Ministerstwo Finansów

Specyfikacja interfejsów usług Jednolitego Pliku Kontrolnego

Centrum Informatyki Resortu Finansów

1 lipca 2026 r.

Wersja systemu 5.5.1

Wersja dokumentu 5.5.1.v1

Data	Wersja systemu (wersja dokumentu)	Opis
23.05.2016	1.3	Opublikowanie specyfikacji technicznej usług Jednolitego Pliku Kontrolnego.
10.06.2016	1.4	1. Zmiana metody dzielenia spakowanego pliku z metody TAR na binarne dzielenie pliku SPLIT. 2. Metoda Status: • zmiana zwracanej zawartości dla kodu http: 200 i 400. 3. Metoda InitUploadSigned w przypadku kodu http: 200 • zmiana typu dla właściwości TimeoutInSec z Timespan na int. 4. Zmiany schematu XSD pliku metadanych: • dodanie typu dokumentu JPKAH (JPK ad hoc) dla plików przysłanych w ramach kontroli, • poprawienie nazwy (literówka) EncrypionKey na EncryptionKey, • poprawienie formatu wersji REST API, • poprawienie formatu nazwy pliku, • dodanie całkowitej liczby części podzielonego pliku oraz liczby porządkowej dla poszczególnych części, • usunięcie atrybutów type oraz mode z listy plików cząstkowych FileSignatureList, • dodanie elementu (Packaging) w liście plików cząstkowych FileSignatureList wraz z możliwością wyboru rodzaju podziału i kompresji pliku. Obecnie możliwe jest użycie kompresji zip (deflate) z podziałem binarnym – element SplitZip z atrybutami type (split) oraz mode (zip), • dodanie elementu Encryption w liście plików cząstkowych FileSignatureList wraz z możliwością wyboru algorytmu szyfrowania. Obecnie wykorzystanie algorytmu AES256 – element AES z atrybutami size (256), block (16), mode (CBC), padding (PKCS#7) oraz elementem IV (Initialization Vector) z atrybutami bytes (16) i encoding (Base64).
17.06.2016	1.5	Zmiany schematu XSD pliku metadanych:

Data	Wersja systemu (wersja dokumentu)	Opis
		• ustalenie obsługiwanej wersji REST API – 01.02.01.20160617, • zmiana wyrażenia regularnego elementu FileName, • uzupełnienie zbioru kodów odpowiedzi dla metody Status.
04.07.2016	1.6	1. Dodanie opisu specyfikacji szyfrowania klucza szyfrującego. 2. Zmiana w opisie interfejsów – przetłumaczenie komunikatów na język polski. 3. Dodanie identyfikatora żądania (RequestId) w strukturze odpowiedzi dla kodu http: 400 i 500. 4. Rozszerzenie zbioru kodów odpowiedzi błędów (400 Bad Request) metody InitUploadSigned. 5. Dodanie informacji o dopuszczalnych transformacjach dla podpisu metadanych. 6. Ograniczenie długości wartości funkcji skrótów w schemacie XSD pliku metadanych.
20.07.2016	1.7	1. Rozszerzenie zbioru kodów odpowiedzi błędów (400 Bad Request) metody InitUploadSigned. 2. Dodanie przykładów prawidłowych odpowiedzi inicjowania sesji metodą InitUploadSigned. 3. Zamieszczenie przykładów wykorzystania narzędzi programistycznych SDK metody Put Blob. 4. Dodanie informacji o parametrze umożliwiającym włączenie weryfikacji podpisu z certyfikatem kwalifikowanym przy inicjowaniu sesji metodą InitUploadSigned na środowisku testowym.
29.07.2016	2.0	Doprecyzowanie mechanizmu kompresji ZIP.
30.09.2016	2.1	1. Uzupełnienie zbioru kodów odpowiedzi błędów (400 Bad Request) metody InitUploadSigned. 2. Wyszczególnienie adresów domenowych używanych przestrzeni Azure Storage.

Data	Wersja systemu (wersja dokumentu)	Opis
31.01.2017	2.2	Zmiana przykładów prawidłowych odpowiedzi inicjowania sesji metodą InitUploadSigned.
31.03.2017	2.3	1. Rozszerzenie opisu funkcjonalności podpisu metadanych o obsługę europejskiego podpisu kwalifikowanego oraz podpisu Profilem Zaufanym. 2. Rozszerzenie opisu adresów domenowych używanych przestrzeni Azure Storage.
11.05.2020	3.0	1. Rozszerzenie opisu przygotowania metadanych uwierzytelniających o możliwość skorzystania z danych autoryzujących (autoryzacja danymi osobowymi oraz wartościami kwot z poprzednich rozliczeń). 2. Aktualizacja kodów statusów: • dodanie nowego kodu 136 zwracanego w metodzie InitUploadSigned, • usunięcie niewystępujących kodów (102, 110, 301, 302, 303, 403, 404, 409, 411, 414), • dodanie nowych kodów 417, 418, 419, 420, 422, 423, 424 zwracanych w metodzie Status. 3. Dodanie opisu pełnomocnictw. 4. Rozszerzenie zakresu adresów magazynów chmurowych (p. 2.2).
25.09.2020	3.1	Aktualizacja kodów statusów: • dodanie nowego kodu błędu 155 zwracanego w metodzie InitUploadSigned.
06.11.2020	3.2	Aktualizacja kodów statusów: • dodanie nowego kodu błędu 411 zwracanego w metodzie Status.
21.01.2021	3.3	Dodanie obsługi plików CUK(1).
27.05.2021	3.4	1. Dodanie obsługi plików CUK(2) i ALK(1). 2. Aktualizacja kodów statusów.

Data	Wersja systemu (wersja dokumentu)	Opis
		• dodanie nowych kodów błędów 99 i 101 zwracanych w metodzie InitUploadSigned, • dodanie nowych kodów 425 i 426 zwracanych w metodzie Status.
13.01.2022 07.12.2022 16.01.2023	3.5	1. Dodanie obsługi plików JPK_V7M(2) i JPK_V7K(2). 2. Dodanie obsługi plików ITP (1) i ITP-Z (1). 3. Dodanie obsługi plików ITP (2) i ITP-Z (2).
21.06.2023	3.6	Dodanie obsługi plików JPK_GV(1).
13.10.2023	4.0	1. Dodanie obsługi CESOP: PSP-IP (4). 2. „Schemat blokowy kroków przygotowywania do wysyłki danych" – dodanie opcjonalnego kroku z podpisem dokumentu.
23.11.2023	4.1	1. Dodanie obsługi CESOP: PSP-FR (1). 2. Zmiany redakcyjne w dokumencie.
05.12.2023	4.2	1. Dodanie obsługi plików ALK(2). 2. Dodanie informacji o datach wdrożenia produkcyjnego schematów ALK(2), PSP-FR(1), PSP-IP(4).
6.03.2024	4.3	1. Dodanie obsługi plików DPI
26.04.2024	4.3.1 (4.3.1.v1)	1. Uzupełnienie listy statusów dla kodu 400, w rozdziale 2.2.1, o brakujące pozycje. 2. Zmiany redakcyjne w dokumencie.
14.11.2024	4.4.0 (4.4.0.v1)	1. Zmiany redakcyjne w dokumencie. 2. Aktualizacja kodów statusów (429). 3. Dodanie obsługi plików ITP (2) w wersji 2-3 i ITP-Z (2) w wersji 2-2 (schemy zaczną obowiązywać od 01.04.2025). 4. Dodanie obsługi plików JPK_KR_PD (1) i JPK_ST_KR (1) (schemy zaczną obowiązywać od 01.01.2025).

Data	Wersja systemu (wersja dokumentu)	Opis
15.01.2025	4.4.1 (4.4.1.v1)	1. Poprawki błędów i optymalizacje
01.07.2025	5.0.0 (5.0.0.v1)	1. Optymalizacje 2. Aktualizacja kodów statusów
20.08.2025	5.1.0 (5.1.0.v1)	Dodanie obsługi plików JPK_EWP (4), JPK_PKPIR (3), JPK_ST (1) (schemy zaczną obowiązywać produkcyjnie od 01.01.2026). Możliwa jest już wysyłka na środowisko testowe.
02.10.2025	5.1.0 (5.1.0.v2)	Dodano maksymalne rozmiary plików oraz doprecyzowano sposób składania podpisów elektronicznych.
31.10.2025	5.1.0 (5.1.0.v3)	Zaktualizowano informacje o akceptowanych typach podpisów.
20.11.2025	5.1.0 (5.1.0.v4)	Zmiany redakcyjne w dokumencie
20.01.2026	5.2.0 (5.2.0.v1)	1. Dodanie obsługi plików JPK_V7M (3), JPK_V7K (3) (schemy zaczną obowiązywać produkcyjnie od 01.02.2026). Możliwa jest już wysyłka na środowisko testowe. 2. Zmiany redakcyjne w dokumencie
05.03.2026	5.3.0 (5.3.0.v1)	Optymalizacje i poprawki błędów
01.04.2026	5.4.0 (5.4.0.v1)	1. Dodanie obsługi plików GIR-1 (1) (schema zacznie obowiązywać produkcyjnie od 01.04.2026).
16.06.2026	5.5.0 (5.5.0.v1)	1. Aktualizacja kodów błędów zawartych w odpowiedzi (400 – Bad Request) 2. Opis konfiguracji zabezpieczeń protokołu TLS
01.07.2026	5.5.1 (5.5.1.v1)	1. Aktualizacja kodów błędów zawartych w odpowiedzi (400 – Bad Request) – usunięcie kodu 99 2. Dodanie możliwości podpisu kwotą przychodu (dane autoryzujące) dla wybranych schem. Szczegóły zostały opisane w niniejszej specyfikacji.

Ministerstwo Finansów

Spis treści

Ministerstwo Finansów

1. Przygotowanie danych JPK

1.1. Format pliku i typ dokumentu

Formatem plików jest zawsze .xml. W przypadku, kiedy mowa jest o **dokumencie XML**, rozumiane jest to jako rodzaj składanego dokumentu, czyli wartość pola „DocumentType". Należy zwrócić uwagę, że plik XML nie musi być dokumentem XML.

Typy dokumentów opisane są w rozdziale 1.4.

1.2. Przygotowanie dokumentów JPK

Dane JPK przygotowywane będą po stronie klienta (np. w systemie ERP) w formie plików XML zgodnych ze schematem XSD opublikowanym przez:

- Ministerstwo Finansów na stronie https://epuap.gov.pl/wps/portal/strefa-urzednika/inne-systemy/crwde lub na stronie Struktury JPK - Ministerstwo Finansów - Krajowa Administracja Skarbowa - Portal Gov.pl (www.gov.pl).
- Komisję Europejską na stronach: https://taxation-customs.ec.europa.eu/taxation-1/central-electronic-system-payment-information-cesop_en.

Nazwy schematów opublikowanych w CRWDE ePUAP:

- **JPK_V7M(1), JPK_V7M(2), JPK_V7M(3)** DEKLARACJA MIESIĘCZNA I EWIDENCJA DLA PODATKU OD TOWARÓW I USŁUG (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_V7K(1), JPK_V7K(2), JPK_V7K(3)** DEKLARACJA KWARTALNA I EWIDENCJA DLA PODATKU OD TOWARÓW I USŁUG (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **CUK (1), CUK (2)** INFORMACJA W SPRAWIE OPŁATY OD ŚRODKÓW SPOŻYWCZYCH. Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **ALK (1), ALK (2)** INFORMACJA W SPRAWIE OPŁATY ZA ZEZWOLENIE NA OBRÓT HURTOWY NAPOJAMI ALKOHOLOWYMI W OPAKOWANIACH DO 300 ML (schemat **ALK(2)** dostępny produkcyjnie od 01.01.2024). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_GV (1)** EWIDENCJA WEWNĘTRZNA CZŁONKÓW GRUPY VAT. Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.

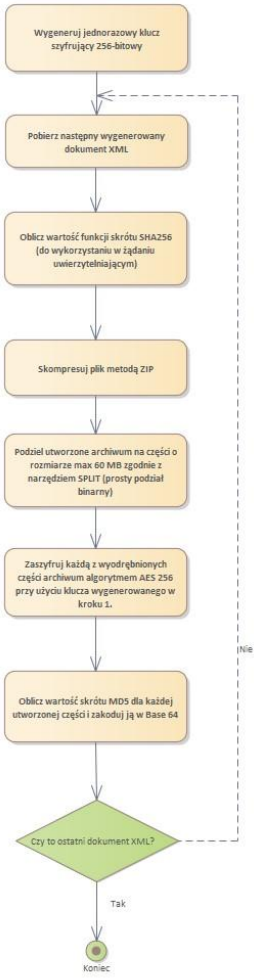
Oprócz ww. schematów opublikowanych w CRWDE e-PUAP obsługiwane są również schematy opublikowane na stronie BIP MF/KAS:

- **JPK_FA(4)** FAKTURA VAT (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.

- **JPK_FA_RR(1)** FAKTURA VAT ROLNICY RYCZAŁTOWI (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_EWP(3)** EWIDENCJA PRZYCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (3)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_EWP(2)** EWIDENCJA PRZYCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (2)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_EWP(1)** EWIDENCJA PRZYCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_PKPIR(2)** PODATKOWA KSIĘGA PRZYCHODÓW I ROZCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (2)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_KR(1)** KSIĘGI RACHUNKOWE (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_MAG(1)** MAGAZYN (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_WB(1)** WYCIĄG BANKOWY (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **ITP (1), ITP (2), ITP-Z (1), ITP-Z (2)** INFORMACJA O TRANSAKCJACH PŁATNICZYCH PRZY UŻYCIU TERMINALI PŁATNICZYCH. Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **PSP-FR(1)** FORMULARZ REJESTRACYJNY DLA INSTYTUCJI PŁATNICZYCH ZOBOWIĄZANYCH DO RAPORTOWANIA W RAMACH CESOP(schemat **PSP-FR(1)** dostępny produkcyjnie od 29.03.2024). Maksymalny całkowity rozmiar dokumentu wynosi 1 GB.
- **PSP-IP(4)** RAPORT OD INSTYTUCJI PŁATNICZYCH ZOBOWIĄZANYCH DO RAPORTOWANIA W RAMACH CESOP (schemat **PSP-IP(4)** dostępny produkcyjnie od 29.03.2024). Maksymalny całkowity rozmiar dokumentu wynosi 1 GB.
- **DPI-FR(1)** FORMULARZ REJESTRACYJNY DLA OPERATORÓW PLATFORM ZOBOWIĄZANYCH DO RAPORTOWANIA (schemat **DPI-FR(1)** dostępny produkcyjnie od 01.07.2024). Maksymalny całkowity rozmiar dokumentu wynosi 1 GB.
- **DPI-IS(1)** INFORMACJA O SPRZEDAWCACH OD OPERATORA PLATFORMY ZOBOWIĄZANEGO DO RAPORTOWANIA (schemat **DPI-IS(1)** dostępny produkcyjnie od 01.07.2024). Maksymalny całkowity rozmiar dokumentu wynosi 1 GB.
- **JPK_ST_KR(1)** EWIDENCJA ŚRODKÓW TRWAŁYCH I WARTOŚCI NIEMATERIALNYCH I PRAWNYCH DLA PODATNIKÓW SKŁADAJĄCYCH JPK_KR_PD (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)) (schemat dostępny produkcyjnie od 01.01.2025). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.

- **JPK_KR_PD(1)** KSIĘGI RACHUNKOWE (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)) (schemat dostępny produkcyjnie od 01.01.2025). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_PKPIR(3)** PODATKOWA KSIĘGA PRZYCHODÓW I ROZCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (3)) (schemat dostępny produkcyjnie od 01.01.2026). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_EWP(4)** EWIDENCJA PRZYCHODÓW (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (4)) (schemat dostępny produkcyjnie od 01.01.2026). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **JPK_ST(1)** EWIDENCJA / WYKAZ ŚRODKÓW TRWAŁYCH ORAZ WARTOŚCI NIEMATERIALNYCH I PRAWNYCH (W FORMIE JEDNOLITEGO PLIKU KONTROLNEGO (1)) (schemat dostępny produkcyjnie od 01.01.2026). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.
- **GIR-1(1)** INFORMACJA O OPODATKOWANIU WYRÓWNAWCZYM (GLOBE INFORMATION RETURN). Maksymalny całkowity rozmiar dokumentu wynosi 200 GB.

Każdy z dokumentów opisanych właściwym schematem ma stanowić osobny plik XML. Wygenerowany plik XML powinien być zakodowany w UTF-8. Przygotowanie dokumentów JPK do wysłania odbywa się zgodnie ze schematem zamieszczonym poniżej:



Rysunek: Schemat blokowy kroków przygotowywania do wysyłki danych.

1.2.1. Kompresja danych

Wygenerowany dokument zostanie skompresowany do pliku w formacie ZIP oraz podzielony binarnie na części o wielkości nie przekraczającej 60 MB.

Wymagana metoda kompresji to format pliku ZIP z użyciem algorytmu DEFLATE, bez stosowania opcji dzielenia (split/multipart). W wyniku kompresji powinien powstać jeden plik ZIP zawierający pojedynczy dokument. Jeżeli rozmiar otrzymanego pliku ZIP przekracza 60MB, należy go podzielić binarnie na odpowiednią liczbę części o wielkości 60MB każda oraz ostatnią część o rozmiarze nie większym niż 60MB.

Wykorzystanie tego podejścia umożliwia zastosowanie powszechnie dostępnych narzędzi i zapewnia łatwość implementacji na różnych platformach.

1.2.2. Szyfrowanie danych

Po skompresowaniu plików, kolejnym etapem jest ich szyfrowanie. Używany jest do tego algorytm AES256, z kluczem generowanym po stronie klienta.

Specyfikacja algorytmu AES:

Długość klucza	Key Size	256 bits / 32 bytes
Tryb szyfru	Cipher Mode	CBC (Cipher Block Chaining)
Dopełnienie	Padding	PKCS#7
Rozmiar bloku	Block Size	16 bytes
Wektor inicjujący	Initialization Vector	16 bytes

Procedura szyfrowania:

- **Generowanie klucza**: Klient tworzy losowy klucz o długości 256 bitów.
- **Szyfrowanie archiwum**: Wszystkie segmenty skompresowanego archiwum są szyfrowane przy użyciu wyżej wymienionego algorytmu AES256 i wygenerowanego klucza.
- **Szyfrowanie klucza**: Klucz używany do szyfrowania plików jest następnie szyfrowany przy pomocy algorytmu asymetrycznego RSA. Do tego celu używany jest certyfikat klucza publicznego udostępnionego przez Ministerstwo Finansów.
- **Dołączanie klucza do metadanych**: Po zaszyfrowaniu klucz jest dołączany do pliku z metadanymi, który jest opisany w dalszej części dokumentacji.

1.2.3. Szyfrowanie klucza szyfrującego

Szyfrowanie klucza szyfrującego należy wykonać algorytmem asymetrycznym RSA z wykorzystaniem certyfikatu klucza publicznego, udostępnionego przez Ministerstwo Finansów.

Specyfikacja algorytmu RSA:

Długość klucza	Key Size	2048 bits
Tryb szyfru	Cipher Mode	ECB (Electronic Codebook)
Dopełnienie	Padding	PKCS#1

1.3. Przygotowanie metadanych uwierzytelniających

Po przygotowaniu zasadniczych dokumentów zgodnych ze schematem odpowiedniego rodzaju pliku, klient, w celu wysłania danych, musi przygotować dane uwierzytelniające, mające postać odpowiedniego pliku w formacie XML, przesłane w metodzie InitUploadSigned (opisanej w dalszej części dokumentacji).

Plik metadanych musi być uwierzytelniony jedną z technik:

1. użycie:
    a. podpisu kwalifikowanego (polski lub europejski),
    b. podpisu zaufanego.
2. umieszczenie elementu AuthData zawierającego zaszyfrowane dane autoryzujące.

Dostępne metody uwierzytelnienia

Schemat	Metody uwierzytelnienia
JPK_V7M(1), JPK_V7M(2), JPK_V7M(3)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_V7K(1), JPK_V7K(2), JPK_V7K(3)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
CUK (1), CUK (2)	podpis kwalifikowany
ALK (1), ALK (2)	podpis kwalifikowany
JPK_GV (1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_FA(4)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_FA_RR(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_EWP(3)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_EWP(2)	podpis kwalifikowany, podpis zaufany
JPK_EWP(1)	podpis kwalifikowany, podpis zaufany
JPK_PKPIR(2)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_KR(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_MAG(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_WB(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
ITP (1), ITP (2), ITP-Z (1), ITP-Z (2)	podpis kwalifikowany, podpis zaufany

Schemat	Metody uwierzytelnienia
PSP-FR(1)	podpis kwalifikowany, podpis zaufany
PSP-IP(4)	podpis kwalifikowany, podpis zaufany
DPI-FR(1)	podpis kwalifikowany, podpis zaufany
DPI-IS(1)	podpis kwalifikowany, podpis zaufany
JPK_ST_KR(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_KR_PD(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_PKPIR(3)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_EWP(4)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
JPK_ST(1)	podpis kwalifikowany, podpis zaufany, dane autoryzujące
GIR-1(1)	podpis kwalifikowany, podpis zaufany

1.3.1. Podpis kwalifikowany lub podpis zaufany

Plik metadanych musi być podpisany cyfrowo **podpisem kwalifikowanym polskim lub europejskim** albo **podpisem zaufanym** zgodnie z algorytmem XAdES Basic Electronic Signature w postaci pliku XML zgodnego ze schematem http://www.w3.org/2000/09/xmldsig, w skrócie XAdES-BES (w tym BES-T ze znacznikiem czasu) w wersji **Enveloped** (podpis jako dodatkowy element ds:Signature w oryginalnym XML) lub **Enveloping** (oryginalny dokument zawarty jako element w podpisanej strukturze). Przy podpisywaniu można dokonać transformacji obiektu podpisywanego zgodnie z kodowaniem http://www.w3.org/2000/09/xmldsig#base64.

Funkcją skrótu wykorzystywaną w podpisie powinna być RSA-SHA256.

Zgodnie ze specyfikacją, prawidłowy podpis XAdES-BES powinien zawierać dwie referencje w elemencie ds:SignedInfo:

- referencję do elementu SignedProperties,
- referencję do całego dokumentu XML.

Brak którejkolwiek z tych referencji spowoduje odrzucenie pliku przez system.

Przykład metadanych uwierzytelniających można znaleźć w dalszej części dokumentu, w rozdziale gdzie opisana jest metoda InitUploadSigned, przyjmująca metadane uwierzytelniające.

1.3.2. Dane autoryzujące

W przypadku korzystania z metody autoryzacji kwotą należy uzupełnić element AuthData:

```
<xs:element name="AuthData" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>To opcjonalne pole powinno zawierać dokument XML zgodny z opublikowaną schemą SIG-2008_v2-0.xsd (https://www.podatki.gov.pl/e-deklaracje/dokumentacja-it/struktury-dokumentow-xml/) zaszyfrowany z wykorzystaniem algorytmu symetrycznego AES256. Powinien zostać wykorzystany ten sam klucz, który jest wykorzystywany do szyfrowania części skompresowanego archiwum pliku JPK i załączany do niniejszego pliku metadanych. Algorytm kodowania zaszyfrowanych danych to Base64.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:element>
```

Pole to powinno zawierać dokument XML zgodny z opublikowanym schematem SIG-2008_v2-0.xsd zaszyfrowany z wykorzystaniem algorytmu symetrycznego AES256 (generowany po stronie klienta). Powinien zostać wykorzystany **ten sam klucz**, który jest wykorzystywany do szyfrowania części skompresowanego archiwum pliku JPK i załączany do pliku metadanych. Algorytm kodowania zaszyfrowanych danych to Base64.

Parametry szyfrowania danych autoryzujących:

Długość klucza	Key Size	256 bits / 32 bytes
Tryb szyfru	Cipher Mode	CBC (Cipher Block Chaining)
Dopełnienie	Padding	PKCS#7
Rozmiar bloku	Block Size	16 bytes
Wektor inicjujący	Initialization Vector	16 bytes

1.4. Typ dokumentu

W zależności od rodzaju przesyłanego pliku, musi posiadać on odpowiedni typ dokumentu zwarty w schemacie. Dostępne są następujące typy dokumentów:

1. JPK dla plików JPK, CUK, ALK, ITP, DPI, PSP-FR, GIR.
2. JPKAH dla plików JPK na żądanie.
3. XML dla plików PSP-IP.

Typ dokumentu umieszczony jest w DocumentType, przykład użycia:

„<DocumentType>JPK</DocumentType>”

2. Specyfikacja interfejsu przyjmującego dokumenty JPK dla klientów

2.1. Wstęp

System przyjęć dokumentów korzysta z architektury RESTful, działającej za pośrednictwem protokołu HTTPS.

2.2. Opis interfejsu

Zasadnicza część interfejsu dla klientów ERP składa się z następujących metod:

- InitUploadSigned
- Put Blob
- FinishUpload
- Status

Komunikacja z interfejsem jest realizowana z wykorzystaniem protokołu TLS w wersji 1.2 oraz 1.3.

Dopuszczone są wyłącznie następujące zestawy szyfrów:

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Implementacja środowiska testowego dostępna jest pod adresem:

https://test-e-dokumenty.mf.gov.pl/

Natomiast adresy poszczególnych metod przedstawiają się następująco:

https://test-e-dokumenty.mf.gov.pl/api/Storage/InitUploadSigned

https://test-e-dokumenty.mf.gov.pl/api/Storage/FinishUpload

https://test-e-dokumenty.mf.gov.pl/api/Storage/Status/{referenceNumber}

Adresy magazynów chmurowych do których wysyłane są pliki JPK:

https://taxdocumentstorage00tst.blob.core.windows.net

https://taxdocumentstorage01tst.blob.core.windows.net

https://taxdocumentstorage02tst.blob.core.windows.net

https://taxdocumentstorage97tst.blob.core.windows.net

https://taxdocumentstorage98tst.blob.core.windows.net

https://taxdocumentstorage99tst.blob.core.windows.net

Implementacja środowiska produkcyjnego dostępna jest pod adresem:

https://e-dokumenty.mf.gov.pl/

Natomiast adresy poszczególnych metod przedstawiają się następująco:

https://e-dokumenty.mf.gov.pl/api/Storage/InitUploadSigned

https://e-dokumenty.mf.gov.pl/api/Storage/FinishUpload

https://e-dokumenty.mf.gov.pl/api/Storage/Status/{referenceNumber}

Adresy magazynów chmurowych do których wysyłane są pliki JPK:

https://taxdocumentstorage00.blob.core.windows.net

https://taxdocumentstorage01.blob.core.windows.net

https://taxdocumentstorage02.blob.core.windows.net

https://taxdocumentstorage97.blob.core.windows.net

https://taxdocumentstorage98.blob.core.windows.net

https://taxdocumentstorage99.blob.core.windows.net

wykorzystywane nazwy domenowe można weryfikować za pomocą wyrażenia regularnego:

https:[/]{2}taxdocumentstorage[0-9]{2}.blob.core.windows.net[/](.*)

Poniżej znajduje się szczegółowy opis działania metod.

2.2.1. InitUploadSigned

Metoda inicjująca sesję klienta. Jej wywołanie jest warunkiem koniecznym do przesłania danych metodą Put Blob usługi Azure.

Nazwa	**InitUploadSigned**
Typ metody	POST
Typ przesyłanej zawartości	application/xml
Typ zwracanej zawartości	application/json
Maksymalny rozmiar żądania	100KB

Opis parametrów przekazywanych w adresie metody:

Nazwa	Opis	Typ	Walidacja
enableValidateQualifiedSignature	W przypadku przekazania wartości **true (na środowisku testowym)**, system zweryfikuje czy przesyłany plik został podpisany poprawnym podpisem kwalifikowanym polskim albo europejskim lub podpisem zaufanym.	bool	Opcjonalny – dopuszczalne wartości: **true**, **false**

Adres metody z włączoną weryfikacją podpisu kwalifikowanego:
https://test-e-dokumenty.mf.gov.pl/api/Storage/InitUploadSigned?enableValidateQualifiedSignature=true

Opis struktury XML stanowiącego zawartość żądania (message body):

Nazwa	Opis	Typ	Walidacja
InitUpload	Metadane dla metody InitUpload	Obiekt	Wymagany
DocumentType	Nazwa typu przesyłanego dokumentu	String	Wymagany – dopuszczalne wartości: JPK – Dokumenty XML zgodne ze schemą wydaną przez Ministerstwo Finansów, przesyłane cyklicznie JPKAH – Dokumenty XML zgodne ze schemą wydaną przez Ministerstwo Finansów i przesyłane na żądanie w ramach kontroli XML – Dokumenty XML zgodne ze schemą wydaną przez podmioty inne niż Ministerstwo

Nazwa	Opis	Typ	Walidacja
			Finansów (dla PSP-IP(4))
Version	Wersja REST API do której adresowane jest zapytanie	String	Wymagany, 01.02.01.20160617 Wymagany, 01.03.01.20231001 (dla PSP-IP(4))
EncryptionKey	Klucz symetryczny zaszyfrowany algorytmem asymetrycznym (RSA)	String	Wymagany
EncryptionKey.algorithm	Algorytm, którym zaszyfrowany jest klucz symetryczny	String – dopuszczalne wartości: RSA	Wymagany
EncryptionKey.mode	Tryb szyfrowania	String – dopuszczalne wartości: ECB	Wymagany
EncryptionKey.padding	Format dopełnienia klucza szyfrującego	String – dopuszczalne wartości: PKCS#1	Wymagany
EncryptionKey.encoding	Algorytm kodowania wartości klucza	String – dopuszczalne wartości: Base64	Wymagany
DocumentList	Lista przesłanych dokumentów	Lista obiektów typu Document	Wymagany. Lista musi zawierać dokładnie jeden dokument

Nazwa	Opis	Typ	Walidacja
Document	Metadane przesyłanego dokumentu	Obiekt	Wymagany
FormCode	Kod Formularza zawarty w nagłówku pliku XML	String	Wymagany
FormCode.systemCode	Atrybut kodSystemowy elementu KodFormularza z pliku XML	String	Wymagany
FormCode.schemaVersion	Atrybut wersjaSchemy elementu KodFormularza z pliku XML	String	Wymagany
FileName	Nazwa pliku	String	Wymagany, unikalny, format: [a-zA-Z0-9_\.\-]{5,55} na przykład JPK_VAT_2016-07-01.xml
ContentLength	Całkowity rozmiar dokumentu	Long	Wymagany
HashValue	Skrót całego dokumentu	String	Wymagany
HashValue.algorithm	Nazwa algorytmu funkcji skrótu	String – dopuszczalne wartości: SHA-256	Wymagany
HashValue.encoding	Algorytm kodowania wartości funkcji skrótu	String – dopuszczalne wartości: Base64	Wymagany
FileSignatureList	Metadane plików wchodzących w skład dokumentu. W przypadku gdy rozmiar przesyłanego	Lista obiektów typu FileSignature	Wymagany. Lista musi zawierać

Nazwa	Opis	Typ	Walidacja
	dokumentu jest mniejszy niż 60MB to lista składa się tylko z jednego pliku		przynajmniej jeden element
FileSignatureList.filesNumber	Liczba wszystkich części pliku	int	Wymagany
Packaging	Możliwe rodzaje podziału i kompresji dokumentu	Lista wyboru	Wymagany
SplitZip	Rodzaj podziału i kompresji dokumentu	Obiekt	Wymagany
SplitZip.type	Rodzaj metody dzielącej dokument na części	String – dopuszczalne wartości: split	Wymagany
SplitZip.mode	Rodzaj algorytmu kompresji	String – dopuszczalne wartości: zip	Wymagany
Encryption	Możliwe metody szyfrowania plików cząstkowych	Lista wyboru	Wymagany
AES	Metoda szyfrowania plików cząstkowych	Obiekt	Wymagany
AES.size	Rozmiar klucza szyfrującego w bitach	Int – dopuszczalne wartości: 256	Wymagany
AES.block	Rozmiar bloku szyfrującego w bajtach	Int – dopuszczalne wartości: 16	Wymagany

Nazwa	Opis	Typ	Walidacja
AES.mode	Tryb szyfrowania	String – dopuszczalne wartości: CBC	Wymagany
AES.padding	Metoda dopełnienia bloku szyfrującego	String – dopuszczalne wartości: PKCS#7	Wymagany
IV	Wektor inicjujący algorytmu szyfrującego	String	Wymagany
IV.bytes	Rozmiar wektora inicjującego w bajtach	String – dopuszczalne wartości: 16	Wymagany
IV.encoding	Metoda kodowania wartość wektora inicjującego	String – dopuszczalne wartości: Base64	Wymagany
FileSignature	Metadane pliku	Obiekt	Wymagany
OrdinalNumber	Liczba porządkowa kolejnej części	Int	Wymagany, unikalny
FileName	Nazwa pliku przesyłanego do Azure Storage	String	Wymagany, unikalny, format: [a-zA-Z0-9_\.\-]{5,55} na przykład JPK_VAT_2016-07-01.xml.zip.001.aes
ContentLength	Długość pliku przesyłanego do Azure Storage	Int	Wymagany. Maksymalny rozmiar

Nazwa	Opis	Typ	Walidacja
			to 62914560 bajtów (60MB)
HashValue	Wartość funkcji skrótu pliku przesyłanego do Azure Storage, zakodowana w Base64 (nie należy konwertować do hex-a przed konwersją do Base64)	String	Wymagany. Długość: 24 znaki
HashValue.algorithm	Nazwa algorytmu funkcji skrótu	String – dopuszczalne wartości: MD5	Wymagany
HashValue.encoding	Algorytm kodowania wartości funkcji skrótu	String – dopuszczalne wartości: Base64	Wymagany
AuthData	To opcjonalne pole powinno zawierać dokument XML zgodny z opublikowaną schemą SIG-2008_v2-0.xsd zaszyfrowany z wykorzystaniem algorytmu symetrycznego AES256. Powinien zostać wykorzystany ten sam klucz, który jest wykorzystywany do szyfrowania części skompresowanego archiwum pliku JPK i załączany do niniejszego pliku metadanych. Algorytm kodowania	String	Opcjonalny

Nazwa	Opis	Typ	Walidacja
	zaszyfrowanych danych to Base64		

Skrót pliku przesyłanego do Storage (element **HashValue** w typie **FileSignatureType**) to wartość funkcji skrótu zgodnie z MD5 zakodowana następnie za pomocą Base64.

Schemat XSD pliku w formacie XML stanowiącego treść żądania jest udostępniony na stronie https://www.podatki.gov.pl/jednolity-plik-kontrolny/ w sekcji „JPK_VAT z deklaracją" link „Pliki do pobrania". We wskazanej lokalizacji umieszczono przykład metadanych podpisanych w formacie XAdES-BES certyfikatem niekwalifikowanym (self-signed).

Schemat XSD dla plików PSP znajduje się na stronie: https://www.gov.pl/web/kas/dostawcy-uslug-platniczych.

Schemat XSD dla plików DPI znajduje się na stronie: https://www.gov.pl/web/kas/struktury-dpi-dla-operatorow-platform.

Metoda InitUploadSigned zwraca trzy typy odpowiedzi:

Kod odpowiedzi	Opis
200 – OK	Poprawnie rozpoczęto sesję
400 – Bad Request	Nieprawidłowe zapytanie. Błędne wywołanie usługi
500 – Server Error	Błędne przetwarzanie zapytania

Opis struktury JSON (application/json) odpowiedzi (200 – OK):

Nazwa	Opis	Typ
ReferenceNumber	Identyfikator rozpoczętej sesji	String
TimeoutInSec	Czas życia (w sekundach) klucza uwierzytelniającego do wysłania dokumentów (uzależniony od liczby zadeklarowanych plików do wysyłki)	Int
RequestToUploadFileList	Lista metadanych wykorzystywanych do zbudowania żądania wysłania plików do Azure Storage	Lista obiektów typu RequestToUploadFile

Nazwa	Opis	Typ
RequestToUploadFile	Metadane wykorzystywane do zbudowania żądania wysłania pliku do Azure Storage	Obiekt
BlobName	Nazwa bloba do którego będzie zapisany plik	String
FileName	Nazwa pliku	String
Url	Adres, do którego nastąpi wysłanie pliku metodą *Put Blob.* Adres jest generowany dynamicznie i jego schemat może ulec zmianie.	String
Method	Metoda przesłania żądania *Put Blob*	String
HeaderList	Lista nagłówków wymaganych do utworzenia żądania *Put Blob.* Zwracane headery są generowane dynamicznie. Ich nazwy jak i ilość elementów może ulec zmianie.	Lista kluczy i wartości
Key	Klucz nagłówka	String
Value	Wartość nagłówka	String

Przykład treści poprawnej odpowiedzi (200 – OK):


```
{
 "ReferenceNumber": "d4fd41850323d2f6000000b013016327",
 "TimeoutInSec": 900,
 "RequestToUploadFileList": [
  {
   "BlobName": "8377ed3d-1b05-4c76-b718-6fddd46fd298",
   "FileName": "jpk_vat_100-01.xml.zip.aes",
   "Url":
"https://taxdocumentstorage09tst.blob.core.windows.net/d4fd41850323d2f6000000b013016327/8377
ed3d-1b05-4c76-b718-6fddd46fd298?sv=2015-07-
08&sr=b&si=d4fd41850323d2f6000000b013016327&sig=yFXyJdsPPkbE0iQwVs5ccLEYEU0lxQHldbVyPfPc
iXw%3D",
   "Method": "PUT",
   "HeaderList": [
    {
     "Key": "Content-MD5",
     "Value": "eXkPLHMM+dHB5GCFoeAvsA=="
    },
    {
     "Key": "x-ms-blob-type",
     "Value": "BlockBlob"
    }
   ]
  },
  {
   "BlobName": "0a80a089-bc10-41e1-a74d-70fd45f27aa3",
   "FileName": "jpk_vat_100-02.xml.zip.aes",
   "Url":
"https://taxdocumentstorage09tst.blob.core.windows.net/d4fd41850323d2f6000000b013016327/0a80
a089-bc10-41e1-a74d-70fd45f27aa3?sv=2015-07-
08&sr=b&si=d4fd41850323d2f6000000b013016327&sig=Fj%2BGjn7hCKIM6hSvMBGWBxSOyV7V%2FLM
M9pnenbaoxks%3D",
   "Method": "PUT",
   "HeaderList": [
    {
     "Key": "Content-MD5",
     "Value": "NZew85QTb16mFLzx9cyKzA=="
    },
    {
     "Key": "x-ms-blob-type",
```

```
      "Value": "BlockBlob"
     }
    ]
   }
 ]
}
```


Odpowiedź dla przykładu pliku podpisanego certyfikatem niekwalifikowanym w formacie XAdES-BES (enveloping) zamieszczonego na stronie w archiwum JPK-VAT-TEST-0001.ZIP:


```
{
 "ReferenceNumber": " ef7d17780087346e0000004c0c7982ec",
 "TimeoutInSec": 900,
 "RequestToUploadFileList": [
  {
   "BlobName": "094951bc-ba54-404e-b2c8-df2591ad0e17",
   "FileName": "JPK-VAT-TEST-0001.xml.zip.aes",
   "Url":
"https://taxdocumentstorage03tst.blob.core.windows.net/ef7d17780087346e0000004c0c7982ec/094951bc-ba54-404e-b2c8-df2591ad0e17?sv=2015-07-08&sr=b&si=ef7d17780087346e0000004c0c7982ec&sig=kN7LlprYkIP9uxod%2F1gcaDGN8WjbEbfDIA4GXuuzOmk%3D",
   "Method": "PUT",
   "HeaderList": [
    { "Key": "Content-MD5", "Value": "5YnivEH4gz5Wg5E8M2XwAQ==" },
    { "Key": "x-ms-blob-type","Value": "BlockBlob" }
   ]
  }
 ]
}
```


Odpowiedź dla przykładu pliku podpisanego certyfikatem niekwalifikowanym w formacie XAdES-BES (enveloped) zamieszczonego na stronie w archiwum JPK-VAT-TEST-0000.ZIP:


```
{
 "ReferenceNumber": " ef81ecf9011a546c0000004d72be8011",
 "TimeoutInSec": 900,
 "RequestToUploadFileList": [
  {
   "BlobName": "55a19799-5f1d-4336-9051-197dc53e5adf",
   "FileName": "JPK-VAT-TEST-0001.xml.zip.aes",
   "Url":
```

"https://taxdocumentstorage02tst.blob.core.windows.net/ef81ecf9011a546c0000004d72be8011/55a19799-5f1d-4336-9051-197dc53e5adf?sv=2015-07-08&sr=b&si=ef81ecf9011a546c0000004d72be8011&sig=HeLYQd8RfRucs4KGgWxITEU36OgQuqSe1RUXZ10n8%2Bs%3D",
"Method": "PUT",
"HeaderList": [
{ "Key": "Content-MD5", "Value": "5YnivEH4gz5Wg5E8M2XwAQ=="},
{ "Key": "x-ms-blob-type", "Value": "BlockBlob" }
]
}
]
}

Opis struktury JSON (application/json) odpowiedzi (400 – Bad Request):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
Code	Kod błędu	String
Errors	Opcjonalnie. Tablica błędów	Lista stringów
RequestId	Unikalny identyfikator błędnego żądania	GUID

Wyszczególnienie kodów zawartych w odpowiedzi (400 – Bad Request):

Code	Komunikat	Opis
100	Niepoprawny XML	Podany dokument nie jest dokumentem XML
101	Nieprawidłowa deklaracja kodowania znaków w pliku xml	Podany dokument zawiera nieprawidłową deklarację kodowania znaków (inną niż <?xml version="1.0" encoding="utf-8"?>)
110	Niepodpisany dokument	Podany dokument jest niepodpisany zgodnie ze specyfikacją

Code	Komunikat	Opis
111	Podpis jest złożony w innym formacie niż XAdES-BES	Plik InitUpload musi być podpisany zgodnie z formatem XAdES-BES (w tym BES-T ze znacznikiem czasu)
112	Niepoprawnie złożony podpis. Niemożliwa weryfikacja	W trakcie weryfikacji podpisu wystąpił nieoczekiwany błąd
113	Podpis złożony w nieobsługiwanym formacie zewnętrznym (detached)	Obsługiwane formaty podpisu to enveloped i enveloping
114	Problem z odczytaniem podpisanego obiektu	
115	Błąd złożonego podpisu. Brak referencji do podpisanego dokumentu xml	Prawidłowy podpis XAdES-BES powinien zawierać dwie referencje
116	Dokument z certyfikatem bez wymaganych atrybutów	Niepoprawny NIP lub PESEL
120	Podpis negatywnie zweryfikowany	Nie udało się poprawnie zweryfikować podpisu
130	Referencje w podpisie zostały negatywnie zweryfikowane. Dane prawdopodobnie zostały zmodyfikowane	Każda zmiana pliku po podpisaniu skutkuje negatywnym wynikiem weryfikacji podpisu
131	Nieznany certyfikat dostawcy usług zaufania	Certyfikat podpisującego został wystawiony przez dostawcę usług zaufania, który nie znajduje się na liście kwalifikowanych dostawców uznawanych w Polsce
132	Nieprawidłowy typ dokumentu	Dozwolone typy dokumentów to JPK, JPKAH, XML
133	Nie ma już możliwości przesłania tego dokumentu	Możliwość przyjęcia danego pliku została już zakończona
134	Nieprawidłowy typ podpisu	Dokument został podpisany innym typem podpisu niż wymagany dla tego rodzaju pliku

Code	Komunikat	Opis
136	Dokument zawiera podpis kwalifikowany i dane autoryzujące	Dokument nie może jednocześnie zawierać podpisu kwalifikowanego i danych autoryzujących. Należy zastosować tylko jedną metodę uwierzytelnienia
137	Dokument z błędnym podpisem kwalifikowanym - błędny certyfikat	
138	Dokument z błędnym podpisem kwalifikowanym - certyfikat stracił ważność	
139	Wielokrotny podpis przesłanego dokumentu xml z kodem formularza {nazwa_typu_dokumentu} jest niedozwolony	
140	Przesłany plik jest niezgody ze schematem XSD	Nie udało się zweryfikować dokumentu zgodnie ze schematem InitUpload.xsd
141	Nieokreślony błąd podczas sprawdzania pliku metadanych ze schematem xsd	
150	Nieobsługiwany kod formularza: „konkretny systemCode”	Kod formularza jest nieobsługiwany
155	Przesłany plik jest niepoprawny. Zadeklarowano co najmniej dwa pliki cząstkowe o takim samym skrócie	Błąd dotyczy zadeklarowania w pliku Initupload co najmniej dwóch plików cząstkowych o takim samym skrócie
156	Dołączanie załączników do dokumentu z kodem formularza {nazwa_typu_dokumentu} jest niedozwolone.	
157	Deklarowany całkowity rozmiar dokumentu musi być większy od 0	
158	Zbyt dużo wysłanych plików InitUpload. Proszę spróbować ponownie później	Błąd oznaczający przekroczenie dozwolonej liczby wysłanych plików w danym okresie

Code	Komunikat	Opis
159	Rozmiar dokumentu jest za duży	Błąd zwracany w przypadku przekroczenia maksymalnego dopuszczalnego rozmiaru dokumentu. Szczegółowe limity rozmiaru zostały określone w niniejszej specyfikacji.
160	Wartość „konkretny HashValue” nie jest zakodowana w Base64	Skrót plików zadeklarowanych do przesłania musi być zakodowany w Base64
170	Przesłano duplikat przetworzonego dokumentu. Numer referencyjny oryginału: XXXXXXXX	Duplikaty są sprawdzane na podstawie wartości skrótu SHA-256 zadeklarowanego dokumentu JPK

Przykład odpowiedzi:


```
{
 "Message": "Podpis negatywnie zweryfikowany",
 "Code": 120,
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```


Opis struktury JSON (application/json) odpowiedzi (500 – Internal Server Error):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
RequestId	Unikalny identyfikator błędnego żądania	GUID

Przykład odpowiedzi:


```
{
 "Message": "Wewnętrzny błąd systemu ",
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```


2.2.2. Put Blob

Metoda wysyłająca zasadnicze dokumenty JPK. Jest to metoda bezpośrednio implementowana przez usługę przestrzeń magazynową Azure (Azure Storage).

Jej pełna dokumentacja dostępna jest pod adresem:

https://learn.microsoft.com/en-us/rest/api/storageservices/Put-Blob

Wysłanie za pomocą klienta http.

Adres żądania

https://<nazwa_konta_storage>.blob.core.windows.net/<reference_number>/<nazwa _bloba>

Pełny adres, do którego klient ma wysłać dokumenty JPK jest zwracany przez metodę InitUploadSigned. Częścią zwracanego adresu jest Shared Access Signature (SAS), jednorazowy klucz, umożliwiający klientowi umieszczenie dokumentów we wskazanym kontenerze. Klucz SAS jest generowany jednorazowo i jest ważny w zadanych ramach czasowych i w zadanym fragmencie przestrzeni Azure Storage – zapewnia więc wysoki poziom bezpieczeństwa.

Metoda żądania

Zwracana jest przez InitUploadSigned.

Nagłówek żądania

Zwracane są przez InitUploadSigned.

Wykorzystywane nagłówki żądań:

Nagłówek żądania	Opis
x-ms-blob-type	Wymagany. Określa rodzaj bloba. Dopuszczalna wartość to **BlockBlob**
Content-MD5	Opcjonalny. Wartość funkcji skrótu MD5. Ten skrót jest używany do weryfikacji integralności danych podczas transportu. Wykorzystując tę wartość, Azure Storage automatycznie sprawdza wartość skrótu danych które otrzymał z zadeklarowanymi. Jeśli obie wartości się różną, operacja zakończy się niepowodzeniem z kodem błędu 400 (Bad Request)

Treść żądania

W treści żądania zawarty jest wysyłany plik.

Pełna dokumentacja dotycząca nagłówków żądań – i innych szczegółów interakcji z Azure Storage – dostępna jest po wskazywanym już adresem:

https://msdn.microsoft.com/en-us/library/azure/dd179451.aspx

Metoda Put Blob zwraca odpowiedzi:

Kod odpowiedzi	Opis
201 – Created	Poprawnie przesłano plik do przestrzeni Azure
4xx	Błędne wywołanie usługi

Kod odpowiedzi	Opis
5xx	Błędne przetwarzanie zapytania

Odpowiedź (201 – Created):

Pusta zawartość odpowiedzi.

Odpowiedzi 4xx oraz 5xx zwracają informację o błędzie w postaci XML (application/xml):

Nazwa	Opis	Typ
Error	Element główny struktury	Object
Code	Opisowy kod błędu	String
Message	Komunikat błędu	String

Przykład:

```
<?xml version="1.0" encoding="utf-8"?>
<Error>
  <Code>AuthenticationFailed</Code>
  <Message>Server failed to authenticate the request. Make sure the value of Authorization header
is formed correctly including the signature.
RequestId:a5124e1c-0001-0056-06b3-ddc62c000000
Time:2016-07-14T09:40:13.7833645Z</Message>
  <AuthenticationErrorDetail>SAS identifier cannot be found for specified signed
identifier</AuthenticationErrorDetail>
</Error>
```

Wysłanie za pomocą SDK

Dostępne implementacje: .NET, Node.js, Java, C++, PHP, Ruby, Python, iOS, Xamarin.

https://azure.microsoft.com/en-gb/documentation/articles/storage-dotnet-how-to-use-blobs/

Przykład:

Wiadomość zwrócona przez InitUploadSigned:

```
{
 "ReferenceNumber": "d8cb2f0f014381ab000000b012f8a3d6",
 "TimeoutInSec": 900,
 "RequestToUploadFileList": [
  {
   "BlobName": "b42748d3-0660-4d81-afc2-3c250fbcdbef",
```

```
    "FileName": "jpk_vat_100.xml.zip.aes",
    "Url":
"https://taxdocumentstorage10tst.blob.core.windows.net/d8cb2f0f014381ab000000b012f8a3d6/b4274
8d3-0660-4d81-afc2-3c250fbcdbef?sv=2015-07-
08&sr=b&si=d8cb2f0f014381ab000000b012f8a3d6&sig=2y%2BZ3cjcyBbBnCM6Mw9a4EPN2KA%2B01kgf
9fro%2FK6Xgw%3D",
    "Method": "PUT",
    "HeaderList": [
     { "Key": "Content-MD5", "Value": "eXkPLHMM+dHB5GCFoeAvsA==" },
     { "Key": "x-ms-blob-type", "Value": "BlockBlob" }
    ]
   }
 ]
}
```


Wysyłka pliku w .NET:

```
var absoluteUri =
"https://taxdocumentstorage10tst.blob.core.windows.net/d8cb2f0f014381ab000000b012f8a3d6/b4274
8d3-0660-4d81-afc2-3c250fbcdbef";
var sas = "sv=2015-07-
08&sr=b&si=d8cb2f0f014381ab000000b012f8a3d6&sig=2y%2BZ3cjcyBbBnCM6Mw9a4EPN2KA%2B01kgf
9fro%2FK6Xgw%3D";
var blob = new CloudBlockBlob(new Uri(absoluteUri), new StorageCredentials(sas));
using (var stream = new FileStream("jpk_vat_100-01.xml.zip.aes", FileMode.Open))
    {
        blob.UploadFromStream(stream);
    }
```

2.2.3. FinishUpload

Metoda kończąca sesję. Jej wywołanie jest warunkiem koniecznym do prawidłowego zakończenia procedury wysyłania dokumentów. Sprawdzane są wtedy wymagane pliki, które używają nazwy i MD5 wartości zadeklarowanych w InitUploadSigned. Brak jej wywołania jest tożsamy z uznaniem, że sesja została przerwana.

Specyfikacja metody:

Nazwa	**FinishUpload**
Typ metody	POST
Typ przesyłanej zawartości	application/json

Nazwa	FinishUpload
Typ zwracanej zawartości	application/json
Maksymalny rozmiar żądania	100KB

Opis struktury JSON (application/json) stanowiącego zawartość żądania (message body):

Nazwa	Opis	Typ	Walidacja
ReferenceNumber	Identyfikator sesji	String	Wymagany
AzureBlobNameList	Lista nazw blobów, które znajdują się w Azure Storage	List stringów	Wymagany. Lista musi zawierać tyle elementów ile plików wysłaliśmy do Azure Storage

Przykład:

```
{
 "ReferenceNumber": "e8505c4703e5fd5b000000b04bc6f43f"
 "AzureBlobNameList": [
  "d1eadd0e-ccd5-44ab-85e7-2f2a552e7f17",
  "5c3ceb5f-8c5d-4720-9005-7c7d1d88f121"
 ],
}
```

Metoda FinishUpload zwraca trzy typy odpowiedzi:

Kod odpowiedzi	Opis
200 – OK	Poprawnie zakończona sesja
400 – Bad Request	Nieprawidłowe zapytanie. Błędne wywołanie usługi
500 – Server Error	Błędne przetworzenie zapytania

Odpowiedź (200 – OK):

Pusta zawartość odpowiedzi

Opis struktury JSON (application/json) odpowiedzi (400 – Bad Request):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
Errors	Opcjonalnie. Tablica błędów	Lista stringów
RequestId	Unikalny identyfikator błędnego żądania	GUID

Przykład:


```
{
 "Message": "Żądanie jest nieprawidłowe"
 "Errors": "['Reference number jest wymagany']"
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```


Opis struktury JSON (application/json) odpowiedzi (500 – Internal Server Error):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
RequestId	Unikalny identyfikator błędnego żądania	GUID

Przykład:


```
{
 "Message": "Wewnętrzny błąd systemu ",
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```


2.2.4. Status

Metoda zwraca Urzędowe Potwierdzenie Odbioru wysłanych dokumentów. Metoda ta jest częścią API dla klientów, dostępną z tej samej usługi co inne metody.

Nazwa	Status
Typ metody	GET
Typ przesyłanej zawartości	Query String
Typ zwracanej zawartości	application/json

Nazwa	Status
Maksymalny rozmiar żądania	100KB
Format	Status/ba96951d00635700000001726b6ec621

Opis przesyłanego parametru:

Nazwa	Opis	Typ	Walidacja
ReferenceNumber	ReferenceNumber – Identyfikator sesji	String	Wymagany

Metoda Status zwraca trzy typy odpowiedzi:

Kod odpowiedzi	Opis
200 – OK	Poprawnie zwrócono potwierdzenie
400 – Bad Request	Nieprawidłowe zapytanie. Błędne wywołanie usługi
500 – Server Error	Błędne przetwarzanie zapytania

Opis struktury JSON (application/json) odpowiedzi (200 – OK):

Nazwa	Opis	Typ
Code	Kod statusu	String
Description	Opis	String
Details	Szczegóły zdarzenia	String
Upo	Opcjonalne. Urzędowe poświadczenie odbioru	String
Timestamp	Znacznik czasu	Datetime

Przykład dla odpowiedzi 200 - OK:


```
{
   "Code": 200,
   "Description": "Przetwarzanie dokumentu zakończone poprawnie. Wygenerowano UPO",
   "Details": "33d6792e03cb513e000000465ed9cb5e",
   "Timestamp": "2016-06-17T09:37:40.773976+00:00",
   "Upo": "Dokument UPO w formacie XML zgody ze schemą"
```

}

Lista statusów:

Statusy są pogrupowane w następujący sposób:

1xx – Kody określające sytuacje związane ze stanem sesji (np. rozpoczęta, wygasła).

2xx – Kody określające prawidłowe zakończenie procesu przetwarzania dokumentu.

3xx – Kody informujące o fazie przetwarzania dokumentu.

4xx – Kody określające niewłaściwe zakończenie procesu przetwarzania dokumentu.

Lista statusów:

Kod status	Opis
100	Rozpoczęto sesję przesyłania plików
101	Odebrano X z Y zadeklarowanych plików
120	Sesja została poprawnie zakończona. Dane zostały poprawnie zapisane. Trwa weryfikacja dokumentu
200	Przetwarzanie dokumentu zakończone poprawnie, pobierz UPO
300	Nieprawidłowy numer referencyjny
401	Weryfikacja negatywna – dokument niezgodny ze schematem XSD
403	Dokument z niepoprawnym podpisem
405	Dokument z odwołanym certyfikatem
406	Dokument z certyfikatem z nieobsługiwanym dostawcą
407	Przesłałeś duplikat dokumentu. Numer referencyjny oryginału to XXXXXXXX
408	Dokument zawiera błędy uniemożliwiające jego przetworzenie
410	Przesłane pliki nie są prawidłowym archiwum ZIP
411	Weryfikacja negatywna – w systemie jest już złożony identyczny dokument
412	Dokument nieprawidłowo zaszyfrowany
413	Suma kontrolna dokumentu niezgodna z deklarowana wartością

Kod status	Opis
415	Przesłany rodzaj dokumentu nie jest obsługiwany w systemie
417	Dokument nieprawidłowo zaszyfrowany. Błąd odszyfrowania danych autoryzujących
418	Weryfikacja negatywna – dane autoryzujące niezgodne ze schematem XSD
419	Weryfikacja negatywna – błąd w danych autoryzujących
420	Brak aktualnego pełnomocnictwa/upoważnienia do podpisywania dokumentu
422	Weryfikacja negatywna – dokument złożony z użyciem danych autoryzujących może złożyć wyłącznie podatnik, będący osobą fizyczną
423	Dokument z certyfikatem bez wymaganych atrybutów
424	Weryfikacja negatywna – dokument nie może być podpisany z użyciem danych autoryzujących
425	Weryfikacja negatywna – niespójne dane
426	Nieprawidłowe kodowanie znaków w danych autoryzujących
427	Dokument z certyfikatem z nieprawidłową ścieżką
428	Błąd walidacji reguł biznesowych
429	Nieprawidłowe kodowanie znaków w dokumencie xml
430	Dokument z nieprawidłowym podpisem
432	Rozmiar dokumentu niezgodny z deklarowaną wartością
433	Rozmiar dokumentu jest za duży. Maksymalny dozwolony rozmiar pliku dla schemy {nazwa i kod schemy} to X GB.

Opis struktury JSON (application/json) odpowiedzi (400 – Bad Request):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
Errors	Opcjonalnie. Tablica błędów	Lista stringów

Nazwa	Opis	Typ
RequestId	Unikalny identyfikator błędnego żądania	GUID

Przykład:


```
{
 "Message": "Żądanie jest nieprawidłowe",
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```


Opis struktury JSON (application/json) odpowiedzi (500 – Internal Server Error):

Nazwa	Opis	Typ
Message	Komunikat błędu	String
RequestId	Unikalny identyfikator błędnego żądania	GUID

Przykład:


```
{
 "Message": "Wewnętrzny błąd systemu ",
 "RequestId": "172dc3cc-5b97-48de-91dd-6903587cba19"
}
```