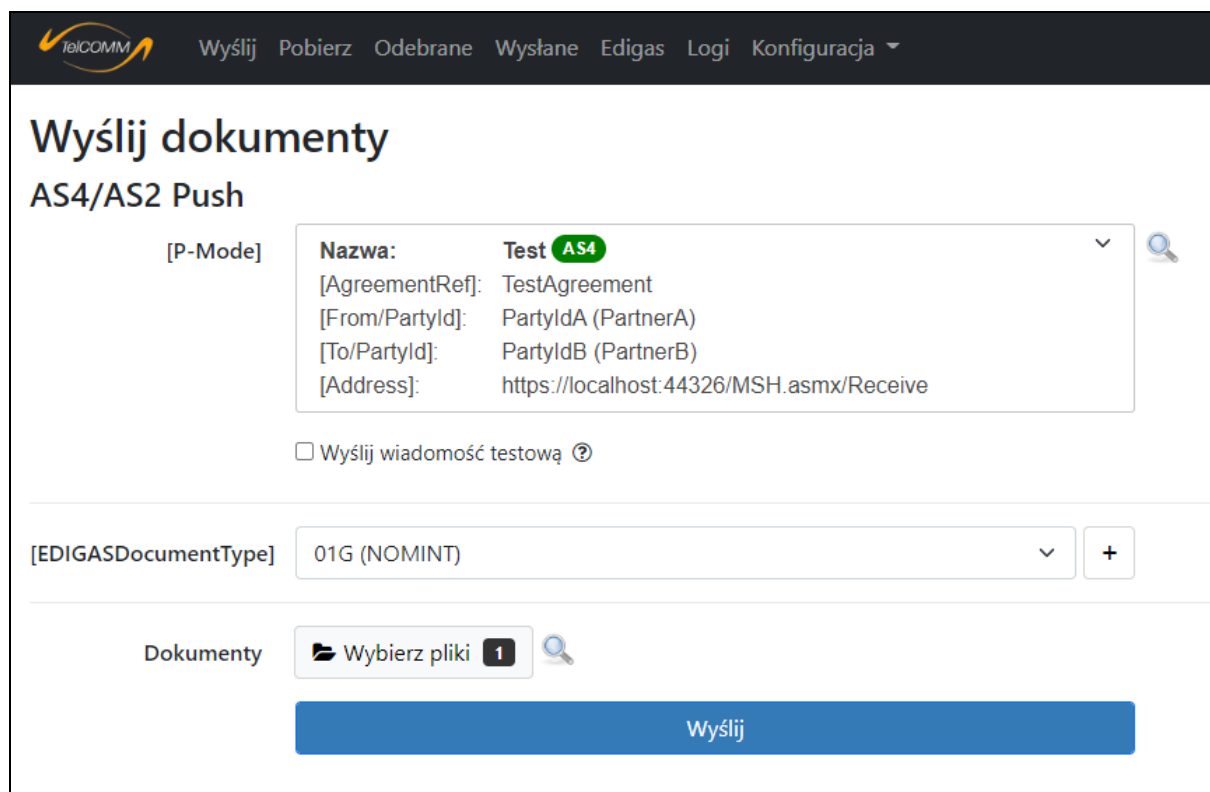
Przydatnymi opcjami ekranu wysyłania są:

- możliwość wysłania wiadomości z testowymi parametrami komunikacji AS4 w celu przetestowania połączenia między partnerami – należy zaznaczyć opcję „Wyślij wiadomość testową”,
- w sytuacji wysyłania wielu plików, możliwość wysłania ich pojedynczo w osobnych wiadomościach – opcja pojawi się po wybraniu więcej niż jednego pliku.

W przypadku włączenia w konfiguracji aplikacji opcji „Wysyłanie dokumentów EDIGAS” dodatkowymi wariantami będzie:

- możliwość niewypełniania pola [EDIGASDocumentType], lecz pobrania go z załączonych plików Edig@s – opcja „Pobierz z załączonych dokumentów” w rozwijanej liście,
- kryjące się pod przyciskiem „+” dodatkowe pola „Identyfikator transakcji” i „Opis transakcji” służące do opisanie wysłanych nominacji jako pojedynczej transakcji zawierającej jeden lub wiele plików. Umożliwia to opcjonalne grupowanie nominacji na ekranie „Edigas”.

Dodatkowo w trybie „Wysyłanie dokumentów EDIGAS” występuje możliwość wysłania plików niebędących dokumentami Edig@s – w polu [EDIGASDocumentType] należy wybrać „Brak”. Przykład ekranu w trybie „Wysyłanie dokumentów EDIGAS” przedstawia poniższy rysunek.



Rysunek 5. Ekran „Wyślij” w trybie “Wysyłanie dokumentów EDIGAS”

EKRAN „POBIERZ”

Służy do pobierania danych od partnera z użyciem wzorca komunikacji **One-Way/Pull** lub **Two-Way/Push-Pull**, który składa się z dwóch etapów. W pierwszym etapie wysyłany jest dokument z żądaniem określającym jakie dane partner ma udostępnić, a w drugim etapie dane te są w sposób automatyczny pobierane. One-Way/Pull obejmuje tylko drugi etap – utworzenie żądania Pull i dodanie go do puli żądań realizowanych automatycznie. W celu pobrania danych AS4 należy:

- wybrać wcześniej zdefiniowany [P-Mode] ([Processing mode]), w którym to zawarty jest komplet informacji na temat połączenia komunikacyjnego między partnerami,

- w polu „Dokumenty” wybrać plik lub pliki zawierające żądanie o udostępnienie konkretnych danych. Struktura pliku jest ustalana między partnerami.

The screenshot shows the 'Pobierz dokumenty' (Download documents) interface for AS4 Pull. The form includes a search bar with the name 'TestPushPull AS4'. Below the search bar, the following fields are populated: [AgreementRef]: TestAgreementPushPull, [From/PartyId]: PartyIdA (PartnerA), [To/PartyId]: PartyIdB (PartnerB), and [Address]: https://localhost:44326/MSH.asmx/Receive. There is an unchecked checkbox for 'One-Way/Pull'. At the bottom, there is a 'Dokumenty' section with a 'Wybierz pliki' (Select files) button and a large blue 'Wyślij' (Send) button.

Rysunek 6. Ekran „Pobierz”

Kliknięcie przycisku „Wyślij” inicjuje pierwszą fazę komunikacji – plik z żądaniem zostaje wysłany zgodnie z protokołem komunikacyjnym AS4 oraz następuje analiza odpowiedzi od odbiorcy pod kątem braku błędów wysłania wiadomości – kod powodzenia HTTP.

This screenshot is identical to the previous one, but it includes a green checkmark icon and the text 'Kod powodzenia HTTP' (HTTP success code) at the bottom of the form, indicating a successful analysis of the outgoing message.

Rysunek 7. Ekran „Pobierz” – po analizie wysyłki

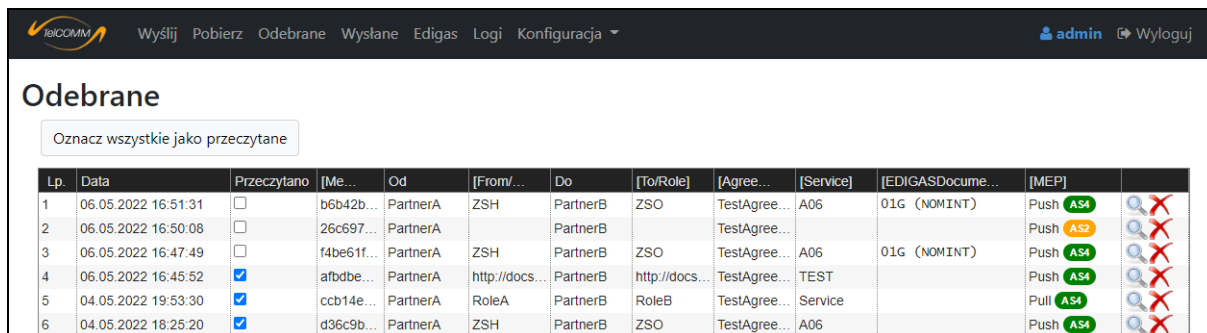
W przypadku otrzymania w odpowiedzi błędów komunikacji AS4 zostaną one wyświetlone na dole ekranu pobierania i zapisane w celu późniejszego podglądu na ekranie „Logi”.

Gdy będzie zaznaczona opcja „One-Way/Pull” wykonane zostanie tylko pobranie danych bez uprzedniego wysłania żądania. Wtedy nie ma możliwości dodania dokumentów.

Dodatkową opcją ekranu pobierania jest, w przypadku wysyłania wielu żądań, możliwość wysłania ich pojedynczo w osobnych wiadomościach – opcja pojawi się po wybraniu więcej niż jednego pliku.


EKRAN „ODEBRANE”

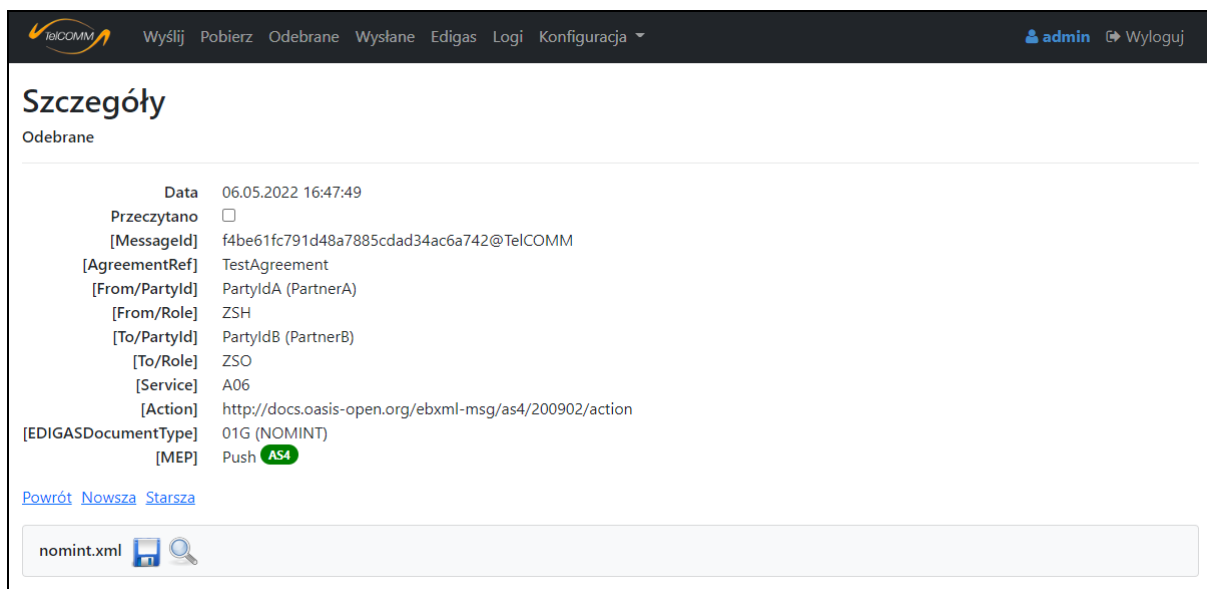
Ekran „Odebrane” spełnia rolę skrzynki odbiorczej aplikacji. Dla każdej odebranej wiadomości/żądania istnieje możliwość nadania jej statusu przeczytanej/nieprzeczytanej – funkcjonalność ta ma przede wszystkim znaczenie przy korzystaniu z aplikacji przez funkcję usługi internetowej (web service), gdzie pobierane są wszystkie nieprzeczytane pozycje. Można również oznaczyć wszystkie pozycje jako przeczytane za pomocą przycisku na górze ekranu.



| Lp. | Data | Przeczytano | [Me...] | Od | [From...] | Do | [To/Role] | [Agree...] | [Service] | [EDIGASDocume...] | [MEP] | |
|-----|---------------------|-------------------------------------|------------|----------|----------------|----------|----------------|--------------|-----------|-------------------|-------|-----|
| 1 | 06.05.2022 16:51:31 | <input type="checkbox"/> | b6b42b... | PartnerA | ZSH | PartnerB | ZSO | TestAgree... | A06 | 01G (NOMINT) | Push | AS4 |
| 2 | 06.05.2022 16:50:08 | <input type="checkbox"/> | 26c697... | PartnerA | | PartnerB | | TestAgree... | | | Push | AS2 |
| 3 | 06.05.2022 16:47:49 | <input type="checkbox"/> | f4be61f... | PartnerA | ZSH | PartnerB | ZSO | TestAgree... | A06 | 01G (NOMINT) | Push | AS4 |
| 4 | 06.05.2022 16:45:52 | <input checked="" type="checkbox"/> | afdbde... | PartnerA | http://docs... | PartnerB | http://docs... | TestAgree... | TEST | | Push | AS4 |
| 5 | 04.05.2022 19:53:30 | <input checked="" type="checkbox"/> | ccb14e... | PartnerA | RoleA | PartnerB | RoleB | TestAgree... | Service | | Pull | AS4 |
| 6 | 04.05.2022 18:25:20 | <input checked="" type="checkbox"/> | d36c9b... | PartnerA | ZSH | PartnerB | ZSO | TestAgree... | A06 | | Push | AS4 |


Rysunek 8. Ekran „Odebrane”

W obrębie aplikacji użyto zestawu ikon , które odpowiednio służą do dodawania, podglądu szczegółów, edycji i usuwania danego elementu. Aby wyświetlić przesłane pliki w danym komunikacie AS4/AS2 należy przejść do szczegółów komunikatu. Można również przełączać się między pozycjami bezpośrednio na ekranie szczegółów korzystając z hiperłączy „Nowsza” i „Starsza”.



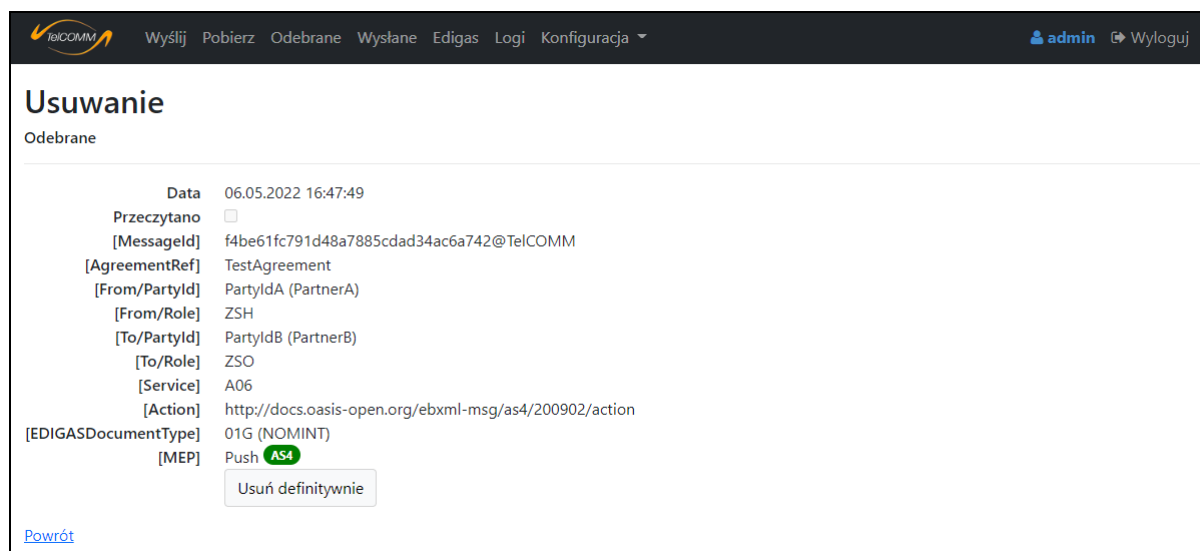
| | |
|----------------------|--|
| Data | 06.05.2022 16:47:49 |
| Przeczytano | <input type="checkbox"/> |
| [MessageId] | f4be61fc791d48a7885cdad34ac6a742@TelCOMM |
| [AgreementRef] | TestAgreement |
| [From/PartyId] | PartyIdA (PartnerA) |
| [From/Role] | ZSH |
| [To/PartyId] | PartyIdB (PartnerB) |
| [To/Role] | ZSO |
| [Service] | A06 |
| [Action] | http://docs.oasis-open.org/ebxml-msg/as4/200902/action |
| [EDIGASDocumentType] | 01G (NOMINT) |
| [MEP] | Push AS4 |

[Powrót](#) [Nowsza](#) [Starsza](#)

nomint.xml 

Rysunek 9. Ekran „Odebrane” – szczegóły komunikatu

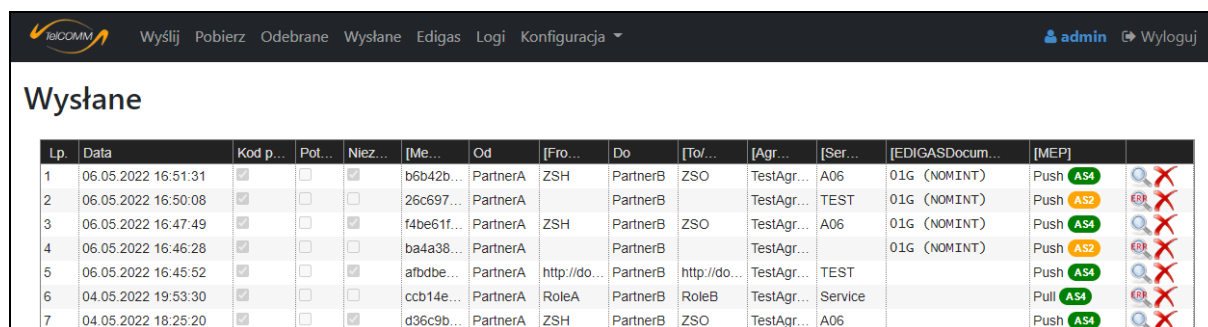
Na ekranie „Odebrane” istnieje również opcja usunięcia wiadomości. Kliknięcie przycisku powoduje wyświetlenie ekranu usuwania danego komunikatu i dopiero na nim należy potwierdzić jej usunięcie przyciskiem „Usuń definitywnie”, aby odpowiedni wpis z bazy danych został trwale usunięty. Taka procedura usuwania, polegająca na potwierdzeniu na osobnym ekranie, jest stosowana w obrębie całej aplikacji.



Rysunek 10. Ekran „Odebrane” – usuwanie wiadomości

EKRAN „WYŚLANE”

Ekran „Wysłane” wyświetla wszystkie komunikaty, które zostały wysłane, a jego struktura jest analogiczna do ekranu „Odebrane”, z tą różnicą, że zamiast opcji nadania statusu wiadomości przeczytanej/nieprzeczytanej znajdują się informacje dotyczące wyniku analizy odpowiedzi od odbiorcy, opisane wcześniej. Również w przypadku wysłania wiadomości z błędami pojawia się nieco zmodyfikowana ikona podglądu szczegółów wiadomości widoczna na poniższym rysunku.






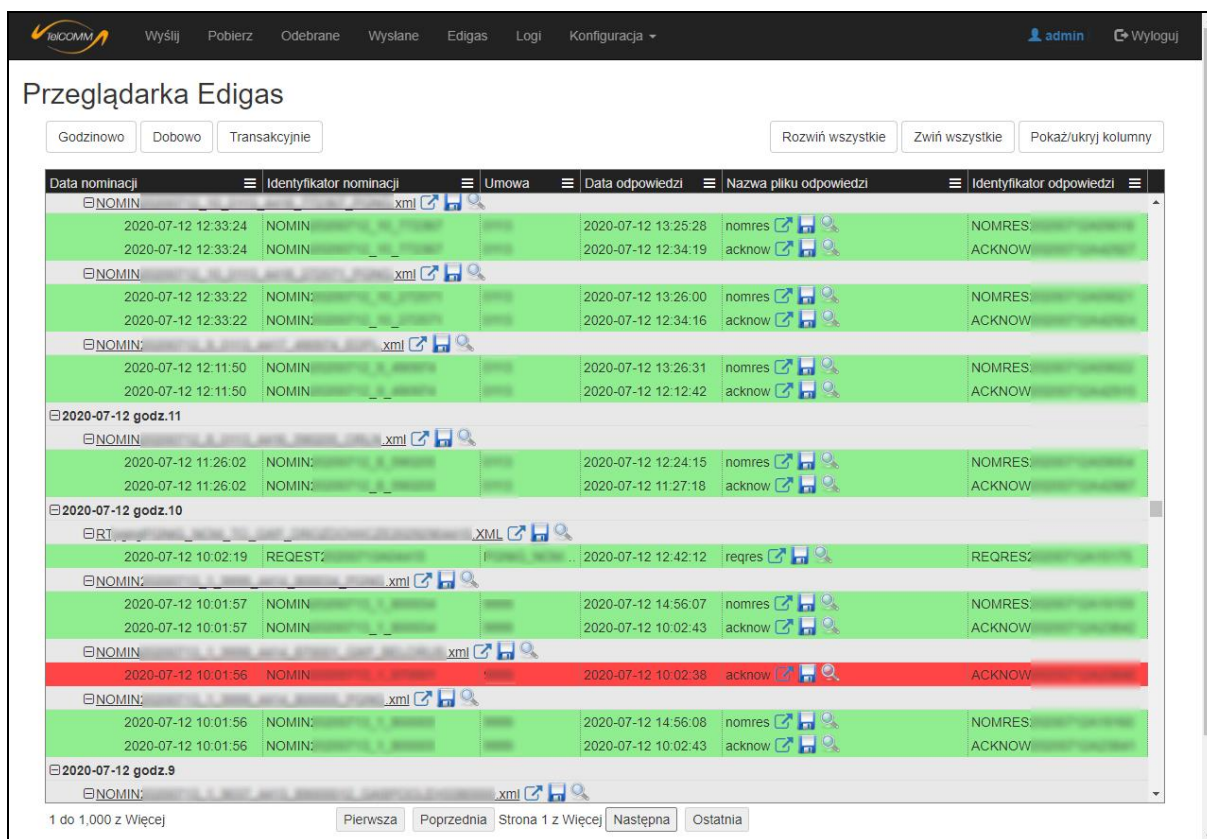
| Lp. | Data | Kod p... | Pot... | Niez... | [Me... | Od | [Fro... | Do | [To/... | [Agr... | [Ser... | [EDIGASDocum... | [MEP] | |
|-----|---------------------|----------|--------------------------|-------------------------------------|------------|----------|-------------|----------|-------------|------------|---------|-----------------|----------|--|
| 1 | 06.05.2022 16:51:31 | ✓ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | b6b42b... | PartnerA | ZSH | PartnerB | ZSO | TestAgr... | A06 | 01G (NOMINT) | Push AS4 | |
| 2 | 06.05.2022 16:50:08 | ✓ | <input type="checkbox"/> | <input type="checkbox"/> | 26c697... | PartnerA | | PartnerB | | TestAgr... | TEST | 01G (NOMINT) | Push AS2 | |
| 3 | 06.05.2022 16:47:49 | ✓ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | f4be61f... | PartnerA | ZSH | PartnerB | ZSO | TestAgr... | A06 | 01G (NOMINT) | Push AS4 | |
| 4 | 06.05.2022 16:46:28 | ✓ | <input type="checkbox"/> | <input type="checkbox"/> | ba4a38... | PartnerA | | PartnerB | | TestAgr... | | 01G (NOMINT) | Push AS2 | |
| 5 | 06.05.2022 16:45:52 | ✓ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | afbdbc... | PartnerA | http//do... | PartnerB | http//do... | TestAgr... | TEST | | Push AS4 | |
| 6 | 04.05.2022 19:53:30 | ✓ | <input type="checkbox"/> | <input type="checkbox"/> | ccb14e... | PartnerA | RoleA | PartnerB | RoleB | TestAgr... | Service | | Pull AS4 | |
| 7 | 04.05.2022 18:25:20 | ✓ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | d36c9b... | PartnerA | ZSH | PartnerB | ZSO | TestAgr... | A06 | | Push AS4 | |

Rysunek 11. Ekran „Wysłane”

EKRAN „EDIGAS”

Ekran „Edigas” dostępny jest tylko przy zaznaczonej opcji „Wysyłanie dokumentów EDIGAS” w konfiguracji aplikacji. Pozwala on przeglądać wysłane oraz odebrane dokumenty Edig@s na zasadzie parowania. Funkcja parowania pozwala powiązać dokument NOMINT z ACKNOW i NOMRES oraz REQUEST z REQRES w sposób, jaki przedstawiono na poniższym rysunku. Dokumenty odpowiedzi wyróżnione są kolorem w zależności od tego czy mają status zatwierdzenia (kolor zielony) czy odrzucenia (czerwony). Dokumenty można posortować według informacji zawartych w pliku (Identyfikator i nr Umowy). Przyciski „Godzinowo”, „Dobowo” i „Transakcyjnie” pozwalają pogrupować dokumenty wg. godziny, daty lub według „Identyfikatora” i „Opisu Transakcji” (jeżeli

podano podczas wysłania). Przycisk „Rozwiń wszystkie” pozwala pokazać na ekranie wszystkie dokumenty, natomiast „Zwiń wszystkie” ukrywa dokumenty, dając możliwość wyświetlenia dokumentów tylko z wybranej grupy. Domyślnie nie wszystkie kolumny są widoczne na ekranie - opcja „Pokaż/ukryj kolumny” pozwala na zarządzanie widocznością kolumn. Kliknięcie na przycisk  pozwala zapisać dokument na komputerze, przycisk  pozwala podejrzeć zawartość pliku w karcie przeglądarki, a przycisk  to hiperłącze do ekranu szczegółów wiadomości, w której został przesłany dany dokument.




The screenshot shows the 'Przeglądarka Edigas' interface. At the top, there are navigation buttons: 'Wyślij', 'Pobierz', 'Odebrane', 'Wysłane', 'Edigas', 'Logi', and 'Konfiguracja'. The user is logged in as 'admin'. Below the navigation bar, there are filter buttons for 'Godzinowo', 'Dobowo', and 'Transakcyjnie', and action buttons for 'Rozwiń wszystkie', 'Zwiń wszystkie', and 'Pokaż/ukryj kolumny'. The main area contains a table with the following columns: 'Data nominacji', 'Identyfikator nominacji', 'Umowa', 'Data odpowiedzi', 'Nazwa pliku odpowiedzi', and 'Identyfikator odpowiedzi'. The table lists several records, including 'NOMIN' and 'acknow' entries, with dates and times. A red row is highlighted, indicating an error or specific status. At the bottom, there are pagination controls: '1 do 1,000 z Więcej', 'Pierwsza', 'Poprzednia', 'Strona 1 z Więcej', 'Następna', and 'Ostatnia'.

Rysunek 12. Ekran „Edigas”

EKRAN „LOGI”

Kolejne ekrany dostępne są jedynie dla administratora. Ekran „Logi” zawiera błędy komunikacji AS4/AS2 oraz wybrane informacje na temat działań użytkowników. Dwukrotne kliknięcie w kolumnie „Treść” pozwala wyświetlić pełną treść komunikatu.

 Wyślij Pobierz Odebrane Wysłane Edigas Logi Konfiguracja
admin Wyloguj

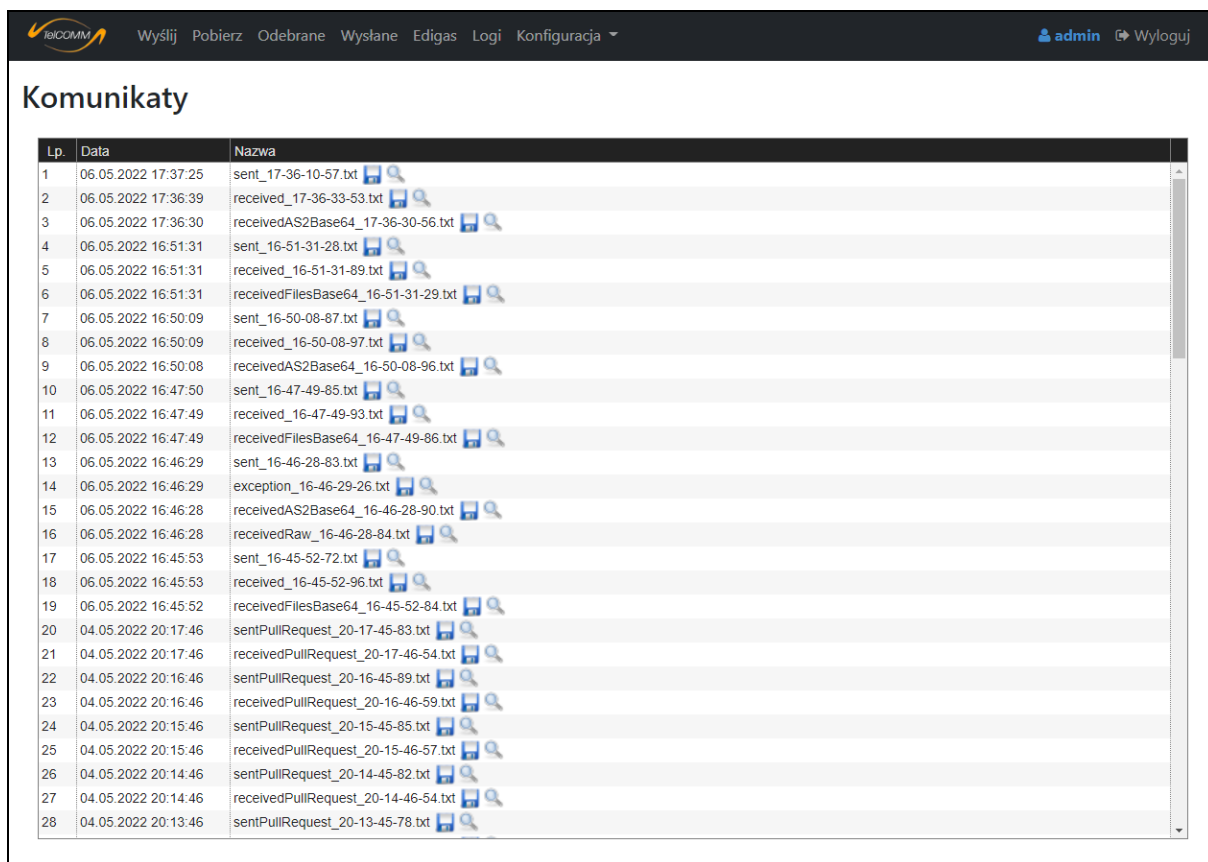
Logi

Komunikaty

| Lp. | Data | Treść |
|-----|---------------------|--|
| 1 | 06.05.2022 17:07:32 | [admin, PModes/Edit]: Edycja [P-Mode]. ([Agreement]: TestAgreementPushPull, [Initiator.Party]: PartyIdA, [Responder.Party]: PartyIdB, [Initiator.Role]: Role... |
| 2 | 06.05.2022 17:03:59 | [admin, PModes/Edit]: Edycja [P-Mode]. ([Agreement]: TestAgreement, [Initiator.Party]: PartyIdA, [Responder.Party]: PartyIdB, [Initiator.Role]: ZSH, [Respo... |
| 3 | 06.05.2022 16:50:40 | [admin, Received/MarkAsRead]: Oznaczenie wiadomości odebranej: Przeczytano = True, Data: 06.05.2022 16:45:52, [MessageId]: afbdbeb4074c4cde928... |
| 4 | 06.05.2022 16:49:53 | [admin, PModes/Edit]: Edycja [P-Mode]. ([Agreement]: TestAgreementAS2, [Initiator.Party]: PartyIdA, [Responder.Party]: PartyIdB, [Initiator.Role]: http://doc... |
| 5 | 06.05.2022 16:48:50 | [admin, Received/MarkAsRead]: Oznaczenie wiadomości odebranej: Przeczytano = True, Data: 04.05.2022 18:25:20, [MessageId]: d36c9bac70214f9a888... |
| 6 | 06.05.2022 16:48:49 | [admin, Received/MarkAsRead]: Oznaczenie wiadomości odebranej: Przeczytano = True, Data: 04.05.2022 19:53:30, [MessageId]: ccb14e7222d14c8eb0... |
| 7 | 06.05.2022 16:46:29 | [admin, AS2/Send]: Wiadomość dotarła do partnera. Szczegóły przetwarzania wiadomości przez partnera: [processed/error: authentication-failed] (An error... |
| 8 | 06.05.2022 16:46:29 | [admin, AS2SigningPss/CheckSignature]: Failed to verify digital signature: certificate expired on 2017053111403GMT+00:00 Inner Message: certificate ex... |
| 9 | 06.05.2022 16:46:29 | [-, ExtractPayload]: Sprawdzenie podpisu wiadomości AS2 dało wynik negatywny. ([SenderPartyId]: PartyIdA, [ReceiverPartyId]: PartyIdB) StackTrace: w T... |
| 10 | 06.05.2022 16:46:29 | [-, AS2SigningPss/CheckSignature]: Failed to verify digital signature: certificate expired on 20170510140547GMT+00:00 Inner Message: certificate expired... |
| 11 | 05.05.2022 19:06:34 | [-, Application_Error]: Not found StackTrace: w TelCOMM.Models.CustomDefaultControllerFactory.CreateController(RequestContext requestContext, String... |
| 12 | 06.05.2022 20:50:29 | [-, Application_Error]: Not found StackTrace: w TelCOMM.Models.CustomDefaultControllerFactory.CreateController(RequestContext requestContext, String... |
| 13 | 04.05.2022 20:17:46 | [-, GetResponse]: Serwer zdalny zwrócił błąd: (400) Złe żądanie. StackTrace: w System.Net.HttpWebRequest.GetResponse() w TelCOMM.Models.AS4.Pu... |
| 14 | 04.05.2022 20:17:46 | [-, Żądanie Pull zakończyło działanie niepowodzeniem. (Data: 04.05.2022 20:07:47, [SenderPartyId]: PartyIdA, [ReceiverPartyId]: PartyIdB, [MPC]: http://l... |
| 15 | 04.05.2022 20:17:46 | [-, SecurityDecryptAndVerifySign]: Sprawdzenie podpisu wiadomości dało wynik negatywny. ([AgreementRef]: -, [SenderPartyId]: PartyIdB, [ReceiverParty]... |
| 16 | 04.05.2022 20:17:45 | [-, CompareReceivedCerts_BinarySecurityToken]: Niezgodność certyfikatu podpisu przesłanego w wiadomości z ustalonym w [P-Mode]. ([Agreement]: Tes... |
| 17 | 04.05.2022 20:16:46 | [-, GetResponse]: Serwer zdalny zwrócił błąd: (400) Złe żądanie. StackTrace: w System.Net.HttpWebRequest.GetResponse() w TelCOMM.Models.AS4.Pu... |
| 18 | 04.05.2022 20:16:46 | [-, SecurityDecryptAndVerifySign]: Sprawdzenie podpisu wiadomości dało wynik negatywny. ([AgreementRef]: -, [SenderPartyId]: PartyIdB, [ReceiverParty]... |
| 19 | 04.05.2022 20:16:45 | [-, CompareReceivedCerts_BinarySecurityToken]: Niezgodność certyfikatu podpisu przesłanego w wiadomości z ustalonym w [P-Mode]. ([Agreement]: Tes... |
| 20 | 04.05.2022 20:15:46 | [-, GetResponse]: Serwer zdalny zwrócił błąd: (400) Złe żądanie. StackTrace: w System.Net.HttpWebRequest.GetResponse() w TelCOMM.Models.AS4.Pu... |
| 21 | 04.05.2022 20:15:46 | [-, SecurityDecryptAndVerifySign]: Sprawdzenie podpisu wiadomości dało wynik negatywny. ([AgreementRef]: -, [SenderPartyId]: PartyIdB, [ReceiverParty]... |
| 22 | 04.05.2022 20:15:45 | [-, CompareReceivedCerts_BinarySecurityToken]: Niezgodność certyfikatu podpisu przesłanego w wiadomości z ustalonym w [P-Mode]. ([Agreement]: Tes... |
| 23 | 04.05.2022 20:14:46 | [-, GetResponse]: Serwer zdalny zwrócił błąd: (400) Złe żądanie. StackTrace: w System.Net.HttpWebRequest.GetResponse() w TelCOMM.Models.AS4.Pu... |
| 24 | 04.05.2022 20:14:46 | [-, SecurityDecryptAndVerifySign]: Sprawdzenie podpisu wiadomości dało wynik negatywny. ([AgreementRef]: -, [SenderPartyId]: PartyIdB, [ReceiverParty]... |
| 25 | 04.05.2022 20:14:45 | [-, CompareReceivedCerts_BinarySecurityToken]: Niezgodność certyfikatu podpisu przesłanego w wiadomości z ustalonym w [P-Mode]. ([Agreement]: Tes... |
| 26 | 04.05.2022 20:13:46 | [-, GetResponse]: Serwer zdalny zwrócił błąd: (400) Złe żądanie. StackTrace: w System.Net.HttpWebRequest.GetResponse() w TelCOMM.Models.AS4.Pu... |
| 27 | 04.05.2022 20:13:46 | [-, SecurityDecryptAndVerifySign]: Sprawdzenie podpisu wiadomości dało wynik negatywny. ([AgreementRef]: -, [SenderPartyId]: PartyIdB, [ReceiverParty]... |
| 28 | 04.05.2022 20:13:45 | [-, CompareReceivedCerts_BinarySecurityToken]: Niezgodność certyfikatu podpisu przesłanego w wiadomości z ustalonym w [P-Mode]. ([Agreement]: Tes... |

Rysunek 13. Ekran „Logi”

Z ekranu „Logi” można przejść na dodatkowy ekran „Komunikaty”, na którym istnieje możliwość podejrzenia całych żądań i odpowiedzi HTTP, które były wysyłane i odbierane przez aplikację. Ekran służy szczegółowej analizie surowych komunikatów AS4 i AS2.

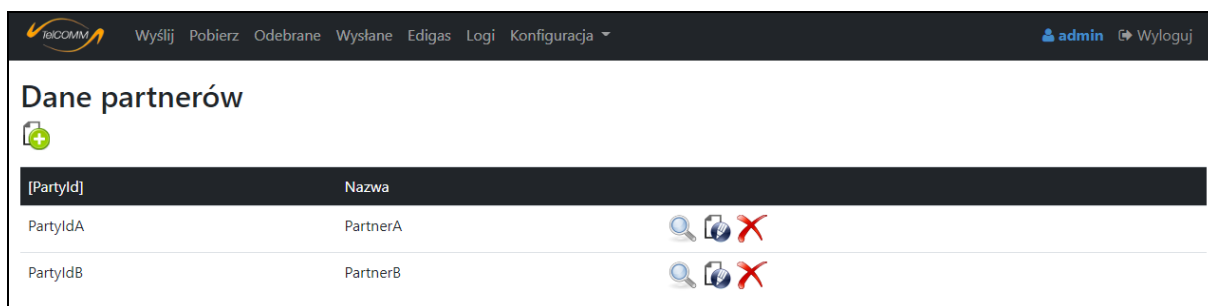


| Lp. | Data | Nazwa |
|-----|---------------------|-------------------------------------|
| 1 | 06.05.2022 17:37:25 | sent_17-36-10-57.txt |
| 2 | 06.05.2022 17:36:39 | received_17-36-33-53.txt |
| 3 | 06.05.2022 17:36:30 | receivedAS2Base64_17-36-30-56.txt |
| 4 | 06.05.2022 16:51:31 | sent_16-51-31-28.txt |
| 5 | 06.05.2022 16:51:31 | received_16-51-31-89.txt |
| 6 | 06.05.2022 16:51:31 | receivedFilesBase64_16-51-31-29.txt |
| 7 | 06.05.2022 16:50:09 | sent_16-50-08-87.txt |
| 8 | 06.05.2022 16:50:09 | received_16-50-08-97.txt |
| 9 | 06.05.2022 16:50:08 | receivedAS2Base64_16-50-08-96.txt |
| 10 | 06.05.2022 16:47:50 | sent_16-47-49-85.txt |
| 11 | 06.05.2022 16:47:49 | received_16-47-49-93.txt |
| 12 | 06.05.2022 16:47:49 | receivedFilesBase64_16-47-49-86.txt |
| 13 | 06.05.2022 16:46:29 | sent_16-46-28-83.txt |
| 14 | 06.05.2022 16:46:29 | exception_16-46-29-26.txt |
| 15 | 06.05.2022 16:46:28 | receivedAS2Base64_16-46-28-90.txt |
| 16 | 06.05.2022 16:46:28 | receivedRaw_16-46-28-84.txt |
| 17 | 06.05.2022 16:45:53 | sent_16-45-52-72.txt |
| 18 | 06.05.2022 16:45:53 | received_16-45-52-96.txt |
| 19 | 06.05.2022 16:45:52 | receivedFilesBase64_16-45-52-84.txt |
| 20 | 04.05.2022 20:17:46 | sentPullRequest_20-17-45-83.txt |
| 21 | 04.05.2022 20:17:46 | receivedPullRequest_20-17-46-54.txt |
| 22 | 04.05.2022 20:16:46 | sentPullRequest_20-16-45-89.txt |
| 23 | 04.05.2022 20:16:46 | receivedPullRequest_20-16-46-59.txt |
| 24 | 04.05.2022 20:15:46 | sentPullRequest_20-15-45-85.txt |
| 25 | 04.05.2022 20:15:46 | receivedPullRequest_20-15-46-57.txt |
| 26 | 04.05.2022 20:14:46 | sentPullRequest_20-14-45-82.txt |
| 27 | 04.05.2022 20:14:46 | receivedPullRequest_20-14-46-54.txt |
| 28 | 04.05.2022 20:13:46 | sentPullRequest_20-13-45-78.txt |

Rysunek 14. Ekran „Komunikaty”

EKRAN „PARTNERZY”

Na ekranie „Konfiguracja->Partnerzy” istnieje możliwość powiązania wartości [PartyId] z przyjazną nazwą, która będzie wyświetlana w obrębie aplikacji zamiast identyfikatora partnera. W komunikacji Edig@s dla AS4 identyfikatorem tym jest kod EIC.

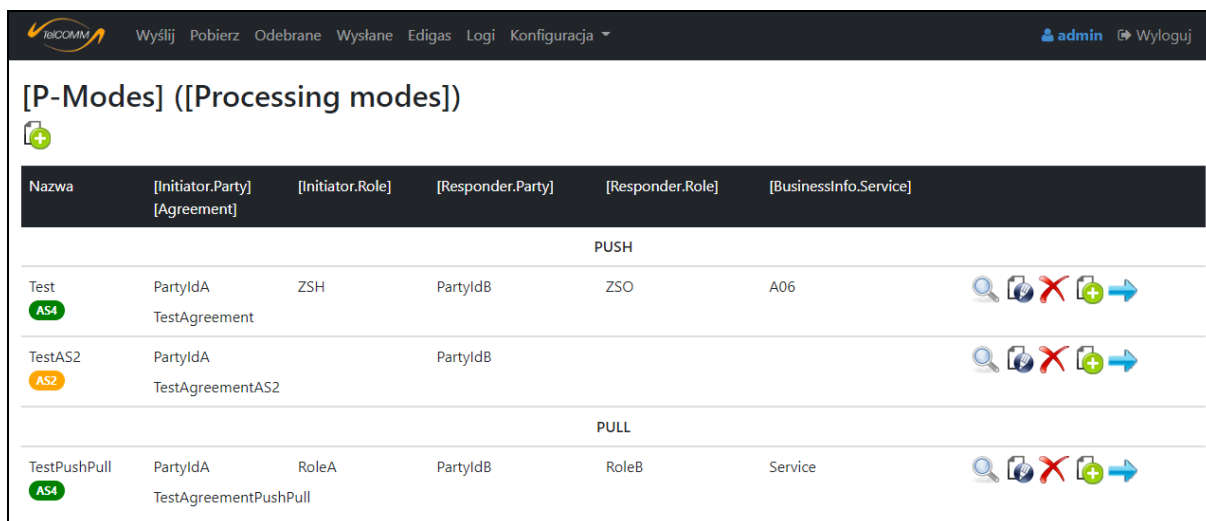


| [PartyId] | Nazwa |
|-----------|----------|
| PartyIdA | PartnerA |
| PartyIdB | PartnerB |

Rysunek 15. Ekran „Partnerzy”

EKRAN „[P-MODES]”

Ekran „Konfiguracja->[P-Modes]” ([Processing modes]) służy definiowaniu wszystkich informacji związanych z połączeniem między partnerami komunikacji. Dane te są ściśle związane z protokołem AS4 i jest to kluczowy ekran konfiguracyjny w aplikacji. [P-Modes] podzielone są na dwie grupy: PUSH – dotyczy wysyłania, PULL – dotyczy pobierania. W aplikacji można również wykorzystać [P-Mode] do zdefiniowania połączenia AS2.



| Nazwa | [Initiator.Party] [Agreement] | [Initiator.Role] | [Responder.Party] | [Responder.Role] | [BusinessInfo.Service] | |
|---------------------|-----------------------------------|------------------|-------------------|------------------|------------------------|--|
| PUSH | | | | | | |
| Test AS4 | PartyIdA TestAgreement | ZSH | PartyIdB | ZSO | A06 | |
| TestAS2 AS2 | PartyIdA TestAgreementAS2 | | PartyIdB | | | |
| PULL | | | | | | |
| TestPushPull AS4 | PartyIdA TestAgreementPushPull | RoleA | PartyIdB | RoleB | Service | |

Rysunek 16. Ekran „[P-Modes]”

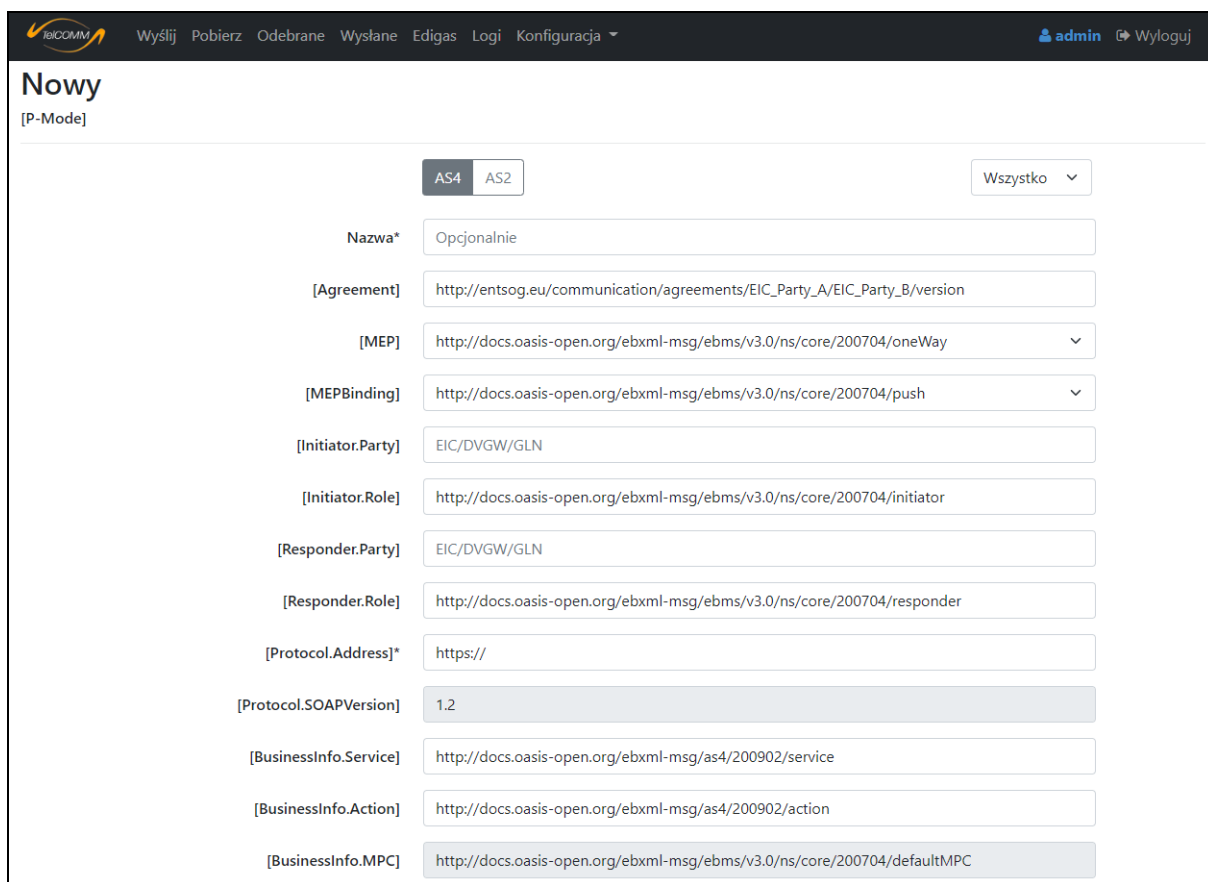
Jedną z informacji zawartych w [P-Mode] są certyfikaty partnerów komunikacji. Importowanie certyfikatu odbywa się na ekranie tworzenia i edycji [P-Mode]. Może się ono odbyć na dwa sposoby: poprzez podanie przyjaznej nazwy lub części pola podmiot certyfikatu zainstalowanego wcześniej w magazynie certyfikatów Windows albo ewentualnie poprzez wczytanie pliku certyfikatu wraz z podaniem hasła do pliku, skutkującego zapisem w bazie danych aplikacji. [P-Mode] odzwierciedla komunikację w jednym kierunku, dlatego jeśli komunikacja ma przebiegać w obydwu kierunkach, teoretycznie należałoby zdefiniować dwa [P-Mode], jednak przez to, że w praktyce używany jest ten sam certyfikat przez partnerów do podpisu i szyfrowania, wystarczy zdefiniować [P-Mode] w jednym kierunku. W przypadku definiowania [P-Mode] w obu kierunkach w [P-Mode] służącym do wysyłania certyfikat podpisu musi posiadać klucz prywatny, po to aby można było nim podpisać wysyłaną wiadomość, natomiast w [P-Mode] służącym do odbierania klucz prywatny musi być zawarty w certyfikacie szyfrowania, aby odszyfrować wiadomość odebraną. Stąd konieczność podania hasła w przypadku wczytywania, ponieważ plik certyfikatu posiadający klucz prywatny zostaje zabezpieczony hasłem. W przypadku definiowania jednego [P-Mode] dla obu kierunków lepiej zdefiniować [P-Mode] wysyłający, ewentualnie można odbierający.

Na ekranie tworzenia nowego [P-Mode] dla AS4 niektóre wartości są odgórnie ustalone w oparciu o dokumentację protokołu, dla wielu pozostałych pojawiają się wartości domyślne, które mogą zostać zmienione na inne. Ważnym polem jest [Agreement], którego wartość staje się identyfikatorem umowy w komunikacji między partnerami. Pola [MEP] i [MEPBinding] służą określeniu wzorca komunikacji, w polach [Initiator.Party] i [Responder.Party] należy podać identyfikator partnera – w komunikacji Edig@s jest nim kod EIC, a [Protocol.Address] odpowiada adresowi URL na jaki wysyłane są dokumenty.

W komunikacji AS2 pola [Initiator.Party] i [Responder.Party] zostają zamienione na [AS2-From] i [AS2-To], a pole [Agreement] używane jest jedynie wewnątrz aplikacji.

Jest również kilka pól dodatkowych, niezawartych w dokumentacji, jak np. Nazwa, Algorytm szyfrowania klucza (dla AS4) czy Certyfikat TLS serwera bądź klienta. Nazwa oprócz tego, że pozwala na łatwiejszą identyfikację danego [P-Mode] na ekranach aplikacji, jest używana podczas wysyłania wiadomości przez usługę internetową (web service). Algorytm szyfrowania klucza pozwala na wybranie standardowego szyfrowania z użyciem certyfikatu RSA lub szyfrowania z użyciem certyfikatu ECC. Wczytanie certyfikatu TLS serwera nie jest wymagane, ponieważ domyślnie podczas wysyłania certyfikat znajdujący się po stronie serwera jest sprawdzany czy jest prawidłowy (ma to miejsce gdy adres URL odbiorcy rozpoczyna się od „https://”, a nie „http://”). Można go wczytać gdy jedyną walidacją jaką chcemy wykonać to sprawdzenie czy certyfikat po stronie serwera jest identyczny co wczytany (mamy wtedy silniejszą weryfikację, ale nie ma

sprawdzenia czy certyfikat jest prawidłowy, czyli np. czy nie wygasł). Załączenie certyfikatu TLS klienta przyda się w niestandardowym przypadku gdy partner zgłosi taką potrzebę.

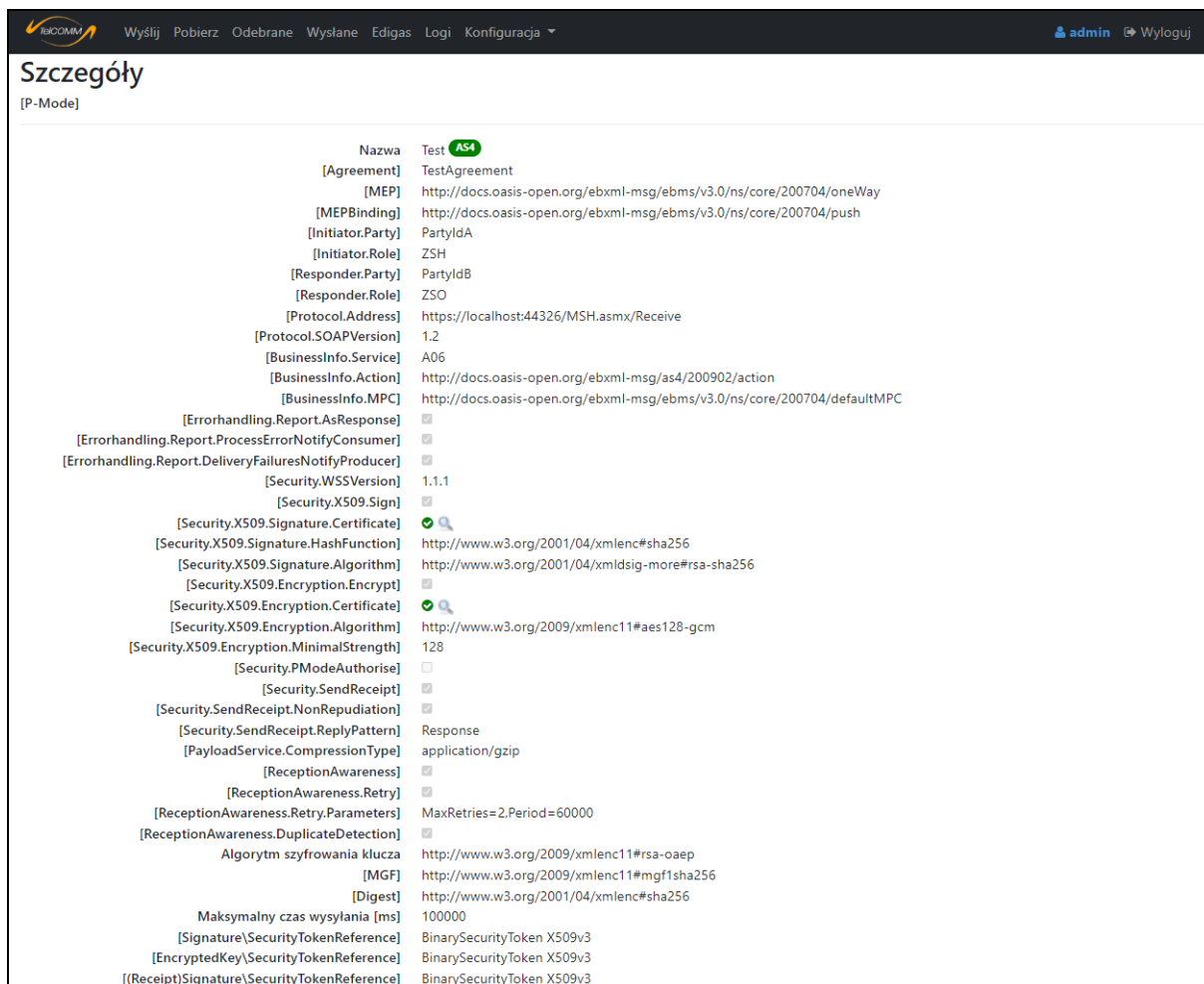


The screenshot shows a web interface for configuring a new profile. At the top, there is a navigation bar with the 'ibCOMM' logo and menu items: 'Wyślij', 'Pobierz', 'Odebrane', 'Wysłane', 'Edigas', 'Logi', and 'Konfiguracja'. On the right, it shows the user 'admin' and a 'Wyloguj' button. The main heading is 'Nowy' with a sub-heading '[P-Mode]'. Below this, there are two tabs: 'AS4' (selected) and 'AS2'. A dropdown menu is set to 'Wszystko'. The configuration fields are as follows:

| | |
|------------------------|---|
| Nazwa* | Opcjonalnie |
| [Agreement] | http://entsog.eu/communication/agreements/EIC_Party_A/EIC_Party_B/version |
| [MEP] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay |
| [MEPBinding] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push |
| [Initiator.Party] | EIC/DVGW/GLN |
| [Initiator.Role] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator |
| [Responder.Party] | EIC/DVGW/GLN |
| [Responder.Role] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder |
| [Protocol.Address]* | https:// |
| [Protocol.SOAPVersion] | 1.2 |
| [BusinessInfo.Service] | http://docs.oasis-open.org/ebxml-msg/as4/200902/service |
| [BusinessInfo.Action] | http://docs.oasis-open.org/ebxml-msg/as4/200902/action |
| [BusinessInfo.MPC] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC |

Rysunek 17. Część ekranu tworzenia nowego [P-Mode]

Na ekranie szczegółów istnieje możliwość podglądu właściwości [P-Mode], w tym także podglądu certyfikatów podpisu, szyfrowania i TLS.



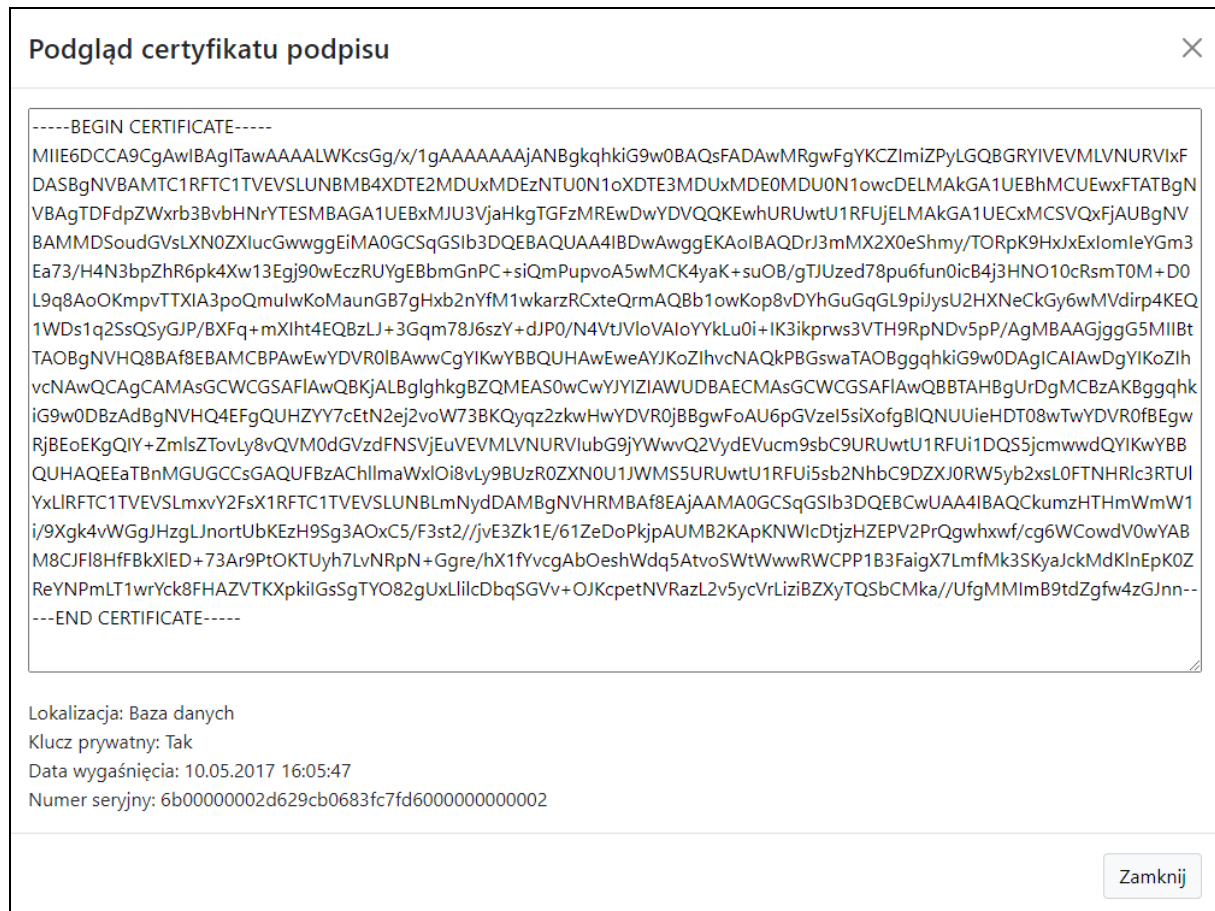
The screenshot shows a web interface with a top navigation bar containing 'Wyślij', 'Pobierz', 'Odebrane', 'Wysłane', 'Edigas', 'Logi', and 'Konfiguracja'. The main content area is titled 'Szczegóły' and '[P-Mode]'. It displays a list of configuration parameters for a test named 'AS4'. The parameters are listed in two columns, with their values and some status indicators (checkboxes, icons).

| Nazwa | Test AS4 |
|---|--|
| [Agreement] | TestAgreement |
| [MEP] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay |
| [MEPBinding] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push |
| [Initiator.Party] | PartyIdA |
| [Initiator.Role] | ZSH |
| [Responder.Party] | PartyIdB |
| [Responder.Role] | ZSO |
| [Protocol.Address] | https://localhost:44326/MSH.asmx/Receive |
| [Protocol.SOAPVersion] | 1.2 |
| [BusinessInfo.Service] | A06 |
| [BusinessInfo.Action] | http://docs.oasis-open.org/ebxml-msg/as4/200902/action |
| [BusinessInfo.MPC] | http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC |
| [Errorhandling.Report.AsResponse] | <input checked="" type="checkbox"/> |
| [Errorhandling.Report.ProcessErrorNotifyConsumer] | <input checked="" type="checkbox"/> |
| [Errorhandling.Report.DeliveryFailuresNotifyProducer] | <input checked="" type="checkbox"/> |
| [Security.WSSVersion] | 1.1.1 |
| [Security.X509.Sign] | <input checked="" type="checkbox"/> |
| [Security.X509.Signature.Certificate] | <input checked="" type="checkbox"/> |
| [Security.X509.Signature.HashFunction] | http://www.w3.org/2001/04/xmldsig-more#sha256 |
| [Security.X509.Signature.Algorithm] | http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 |
| [Security.X509.Encryption.Encrypt] | <input checked="" type="checkbox"/> |
| [Security.X509.Encryption.Certificate] | <input checked="" type="checkbox"/> |
| [Security.X509.Encryption.Algorithm] | http://www.w3.org/2009/xmldsig-more#aes128-gcm |
| [Security.X509.Encryption.MinimalStrength] | 128 |
| [Security.PModeAuthorise] | <input type="checkbox"/> |
| [Security.SendReceipt] | <input checked="" type="checkbox"/> |
| [Security.SendReceipt.NonRepudiation] | <input checked="" type="checkbox"/> |
| [Security.SendReceipt.ReplyPattern] | Response |
| [PayloadService.CompressionType] | application/gzip |
| [ReceptionAwareness] | <input checked="" type="checkbox"/> |
| [ReceptionAwareness.Retry] | <input checked="" type="checkbox"/> |
| [ReceptionAwareness.Retry.Parameters] | MaxRetries=2,Period=60000 |
| [ReceptionAwareness.DuplicateDetection] | <input checked="" type="checkbox"/> |
| Algorytm szyfrowania klucza | http://www.w3.org/2009/xmldsig-more#rsa-oaep |
| [MGF] | http://www.w3.org/2009/xmldsig-more#mgf1sha256 |
| [Digest] | http://www.w3.org/2001/04/xmldsig-more#sha256 |
| Maksymalny czas wysyłania [ms] | 100000 |
| [Signature\SecurityTokenReference] | BinarySecurityToken X509v3 |
| [EncryptedKey\SecurityTokenReference] | BinarySecurityToken X509v3 |
| [(Receipt)Signature\SecurityTokenReference] | BinarySecurityToken X509v3 |

Rysunek 18. Ekran szczegółów [P-Mode]

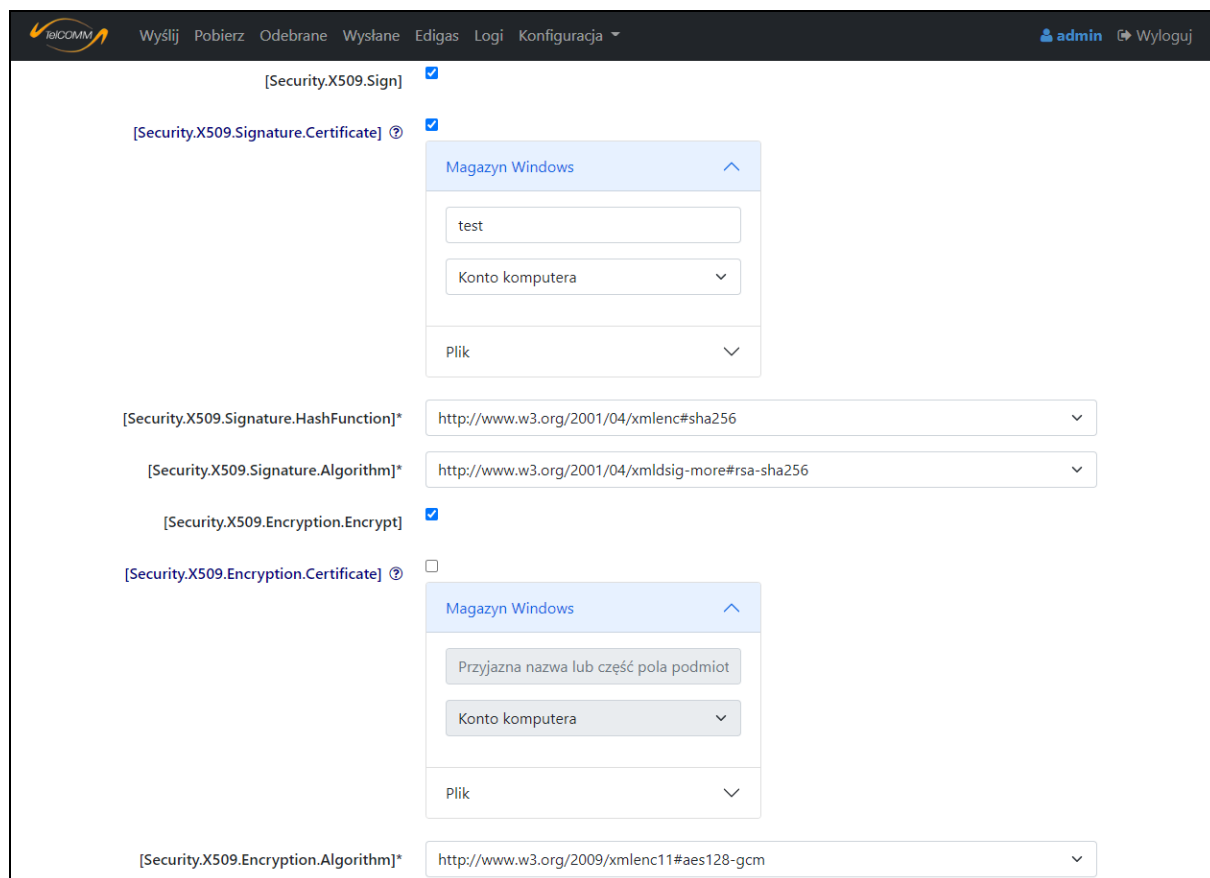
Zawartość okna „Podgląd certyfikatu” przedstawia certyfikat z kluczem publicznym w standardzie X.509 szyfrowany algorytmem Base-64 wraz z informacjami uzupełniającymi (szczegóły na poniższym rysunku). W przypadku włączonej w konfiguracji aplikacji opcji „Walidacja certyfikatów” pojawia się dodatkowa informacja, jeśli walidacja certyfikatu zakończyła się wynikiem negatywnym. Własne certyfikaty należy przekazać partnerowi komunikacji bez klucza prywatnego (!) w postaci tekstowej (zawartość okna) lub pliku (.cer, .pem), aby korzystając z zawartego w certyfikacie klucza publicznego mógł on:

- zweryfikować podpis wysłanej do niego wiadomości,
- zaszyfrować wysłaną przez niego do nas wiadomość.



Rysunek 19. Podgląd przykładowego certyfikatu podpisu na ekranie szczegółów [P-Mode]

W przypadku edycji [P-Mode] zmiana certyfikatu jest możliwa po zaznaczeniu pola wyboru „Wczytaj” przy polu certyfikatu. Dzięki temu podczas edycji można zachować poprzednie certyfikaty lub zmienić dowolny z nich.

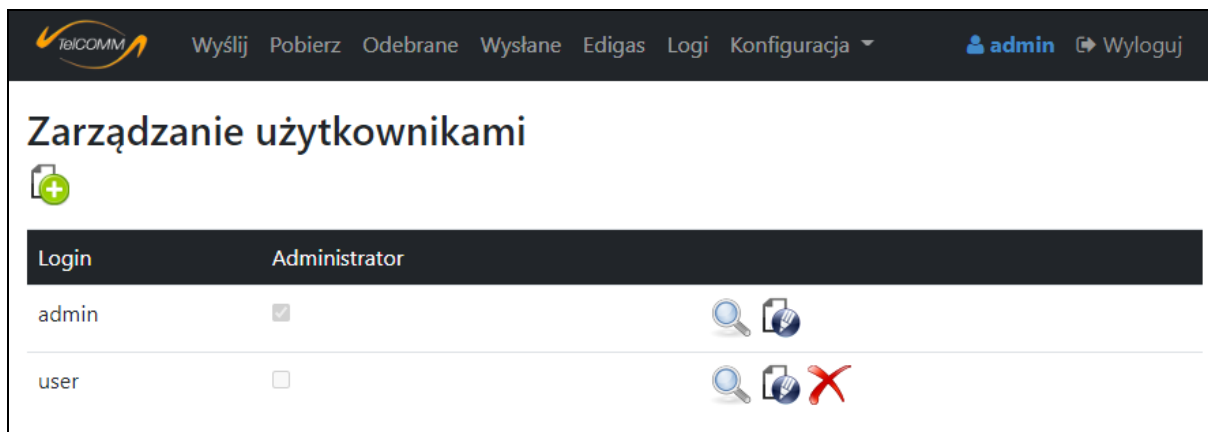


Rysunek 20. Część ekranu edycji [P-Mode]

Edycja certyfikatów w istniejącym [P-Mode] nie jest preferowanym rozwiązaniem – lepiej stworzyć nową wersję dla danego [P-Mode]. Na ogólnym ekranie jak i na ekranie szczegółów istnieje taka możliwość jako jedna z opcji dla [P-Mode] – pojawia się wtedy ekran tworzenia nowej wersji [P-Mode] z wypełnionymi polami aktualnymi wartościami. Korzystanie z takiego podejścia powoduje, że w komunikacji nadal będzie używany aktualny [P-Mode], do czasu gdy przestanie działać – wtedy nastąpi jego dezaktywowanie oraz przełączenie się na korzystanie z nowego [P-Mode] (więcej w rozdziale [Automatyczna aktualizacja certyfikatów](#)). Na obu wyżej wspomnianych ekranach istnieje również możliwość wyeksportowania [P-Mode] do XML. Usunąć [P-Mode] można definitywnie lub jedynie dezaktywować tj. [P-Mode] przestaje być używane w aplikacji, ale pozostaje w bazie danych.

EKRAN „UŻYTKOWNICY”

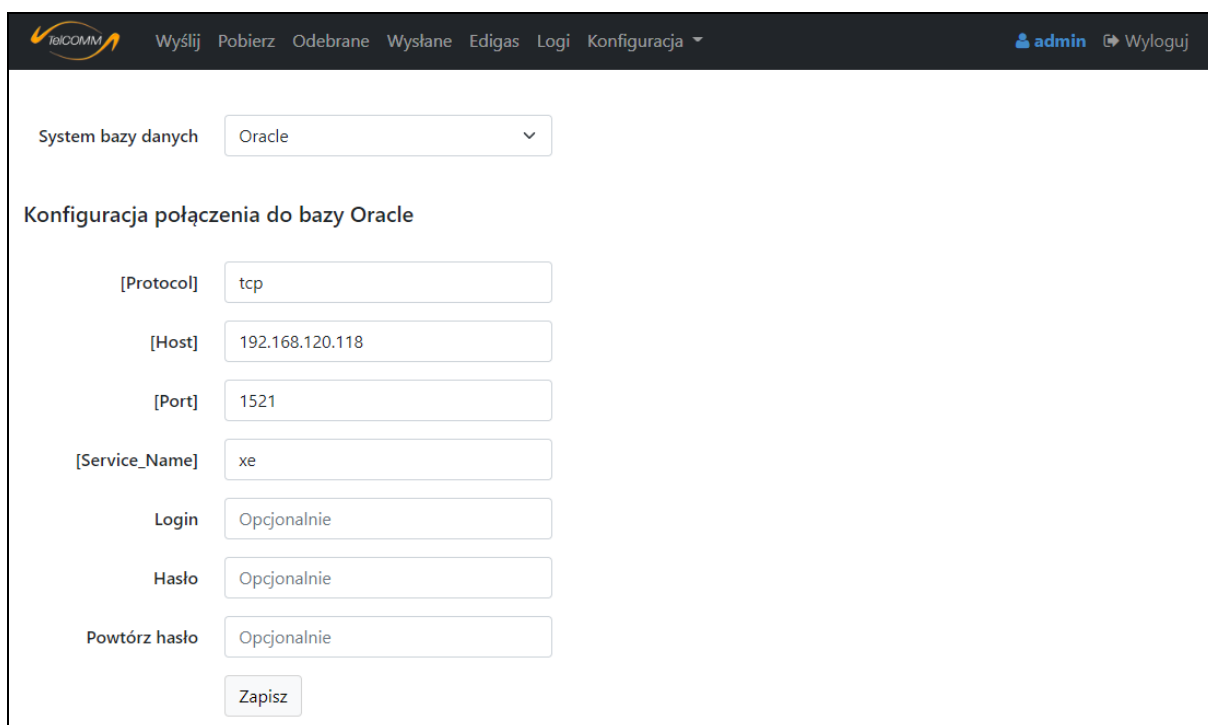
Ekran „Konfiguracja->Użytkownicy” służy do zarządzania użytkownikami. Użytkownik może otrzymać uprawnienia zwykłe lub administratora. W programie istnieje nieusuwalny użytkownik „admin”, którego początkowe hasło również „admin” należy zmienić na ekranie edycji użytkownika po pierwszym logowaniu (i login ew. też).



Rysunek 21. Ekran „Użytkownicy”

EKRAN „BAZA DANYCH”

Na ekranie „Konfiguracja-> Baza danych” istnieje możliwość wyboru systemu bazy danych SQLite lub Oracle i konfiguracji połączenia do bazy w przypadku wyboru systemu Oracle.



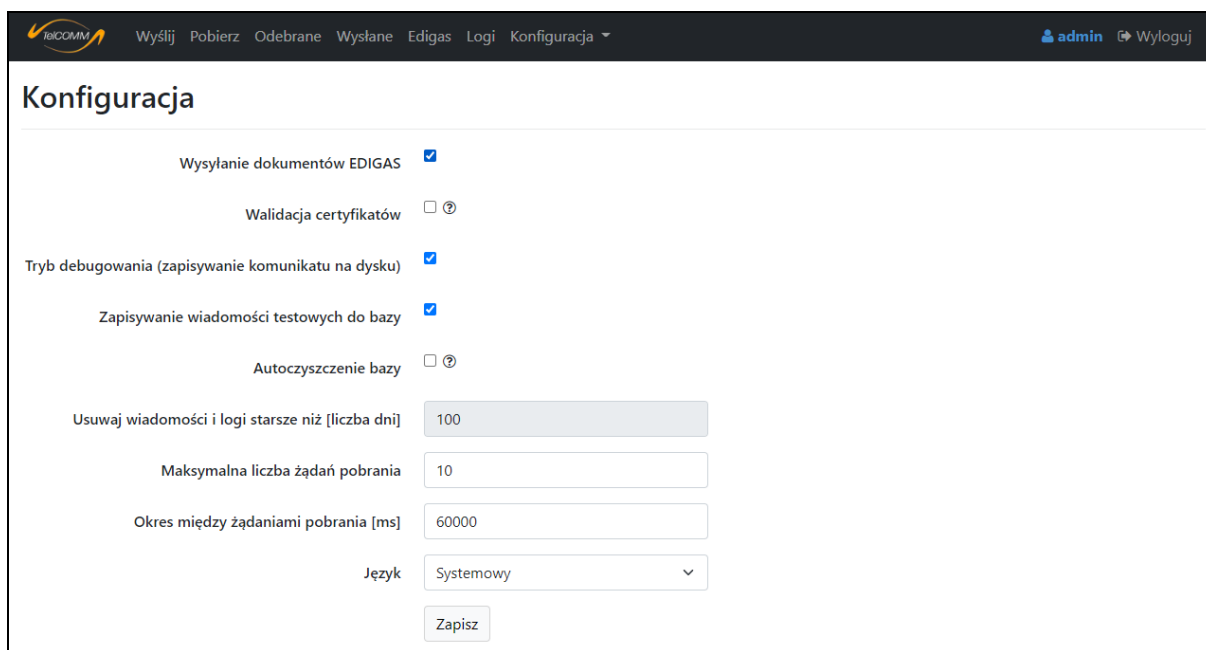
Rysunek 22. Ekran „Baza danych”

EKRAN „OPCJE”

Ekran „Konfiguracja->Opcje” posiada następujące opcje:

- „Wysyłanie dokumentów EDIGAS” – ustawienie określające czy w obrębie aplikacji mają pojawiać się pola odpowiednie dla wysyłania dokumentów Edig@s. W przypadku wybrania tej opcji, na pasku pojawia się zakładka „Edigas” a na ekranach wysyłania wiadomości oraz definiowania [P-Mode] pojawią się elementy związane komunikacją Edig@s.
- „Walidacja certyfikatów” – sprawdzanie poprawności używanych w obrębie aplikacji certyfikatów (również OCSP/CRL, dlatego wymagane jest połączenie z serwerem urzędu certyfikacji) i w przypadku nieprawidłowości blokowanie komunikatu,

- „Tryb debugowania (zapisywanie komunikatu na dysku)” – aplikacja posiada możliwość pracy w trybie debugowania, który sprowadza się do zapisywania całych żądań i odpowiedzi HTTP wysłanych i odebranych w folderze Debug (przechowywane są komunikaty z ostatnich 100 dni, do podglądu również na ekranie „Komunikaty”).
- „Zapisywanie wiadomości testowych do bazy” – opcja pozwala określić czy wiadomości testowe mają być zapisywane w bazie danych i tym samym widoczne na ekranach „Odebrane” i „Wysłane”.
- „Autoczyszczenie bazy” – opcja ta odpowiada za automatyczne usuwanie z bazy danych starszych wiadomości i wpisów w logu w oparciu o wartość z pola „Usuwanie wiadomości i logi starsze niż [liczba dni]”. Operacja jest wykonywana raz na dobę (dlatego odtwarzanie puli powinno być ustawione nie częściej niż co 24h).
- „Maksymalna liczba żądań pobrania” – parametr ten i następny dotyczą pobierania plików Pull; liczba wskazuje programowi ile razy po wysłaniu żądania ma próbować pobrać plik od partnera.
- „Okres między żądaniem pobrania [ms]” – określa ile czasu ma upłynąć do następnej próby pobrania pliku.
- „Język” – możliwość ustawienia języka polskiego lub angielskiego w obrębie aplikacji.



Wysyłanie dokumentów EDIGAS

Walidacja certyfikatów

Tryb debugowania (zapisywanie komunikatu na dysku)

Zapisywanie wiadomości testowych do bazy

Autoczyszczenie bazy

Usuwanie wiadomości i logi starsze niż [liczba dni]

Maksymalna liczba żądań pobrania

Okres między żądaniem pobrania [ms]

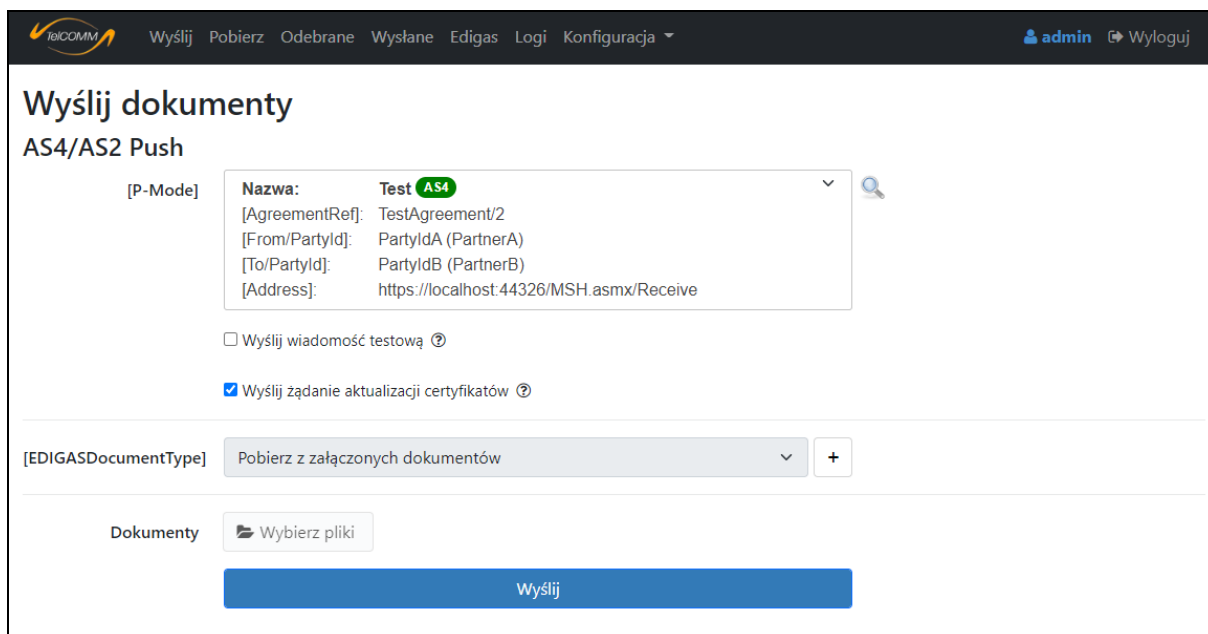
Język

Zapisz

Rysunek 23. Ekran „Opcje”

Automatyczna aktualizacja certyfikatów

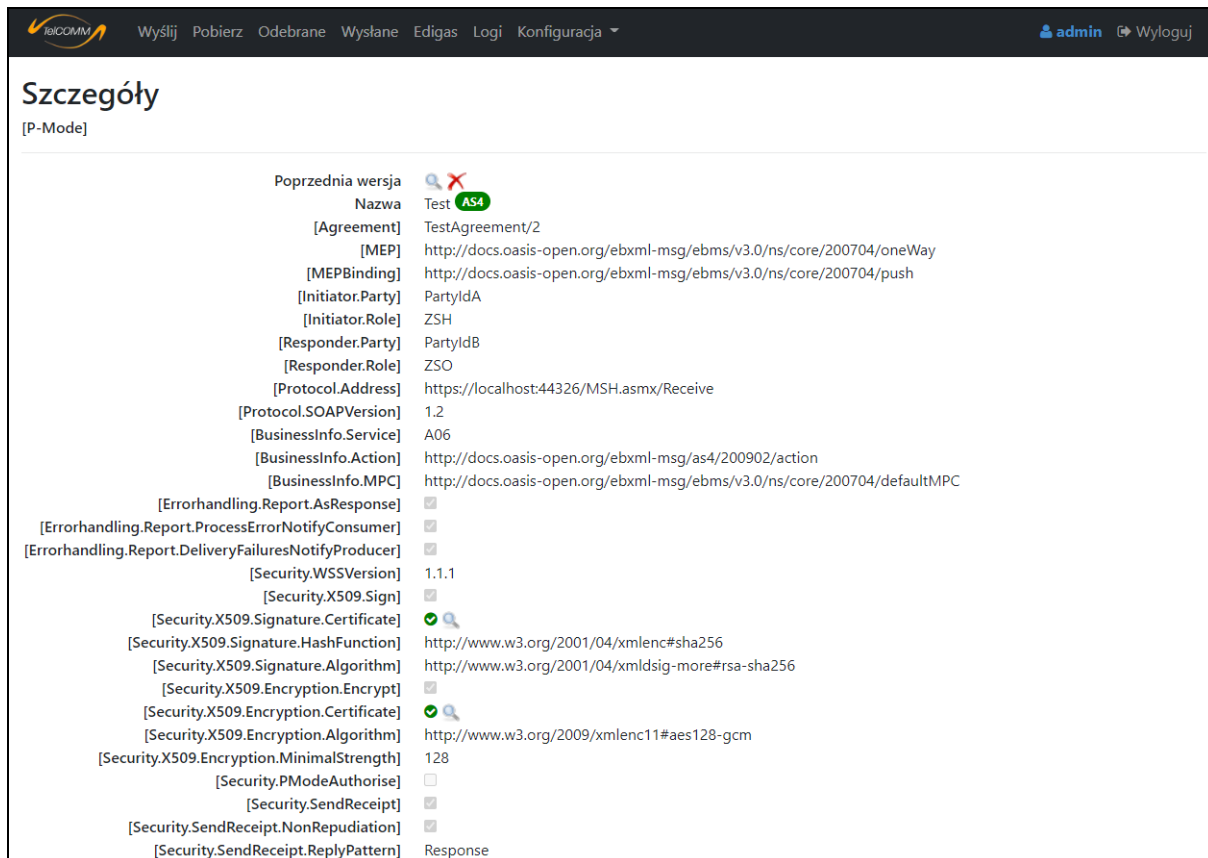
Funkcjonalność ta jest zaimplementowana dla protokołu AS4 i ma na celu zautomatyzowanie procesu aktualizacji certyfikatów podpisu i szyfrowania u partnerów komunikacji, który wynika z ograniczonej ważności certyfikatów. Aby można z niej skorzystać oprogramowanie po obu stronach komunikacji musi wspierać to rozwiązanie. Aplikacja TelCOMM przyjmuje żądania aktualizacji certyfikatów od partnerów (odbiorców), gdy chcą oni poinformować o aktualizacji swojego certyfikatu. Jeśli chodzi o ten kierunek aktualizacji nie są wymagane żadne działania ze strony użytkownika – wszystkie potrzebne operacje wykonywane są w sposób automatyczny, a ich wynik odnotowany w logu (podczas przełączenia na nowy certyfikat pojawi się wpis z błędem dotyczący odbioru, a następnie informacja o zakończeniu używania dotychczasowego [P-Mode]). Sytuacja ma się inaczej gdy aktualizacji wymaga certyfikat partnera (nadawcy) po stronie TelCOMM. Należy wtedy dla [P-Mode], w którym występuje dany certyfikat, dodać nie nowy [P-Mode], a nową wersję dla danego [P-Mode] z nowym certyfikatem. Jeśli [P-Mode] posiadających zmieniany certyfikat jest więcej, należy zaznaczyć przy dodawaniu nowego certyfikatu opcję „aktualizacja dla wszystkich [P-Mode]” – wtedy dla odpowiednich [P-Mode] zostanie również utworzona nowa wersja z nowym certyfikatem. Przy selekcji odpowiednich [P-Mode] brany jest pod uwagę jedynie certyfikat tego samego rodzaju np. podpisu, dlatego jeśli certyfikat, który ma zostać zmieniony jest używany także jako np. szyfrowania w [P-Mode] w kierunku odwrotnym, przy definiowaniu [P-Mode] w obu kierunkach, to również należy dla tego [P-Mode] dodać nową wersję i skorzystać z opcji „aktualizacja dla wszystkich [P-Mode]”. Teoretycznie w nowej wersji [P-Mode] powinna pojawić się nowa wartość dla [AgreementRef], ale jest dopuszczone zachowanie aktualnej wartości, chociaż wiąże się z tym ograniczenie. Na ekranie wysyłania (lub pobierania) dla każdej nowej wersji [P-Mode] (pod warunkiem, że w nowej wersji jest nowa wartość [AgreementRef]) pojawi się opcja „Wyślij żądanie aktualizacji certyfikatu”, którą należy zaznaczyć i wysłać komunikat.



Rysunek 24. Ekran „Wyślij” – wysyłanie żądania aktualizacji certyfikatu

Wysłanie żądania aktualizacji certyfikatu jest specjalnie spreparowanym komunikatem AS4 działającym na tych samych zasadach co pozostałe komunikaty AS4. Po wysłaniu reszta procesu polegająca na dezaktywowaniu aktualnej wersji [P-Mode] i aktywowaniu nowej odbywa się automatycznie w momencie gdy partner zaktualizuje konfigurację po swojej stronie, a szczegóły tej operacji zostają zapisane w logu (podczas przełączenia na nowy certyfikat pojawi się wpis z błędem dotyczący wysłania, a następnie informacja o zakończeniu używania dotychczasowego [P-Mode]).


W przypadku gdy chcemy już w danym momencie rozpocząć korzystanie z nowej wersji [P-Mode] należy na górze ekranu szczegółów dla nowej wersji [P-Mode] dezaktywować poprzednią wersję. Wtedy dla danego [P-Mode] będzie używany od tego momentu nowy certyfikat.



Wyślij Pobierz Odebrane Wysłane Edigas Logi Konfiguracja admin Wyloguj

Szczegóły

[P-Mode]

Poprzednia wersja  Test **AS4**

[Agreement] TestAgreement/2

[MEP] http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay

[MEPBinding] http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push

[Initiator.Party] PartyIdA

[Initiator.Role] ZSH

[Responder.Party] PartyIdB

[Responder.Role] ZSO

[Protocol.Address] https://localhost:44326/MSH.asmx/Receive

[Protocol.SOAPVersion] 1.2

[BusinessInfo.Service] A06

[BusinessInfo.Action] http://docs.oasis-open.org/ebxml-msg/as4/200902/action

[BusinessInfo.MPC] http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC


[Errorhandling.Report.AsResponse]

[Errorhandling.Report.ProcessErrorNotifyConsumer]

[Errorhandling.Report.DeliveryFailuresNotifyProducer]

[Security.WSSVersion] 1.1.1


[Security.X509.Sign]

[Security.X509.Signature.Certificate] 

[Security.X509.Signature.HashFunction] http://www.w3.org/2001/04/xmlenc#sha256

[Security.X509.Signature.Algorithm] http://www.w3.org/2001/04/xmldsig-more#rsa-sha256

[Security.X509.Encryption.Encrypt]

[Security.X509.Encryption.Certificate] 

[Security.X509.Encryption.Algorithm] http://www.w3.org/2009/xmlenc11#aes128-gcm

[Security.X509.Encryption.MinimalStrength] 128

[Security.PModeAuthorise]

[Security.SendReceipt]

[Security.SendReceipt.NonRepudiation]

[Security.SendReceipt.ReplyPattern] Response

Rysunek 25. Ekran szczegółów [P-Mode] – dezaktywowanie poprzedniej wersji

Interfejs do współpracy z aplikacją zewnętrzną

Aplikacja TelCOMM może spełniać zadanie bramki do komunikacji B2B, przez którą inna aplikacja może automatycznie wysyłać i odbierać wiadomości kanałem AS4 lub AS2, ponieważ udostępnia odpowiednie metody w usłudze internetowej /WebServices/**Gateway.asmx**. Aby móc korzystać z tych metod należy najpierw zalogować się w aplikacji TelCOMM. W tym celu zostały udostępnione dwie metody przeznaczone dla aplikacji zewnętrznych:

- a) /Login/LogOnExternal – metoda typu POST służąca do logowania aplikacji zewnętrznej w aplikacji TelCOMM. Login i hasło należy przekazać w parametrach „login” i „password”. W przypadku poprawnego zalogowania w odpowiedzi HTTP umieszczone zostaną ciastka (cookie), które należy załączyć do ciastek żądania HTTP adresowanego do metody z web service Gateway.asmx,
- b) /Login/LogOffExternal – metoda typu GET wylogowująca aplikację zewnętrzną z aplikacji TelCOMM.

W web service **Gateway.asmx** udostępniono metody, z których najważniejsze to:

- a) SendByPModeName – metoda służąca do wysyłania, jako argumenty przyjmuje pliki (nazwa pliku i binarna treść) oraz nazwę zdefiniowanego [PMode]. Poza tym ewentualnie podawana jest informacja o typie wysyłanego dokumentu Edig@s i o wysłaniu plików w osobnych wiadomościach, a zwracany jest obiekt z wynikiem analizy odpowiedzi od odbiorcy,
- b) GetSendOptions – metoda pomocnicza zwracająca możliwe wartości dla niektórych parametrów, z których można skorzystać w powyższych metodach,
- c) GetUnreadMessagesIds – zwraca listę wartości tekstowych „MessageId” wiadomości odebranych, które nie są oznaczone jako przeczytane; GetUnreadPullMessagesIds i GetUnreadPushMessagesIds – jw. z tą różnicą, że wyniki dotyczą odpowiedniego wzorca komunikacji,
- d) GetMessage – pobiera szczegóły wiadomości odebranej podając jako argument „MessageId”; GetMessages – pobiera szczegóły wielu wiadomości odebranych podając jako argument listę wartości „MessageId”; GetMessageSent – pobiera szczegóły wiadomości wysłanej,
- e) MarkAsRead – oznacza wiadomości jako przeczytane, przyjmując jako argument listę wartości „MessageId”,
- f) AddPullRequestByPModeName – bezpośrednio pobranie korzystając z wzorca One-Way/Pull,
- g) SendPullResponse – metoda wykorzystywana przy udostępnianiu danych z użyciem wzorca Two-Way/Push-Pull.

Poniższa tabela przedstawia jakich metod usługi internetowej Gateway.asmx należy użyć aby wysłać do lub odebrać od partnera komunikat dla danego wzorca komunikacji AS4 przez aplikację zewnętrzną:

| Wzorzec komunikacji AS4 | Kierunek | Metody web service Gateway.asmx |
|-------------------------|-----------|--|
| One-Way/Push | Wysłanie | 1. SendByPModeName lub SendAS2ByPModeName |
| | Odebranie | 1. GetUnreadPushMessagesIds 2. GetMessage lub GetMessages |

| | | |
|--------------------------|-----------|---|
| | | 3. MarkAsRead |
| One-Way/Pull | Wysłanie | 1. AddPullRequestByPModeName |
| Two-Way/Push-Pull | Wysłanie | 1. SendByPModeName |
| | Odebranie | <ol style="list-style-type: none"> 1. GetUnreadPullMessagesIds 2. GetMessage lub GetMessages 3. MarkAsRead 4. SendPullResponse (w przypadku udostępniania danych) |

Tab. 1. Metody web service Gateway używane przy wysłaniu i odbieraniu dla danego wzorca komunikacji

Przykład w środowisku .NET zalogowania do aplikacji, przekazania ciastka i (synchronicznego) wykorzystania metody z web service **Gateway.asmx** przedstawiony został na poniższym rysunku. Kod prezentuje również wylogowanie z aplikacji. W podanym przykładzie skorzystano z wcześniej dodanego Web Reference o nazwie „localhost”. Natomiast na rysunku kolejnym przedstawiony został przykład wysłania komunikatu AS4 w oparciu o zdefiniowany [P-Mode] o nazwie „nomint”. Wysyłane pliki zostały przekazane w argumencie „files” funkcji „SendByPModeName”, który jest listą obiektów o dwóch właściwościach: dane binarne i nazwa pliku. Dodatkowo określony zostaje typ wysyłanego dokumentu Edig@s i każdy plik zostaje wysłany w osobnej wiadomości (opcja ta jest brana pod uwagę przy wysłaniu więcej niż jednego pliku).

```

public void Example()
{
    CookieCollection cookieCollection = Login("https://localhost:44326/Login/LogOnExternal", "login=admin&password=admin");

    localhost.Gateway gateway = new localhost.Gateway();
    gateway.CookieContainer = new CookieContainer();
    gateway.CookieContainer.Add(cookieCollection);
    string[] unreadMessagesIds = gateway.GetUnreadPushMessagesIds();

    Logoff("https://localhost:44326/Login/LogOffExternal", cookieCollection);
}

1 reference | MSiatkowski, 57 days ago | 1 author, 2 changes
public CookieCollection Login(string uri, string parameters)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
    request.ContentType = "application/x-www-form-urlencoded";
    request.Method = "POST";
    byte[] bytes = System.Text.Encoding.ASCII.GetBytes(parameters);
    request.ContentLength = bytes.Length;
    request.CookieContainer = new CookieContainer();
    using (System.IO.Stream requestStream = request.GetRequestStream())
    {
        requestStream.Write(bytes, 0, bytes.Length);
        using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
        {
            return response.Cookies;
        }
    }
}

1 reference | MSiatkowski, 57 days ago | 1 author, 2 changes
public void Logoff(string uri, CookieCollection cookieCollection)
{
    HttpRequest request = (HttpRequest)WebRequest.Create(uri);
    request.Method = "GET";
    request.CookieContainer = new CookieContainer();
    request.CookieContainer.Add(cookieCollection);
    using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())
    {
    }
}

```

Rysunek 26. Przykład wykorzystania metody web service Gateway

```

localhost.Gateway gateway = new localhost.Gateway();
gateway.CookieContainer = new CookieContainer();
gateway.CookieContainer.Add(cookieCollection);

localhost.AS4SendModelPModeName sendModelPModeName = new localhost.AS4SendModelPModeName()
{
    PModeName = "nomint",
    EDIGASDocumentTypeCode = "01G",
    AttachmentsApart = true
};

localhost.SendResult result = gateway.SendByPModeName(files, sendModelPModeName);

```

Rysunek 27. Przykład wysłania komunikatu AS4



Usługa umożliwiająca odbiór dokumentów od partnera

Integralną częścią aplikacji TelCOMM jest usługa do odbierania komunikatów AS4 i AS2. Służy do tego metoda **Receive** w web service **MSH.asmx**. Adres względny odbierania to: **/MSH.asmx/Receive**.

Po odebraniu wiadomości wysyłana jest odpowiedź (synchronicznie lub w osobnym komunikacie) spełniająca wymaganie niezaprzeczalności odbioru (pod warunkiem, że wiadomość była podpisana i aplikacja dysponuje certyfikatem podpisu dla nadawcy odpowiedzi) lub z potwierdzeniem odbioru. Moduł odbierania wyposażony jest również w funkcjonalność wykrywania duplikatów wiadomości (duplicate detection) oraz obsługę błędów (error handling).



Udostępnianie danych

W przypadku korzystania z aplikacji TelCOMM w celu udostępniania danych za pomocą wzorca komunikacji **Two-Way/Push-Pull** analiza żądań o dane i zwracanie wyników do aplikacji odbywa się poza aplikacją TelCOMM z wykorzystaniem odpowiednich metod web service Gateway.asmx podanych w tabeli 1.

Załączniki

TelCOMM – Wymagania i architektura

TelCOMM - Procedura instalacji

TelCOMM – Konfiguracja AS4 – Szablon 1

TelCOMM – Konfiguracja AS4 – Szablon 2

Materiały źródłowe

Aplikacja TelCOMM została wykonana w oparciu o poniższe dokumenty, w których znajdują się szczegółowe informacje na temat komunikacji z użyciem protokołów AS4 i AS2.

[\[ENTSOG\] AS4 Usage profile](#)

[\[OASIS\] AS4 Profile of ebMS 3.0 Version 1.0](#)

[\[OASIS\] ebCore Agreement Update Specification Version 1.0](#)

[\[GAZ-SYSTEM\] Instrukcja w zakresie wymiany danych protokołem AS4](#)

[\[GAZ-SYSTEM\] Techniczny opis rozwiązania dla wymiany komunikatów Edig@s z wykorzystaniem standardu AS4](#)

[\[GAZ-SYSTEM\] Techniczny opis rozwiązania dla udostępniania danych pomiarowych i zagregowanych z wykorzystaniem standardu AS4](#)

[\[IETF\] AS2 Specification](#)

[\[IETF\] AS2 Compression](#)