Specyfikacja interfejsów usług Jednolitego Pliku Kontrolnego

Ministerstwo Finansów

Departament Informatyzacji

20 czerwca 2016

Wersja 1.5

Zmiany

Data	Wersja	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).
20.06.2016	1.5	1. Zmiany schematu XSD pliku metadanych: - ustalenie obsługiwanej wersji REST API - 01.02.01.20160617, - zmiana wyrażenia regularnego elementu FileName. 2. Uzupełnienie zbioru kodów odpowiedzi dla metody Status.

Spis treści

1 Przygotowanie danych JPK

1.1 Przygotowanie dokumentów JPK

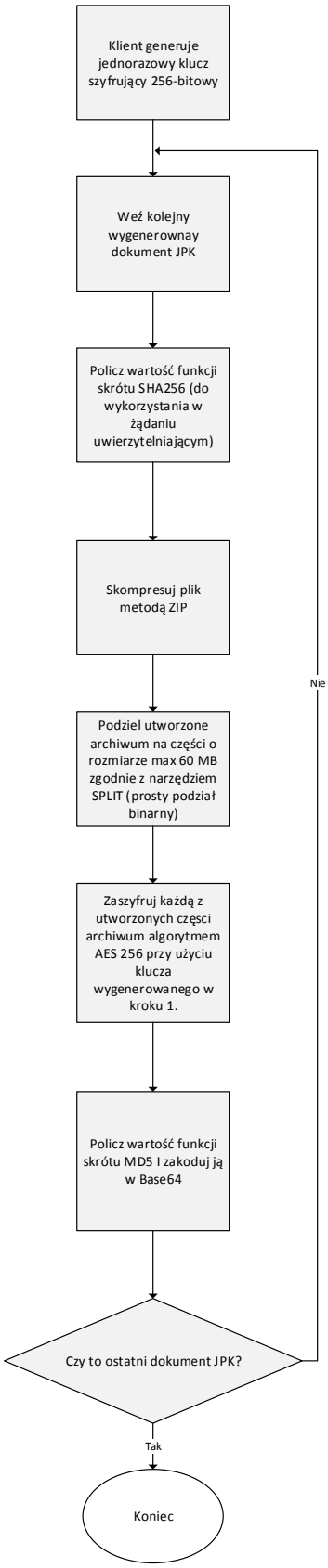
Dane JPK przygotowywane będą po stronie klienta (np. systemu ERP) w formie plików XML zgodnych ze schematem XSD opublikowanym przez Ministerstwo Finansów:

http://www.mf.gov.pl/kontrola-skarbowa/dzialalnosc/jednolity-plik-kontrolny/-/asset_publisher/2NoO/content/struktury-jpk

Każdy z dokumentów opisanych właściwym schematem ma stanowić osobny plik XML.

Wygenerowany plik XML powinien być zakodowany w UTF-8.

Dla każdego z plików JPK zostaną wykonane następujące operacje:


Klient generuje jednorazowy klucz szyfrujący 256-bitowy
Weź kolejny wygenerownay dokument JPK
Policz wartość funkcji skrótu SHA256 (do wykorzystania w żądaniu uwierzytelniającym)
Skompresuj plik metodą ZIP
Podziel utworzone archiwum na części o rozmiarze max 60 MB zgodnie z narzędziem SPLIT (prosty podział binarny)
Zaszyfruj każdą z utworzonych częsci archiwum algorytmem AES 256 przy użyciu klucza wygenerowanego w kroku 1.
Policz wartość funkcji skrótu MD5 I zakoduj ją w Base64
Czy to ostatni dokument JPK?
Nie
Tak
Koniec

Rysunek 14 Schemat blokowy kroków przygotowywania do wysyłki danych JPK

1.1.1 Kompresja danych JPK

Wygenerowany dokument JPK zostanie skompresowany algorytmem zip oraz dzielony na części o wielkości nie przekraczającej 60 MB (w praktyce należy spodziewać się wysokiego stopnia kompresji, dochodzącej nawet do 1:50, co sprawia, że scenariusz w którym będziemy mieli więcej niż jedną część, będzie stosunkowo rzadki).

Proponowana metoda kompresji to algorytm zip (deflate), natomiast dzielenie na części należy dokonać binarnie. Takie podejście z jednej strony zapewnia wykorzystanie znanych i powszechnie stosowanych narzędzi oraz łatwość implementacji dla różnych platform, z drugiej – efektywność, w szczególności operacji kompresji i prostotę API dla tych operacji.

1.1.2 Szyfrowanie danych JPK

Skompresowane pliki będą szyfrowane. Do szyfrowania plików wykorzystany będzie algorytm AES256, z kluczem szyfrującym wygenerowanym po stronie klienta. W implementacji mechanizmu szyfrowania należy użyć następującej specyfikacji algorytmu AES:

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

Algorytm procesu szyfrowania będzie wyglądał następująco:

- klient generuje losowy, 256 bitowy klucz,
- wygenerowanym kluczem szyfrowane są wszystkie części skompresowanego archiwum (zgodnie z pkt. 1.1) - algorytmem szyfrującym jest AES256.
- klucz szyfrujący jest szyfrowany z wykorzystaniem algorytmu asymetrycznego RSA, z wykorzystaniem kryptografii (klucz publiczny) dostarczonej podatnikowi przez Ministerstwo Finansów,

- zaszyfrowany klucz jest dołączany do pliku metadanych, zgodnie z przedstawionym poniżej opisem tego pliku.

1.2 Przygotowanie metadanych uwierzytelniających

Po przygotowaniu zasadniczych dokumentów zgodnych ze schematem Jednolitego Pliku Kontrolnego (JPK), klient, w celu wysłania danych, musi przygotować dane uwierzytelniające, mające postać odpowiedniego XML, przesłane w metodzie InitUploadSigned (opisanej w następnym rozdziale).

Plik metadanych musi być podpisany cyfrowo **podpisem kwalifikowanym** zgodnie z algorytmem XAdES Basic Electronic Signature, w skrócie XAdES-BES w wersji **Enveloped** (podpis jako dodatkowy element ds:Signature w oryginalnym XML) lub **Enveloping** (oryginalny dokument zawarty jako element w podpisanej strukturze).

Funkcją skrótu wykorzystywaną w podpisie powinna być RSA-SHA256 lub RSA-SHA1.

Przykład metadanych uwierzytelniających można znaleźć w p. 2.2.1, gdzie omówiona jest metoda InitUploadSigned, przyjmująca metadane uwierzytelniające.

2 Specyfikacja interfejsu przyjmującego dokumenty JPK dla klientów

2.1 Wstęp

Mechanizm przyjmowania dokumentów oparty jest o usługi REST, działające w oparciu o protokół HTTPS. Takie podejście zapewnia zarówno efektywność i sprawność interfejsu (choćby w porównaniu np. do interfejsów typu SOAP), jak i łatwość integracji z rozwiązaniami ERP i innymi, napisanymi w różnych technologiach.

2.2 Opis interfejsu

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

- InitUploadSigned
- Put Blob
- FinishUpload
- Status

Poniżej znajduje się szczegółowy opis działania tych 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 XML stanowiącego zawartość (body) żądania.

Nazwa	Opis	Typ	Walidacja
InitUpload	Metadane dla metody InitUpload	Obiekt	Wymagany
DocumentType	Nazwa typu przesyłanego dokumentu.	String	Wymagany - dopuszczalne wartości: **JPK** - dokumenty przesyłane cyklicznie **JPKAH** - dokumenty przesyłane doraźnie w ramach kontroli
Version	Wersja REST API do której adresowane jest zapytanie	String	Wymagany, format: [0-9]{2}\.[0-9]{2}\.[0-9]{2}\.[0-9]{8}, na przykład 01.01.01.20160430.
EncryptionKey	Klucz symetryczny zaszyfrowany algorytmem asymetrycznym (RSA)	String	Wymagany
EncryptionKey.algorithm	Algorytm, którym zaszyfrowany jest klucz	String – dopuszczalne wartości: **RSA**	Wymagany

	symetryczny		
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 przynajmniej jeden dokument.
Document	Metadane przesyłanego dokumentu	Obiekt	Wymagany
FormCode	KodFormularza zawarty w nagłówku pliku XML	String	Wymagany
FormCode.systemCode	Atrybut kodSystemowy elementu KodFormularza z pliku XML	String	Wymagany
FormCode.schemaVersion	Atrybut	String	Wymagany

	wersjaSchemy elementu KodFormularza z pliku XML		
FileName	Nazwa pliku JPK.	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 dokumentu jest mniejszy niż	Lista obiektów typu FileSignature	Wymagany. Lista musi zawierać przynajmniej jeden element

	60MB to lista składa się tylko z jednego pliku		
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:	Wymagany

		256	
AES.block	Rozmiar bloku szyfrującego w bajtach	Int – dopuszczalne wartości: **16**	Wymagany
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	String	Wymagany,

	przesyłanego do Azure Storage.		unikalny, format: [a-zA-Z0-9_\.\-]{5,55} na przykład JPK_VAT_2016-07-01.xml.bz2.001
ContentLength	Długość pliku przesyłanego do Azure Storage	Int	Wymagany. Maksymalny rozmiar to 62914560 bajtów (60MB)
HashValue	Wartość funkcji skrótu pliku przesyłanego do Azure Storage, zakodowana w 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

Skrót pliku przesyłanego do Storage (atrybut **HashValue** w type **FileSignature**) to wartość funkcji skrótu zgodnie z MD5 zakodowana następnie za pomocą Base64. Poniższy fragment kodu ilustruje to podejście:

```
var md5 = new MD5CryptoServiceProvider().ComputeHash(Encoding.Default.GetBytes(str));
var md5ToBase64 = Convert.ToBase64String(md5);
```

Schemat XSD dokumentu XML stanowiącego treść żądania:

initupload.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://e-dokumenty.mf.gov.pl" xmlns:mf="http://e-dokumenty.mf.gov.pl"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://e-dokumenty.mf.gov.pl"
elementFormDefault="qualified">
  <xs:element name="InitUpload" type="mf:InitUploadType"/>
  <xs:complexType name="InitUploadType">
    <xs:sequence>
      <xs:element name="DocumentType" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>JPK - dokumenty przesyłane cyklicznie, JPKAH - dokumenty przesyłane
doraźnie w ramach kontroli</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="JPK">
              <xs:annotation>
                <xs:documentation>Dokumenty przesyłane cyklicznie</xs:documentation>
              </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="JPKAH">
              <xs:annotation>
                <xs:documentation>Dokumenty przesyłane doraźnie w ramach kontroli</xs:documentation>
              </xs:annotation>
            </xs:enumeration>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Version" fixed="01.02.01.20160617" minOccurs="1" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:element>
      <xs:element name="EncryptionKey" maxOccurs="1">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="algorithm" use="required" fixed="RSA"/>
              <xs:attribute name="mode" use="required" fixed="ECB"/>
              <xs:attribute name="padding" use="required" fixed="PKCS#1"/>
              <xs:attribute name="encoding" use="required" fixed="Base64"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="DocumentList" type="mf:ArrayOfDocumentType" minOccurs="1"
maxOccurs="1">
        <xs:unique name="UniqueDocumentFileName">
          <xs:selector xpath="mf:Document"/>
          <xs:field xpath="mf:FileName"/>
        </xs:unique>
```

```
    </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="ArrayOfDocumentType">
  <xs:sequence>
   <xs:element name="Document" minOccurs="1">
    <xs:complexType>
     <xs:complexContent>
      <xs:extension base="mf:DocumentType"/>
     </xs:complexContent>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="DocumentType">
  <xs:sequence>
   <xs:element name="FormCode">
    <xs:annotation>
     <xs:documentation>KodFormularza zawarty w nagłówku pliku XML.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:string">
       <xs:attribute name="systemCode" type="xs:string" use="required">
        <xs:annotation>
         <xs:documentation>Atrybut kodSystemowy elementu KodFormularza z pliku
XML.</xs:documentation>
        </xs:annotation>
       </xs:attribute>
       <xs:attribute name="schemaVersion" type="xs:string" use="required">
        <xs:annotation>
         <xs:documentation>Atrybut wersjaSchemy elementu KodFormularza z pliku
XML.</xs:documentation>
        </xs:annotation>
       </xs:attribute>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="FileName">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_\.\-]{5,55}"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="ContentLength" type="xs:long"/>
   <xs:element name="HashValue" type="HashValueSHAType" minOccurs="1" maxOccurs="1"/>
   <xs:element name="FileSignatureList" minOccurs="1" maxOccurs="1">
    <xs:complexType>
     <xs:complexContent>
      <xs:extension base="mf:ArrayOfFileSignatureType">
       <xs:attribute name="filesNumber" use="required">
        <xs:simpleType>
         <xs:restriction base="xs:int">
          <xs:minInclusive value="1"/>
```

```
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:unique name="UniqueFileSignatureFileName">
    <xs:selector xpath="mf:FileSignature"/>
    <xs:field xpath="mf:FileName"/>
  </xs:unique>
  <xs:unique name="UniqueFileSignatureOrdinalNumber">
    <xs:selector xpath="mf:FileSignature"/>
    <xs:field xpath="mf:OrdinalNumber"/>
  </xs:unique>
 </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfFileSignatureType">
 <xs:sequence>
  <xs:element name="Packaging">
   <xs:complexType>
    <xs:choice>
     <xs:element name="SplitZip">
      <xs:complexType>
       <xs:attribute name="type" use="required" fixed="split"/>
       <xs:attribute name="mode" use="required" fixed="zip"/>
      </xs:complexType>
     </xs:element>
    </xs:choice>
   </xs:complexType>
  </xs:element>
  <xs:element name="Encryption">
   <xs:complexType>
    <xs:choice>
     <xs:element name="AES">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="IV">
         <xs:complexType>
          <xs:simpleContent>
           <xs:extension base="xs:string">
            <xs:attribute name="bytes" use="required" fixed="16"/>
            <xs:attribute name="encoding" use="required" fixed="Base64"/>
           </xs:extension>
          </xs:simpleContent>
         </xs:complexType>
        </xs:element>
       </xs:sequence>
       <xs:attribute name="size" type="xs:int" use="required" fixed="256"/>
       <xs:attribute name="block" type="xs:int" use="required" fixed="16"/>
       <xs:attribute name="mode" use="required" fixed="CBC"/>
       <xs:attribute name="padding" use="required" fixed="PKCS#7"/>
      </xs:complexType>
     </xs:element>
    </xs:choice>
   </xs:complexType>
```

```
    </xs:element>
    <xs:element name="FileSignature" type="mf:FileSignatureType" nillable="true" minOccurs="1"
maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FileSignatureType">
   <xs:sequence>
    <xs:element name="OrdinalNumber">
     <xs:simpleType>
      <xs:restriction base="xs:int">
       <xs:minInclusive value="1"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element name="FileName" minOccurs="1" maxOccurs="1">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:pattern value="[a-zA-Z0-9_\.\-]{5,55}"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element name="ContentLength" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="HashValue" type="HashValueMD5Type"/>
   </xs:sequence>
  </xs:complexType>
  <xs:complexType name="HashValueSHAType">
   <xs:simpleContent>
    <xs:extension base="xs:string">
     <xs:attribute name="algorithm" use="required" fixed="SHA-256"/>
     <xs:attribute name="encoding" use="required" fixed="Base64"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="HashValueMD5Type">
   <xs:simpleContent>
    <xs:extension base="xs:string">
     <xs:attribute name="algorithm" use="required" fixed="MD5"/>
     <xs:attribute name="encoding" use="required" fixed="Base64"/>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Przykładowa treść (body) żądania (dla czytelności pokazana jest treść bez elementów związanych z podpisem cyfrowym):

```
<?xml version="1.0" encoding="utf-8"?>
<InitUpload xmlns="http://e-dokumenty.mf.gov.pl">
  <DocumentType>JPK</DocumentType>
  <Version>01.02.01.20160617</Version>
  <EncryptionKey algorithm="RSA" mode="ECB" padding="PKCS#1"
encoding="Base64">hli8TTwUEND+Y4Ldz8quCS0zAvVwNuf1d3MSaMKeftVRsO7K/rUgiaZciTuEKb7ydr
```

```
ROxii0gARkGeSf3OrTsZ0fGRQxD4ZBJv437S9468X3W5VGOmPJUIKZqtMTWzYC+ZlKr3qfHy0WYBxve
dqWUwwUL5MZ38MURh85eBYSA9cl5iwpFPwa1Ig7oVMZDAhDtu9nOn+9l9s9ElkclkkmMxOCrXwWnTL
sRilWPLFvUZouwiGJGA/mlg7XhKysP63rHvqTmMKhOWhxSQZZphWMX/bb6Bcm+a+d0df+tT1jPHWEG
c9sThkgh+R0eMUiDLm4C9VwiNjHUdTyK1z1/aJuZQ==</EncryptionKey>
  <DocumentList>
    <Document>
      <FormCode systemCode="JPK_VAT (1)" schemaVersion="1-0">JPK_VAT</FormCode>
      <FileName>JPK_VAT_2016-07-01.xml</FileName>
      <ContentLength>1234567890</ContentLength>
      <HashValue algorithm="SHA-256" encoding="Base64">HashXml</HashValue>
      <FileSignatureList filesNumber="2">
        <Packaging>
          <SplitZip type="split" mode="zip"/>
        </Packaging>
        <Encryption>
          <AES size="256" block="16" mode="CBC" padding="PKCS#7">
            <IV bytes="16" encoding="Base64">MTIzNDU2Nzg5MDEyMzQ1Ng==</IV>
          </AES>
        </Encryption>
        <FileSignature>
          <OrdinalNumber>1</OrdinalNumber>
          <FileName>JPK_VAT_2016-07-01.xml.zip.001</FileName>
          <ContentLength>62914560</ContentLength>
          <HashValue algorithm="MD5" encoding="Base64">HgwPiDOm1nI2F81KI3npYw==</HashValue>
        </FileSignature>
        <FileSignature>
          <OrdinalNumber>2</OrdinalNumber>
          <FileName>JPK_VAT_2016-07-01.xml.zip.002</FileName>
          <ContentLength>123456</ContentLength>
          <HashValue algorithm="MD5" encoding="Base64"> dnF5x6K/8ZZRzpfSlMMM+w==</HashValue>
        </FileSignature>
      </FileSignatureList>
    </Document>
  </DocumentList>
</InitUpload>
```


Przykładowa treść (body) żądania (wraz z elementami związanymi z podpisem cyfrowym zgodnie z wymaganiami przedstawionymi w p. 1.2~~3.2~~


```
<?xml version="1.0" encoding="utf-8"?>
<InitUpload xmlns="http://e-dokumenty.mf.gov.pl">
  <DocumentType>JPK</DocumentType>
  <Version>01.02.01.20160617</Version>
  <EncryptionKey algorithm="RSA" mode="ECB" padding="PKCS#1"
encoding="Base64">hli8TTwUEND+Y4Ldz8quCS0zAvVwNuf1d3MSaMKeftVRsO7K/rUgiaZciTuEKb7ydr
ROxii0gARkGeSf3OrTsZ0fGRQxD4ZBJv437S9468X3W5VGOmPJUIKZqtMTWzYC+ZlKr3qfHy0WYBxve
dqWUwwUL5MZ38MURh85eBYSA9cl5iwpFPwa1Ig7oVMZDAhDtu9nOn+9l9s9ElkclkkmMxOCrXwWnTL
sRilWPLFvUZouwiGJGA/mlg7XhKysP63rHvqTmMKhOWhxSQZZphWMX/bb6Bcm+a+d0df+tT1jPHWEG
c9sThkgh+R0eMUiDLm4C9VwiNjHUdTyK1z1/aJuZQ==</EncryptionKey>
  <DocumentList>
    <Document>
      <FormCode systemCode="JPK_VAT (1)" schemaVersion="1-0">JPK_VAT</FormCode>
      <FileName>JPK_VAT_2016-07-01.xml</FileName>
      <ContentLength>1234567890</ContentLength>
      <HashValue algorithm="SHA-256" encoding="Base64">HashXml</HashValue>
```

```
    <FileSignatureList filesNumber="2">
      <Packaging>
        <SplitZip type="split" mode="zip"/>
      </Packaging>
      <Encryption>
        <AES size="256" block="16" mode="CBC" padding="PKCS#7">
          <IV bytes="16" encoding="Base64">MTIzNDU2Nzg5MDEyMzQ1Ng==</IV>
        </AES>
      </Encryption>
      <FileSignature>
        <OrdinalNumber>1</OrdinalNumber>
        <FileName>JPK_VAT_2016-07-01.xml.zip.001</FileName>
        <ContentLength>62914560</ContentLength>
        <HashValue algorithm="MD5" encoding="Base64">HgwPiDOm1nI2F81KI3npYw==</HashValue>
      </FileSignature>
      <FileSignature>
        <OrdinalNumber>2</OrdinalNumber>
        <FileName>JPK_VAT_2016-07-01.xml.zip.002</FileName>
        <ContentLength>123456</ContentLength>
        <HashValue algorithm="MD5" encoding="Base64"> dnF5x6K/8ZZRzpfSlMMM+w==</HashValue>
      </FileSignature>
    </FileSignatureList>
  </Document>
 </DocumentList>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>XT5th/g03u9CpJJNNPdKzYHg+sA=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>

<ds:SignatureValue>JJ1G4Ll4bnd8jE9H4gycyhp4imLvvEUqqjI8AUqdFWswvNFsJQJY3hp/8r8P7E3f3/ly8
E0njEeSG7RFp0F99xmQBCjkJhJ6Ha/MmdTkioSV5ZmUn6rljlusikjAxgdGY2mW/p8IoMJRR8GllOmQdPH
ZuqpCc6GuLEeoxD/8GUN52FU+wIAbSnoYO5S9bpW+KO5wfEfO0k1Uo/dDfoNQlOZt5WSLqqZYq9jaiB
BPOnRN/nXHa8dao961CgR/kiJcxJ+3J9iHMdfXVHt05iQv15OIpcuMS9AZePpazxVKVXmH3HfF6BqirNX
WyogXje+xmK0HbnbWZCewofZb4Sn2eA==</ds:SignatureValue>
   <ds:KeyInfo>
    <ds:KeyValue>
     <ds:RSAKeyValue>

<ds:Modulus>tVEy4LmCs7znT8V0Vnzu4VnMssRoM3Yblr9RbK33GtJAwiiMFBW+e+jXPQrIhqkNHUxkdR
phA/Il81UgX5BOzgULeDcDitMFVqHMcOVaeZPK5AmTJeGDvVjcZ5g8PRRaHfbP+wei7zUDt9Lt2lMccW
FWSg7z7UQwPsBj83Gj6ahzgq+Pu1W7Gz5stVgeAQN3zq++XNACulxT0kgY58NlZGqCov61ksT6W/MgR
x3Bo12LcWnfc1r0GhZiQfqWZXdcDPhhFosB/HgkJ8vm/0VB9Jg0dVb4fm4CPBPhNKKrxdxrHzRV8g6qd5
Ro0gxfM12xT+yK8u3MDWe/MpB5Q7dZ2Q==</ds:Modulus>
        <ds:Exponent>AQAB</ds:Exponent>
      </ds:RSAKeyValue>
    </ds:KeyValue>
    <ds:X509Data>
     <ds:X509IssuerSerial>
```

```
        <ds:X509IssuerName>Issuer</ds:X509IssuerName>

<ds:X509SerialNumber>265379454857972297697887687165046992692322305 6</ds:X509SerialNumb
er>
      </ds:X509IssuerSerial>
      <ds:X509SubjectName>Subject</ds:X509SubjectName>

<ds:X509Certificate>MIIGBDCCBOygAwIBAgITdwARQBAhKoGbpNyrHQAAABFAEDANBgkqhkiG9w0B
AQsFADA5MR4wHAYDVQQKExVNaWNyb3NvZnQgQ29ycG9yYXRpb24xFzAVBgNVBAMTDk1TSVQgT
kRFUyBDQSA0MB4XDTE2MDQyMjE3NTE0M1oXDTE2MDcyMTE3NTE0M1owgY4xEzARBgoJkiaJk/IsZ
AEZFgNjb20xGTAXBgoJkiaJk/IsZAEZFgltaWNyb3NvZnQxFDASBgoJkiaJk/IsZAEZFgRjb3JwMRYwFAY
KCZImiZPyLGQBGRYGZXVyb3BlMRUwEwYDVQQLEwxVc2VyQWNjb3VudHMxFzAVBgNVBAMTDlBpb
3RyIEJvbmluc2tpMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtVEy4LmCs7znT8V0Vnzu4
VnMssRoM3Yblr9RbK33GtJAwiiMFBW+e+jXPQrIhqkNHUxkdRphA/Il81UgX5BOzgULeDcDitMFVqHMcO
VaeZPK5AmTJeGDvVjcZ5g8PRRaHfbP+wei7zUDt9Lt2lMccWFWSg7z7UQwPsBj83Gj6ahzgq+Pu1W7G
z5stVgeAQN3zq++XNACulxT0kgY58NlZGqCov61ksT6W/MgRx3Bo12LcWnfc1r0GhZiQfqWZXdcDPhhFo
sB/HgkJ8vm/0VB9Jg0dVb4fm4CPBPhNKKrxdxrHzRV8g6qd5Ro0gxfM12xT+yK8u3MDWe/MpB5Q7dZ2
QIDAQABo4ICrTCCAqkwCwYDVR0PBAQDAgeAMCsGA1UdJQQkMCIGCisGAQQBgjcqAgEGCisGAQQ
BgjcUAgIGCCsGAQUFBwMCMB0GA1UdDgQWBBQ3YhTnGyptfSNGP7N0WvJCZfOvfDAvBgNVHREE
KDAmoCQGCisGAQQBgjcUAgOgFgwUcGlvdHJiQG1pY3Jvc29mdC5jb20wHwYDVR0jBBgwFoAUEcAD
pofSlyPZGKy2kl6mwiIn4+4wgccGA1UdHwSBvzCBvDCBuaCBtqCBs4YraHR0cDovL2NvcnBwa2kvY3JsL
01TSVQlMjBOREVTJTIwCisGAQQBgjcUAgIGCCsGAQUFBwMCMB0GA1UdDgQWBBQ3YhTnGyptfSN
GP7N0WvJCcC9jcmwvTVNJVCUyME5ERVMlMjBDQSUyMDQuY3JshkBodHRwOi8vY3JsLm1pY3Jvc29
mdC5jb20vcGtpL21zY29ycC9jcmwvTVNJVCUyME5ERVMlMjBDQSUyMDQuY3JsMIGTBggrBgEFBQcB
AQSBhjCBgzA3BggrBgEFBQcwAoYraHR0cDovL2NvcnBwa2kvYWlhL01TSVQlMjBOREVTJTIwQ0ElMjA
0LmNydDBIBggrBgEFBQcwAoY8aHR0cDovL3d3dy5taWNyb3NvZnQuY29tL3BraS9tc2NvcnAvTVNJVC
UyME5ERVMlMjBDQSUyMDQuY3J0MDwGCSsGAQQBgjcVBwQvMC0GJSsGAQQBgjcVCIPPiU2t8gKF
oZ8MgvrKfYHh+3SBT4bBwlTsrWYCAWQCAS0wNwYJKwYBBAGCNxUKBCowKDAMBgorBgEEAYI3Kg
IBMAwGCisGAQQBgjcUAgIwCgYIKwYBBQUHAwIwJQYDVR0gBB4wHDAMBgorBgEEAYI3KgEFMAwG
CisGAQQBgjcqARQwDQYJKoZIhvcNAQELBQADggEBAKfR7U4NaXk4xNRo/tMmb2OMTr4ofiHqD/66lS
H6esJ0Ap+9TOMxfXGnVa0B8H5A1lfW/HndGl8KmWuItHPPZIqJLTuwxIRETWFMmJWuLllqn/BfLUB+4D
WtcjZDTWvgET4gcX2VOr3utXthKd0kgfb1AyJY3Tw2cuqRvymBFuDC6s+jeg0L+NLI2ZWkV/MUoiH7Tpy2
65rv28tJrQvhoFJQSanbUQOMhG3chfY/3kMhz2pOjKaYZqWxlANuzxJpRVSo1aTyWbCVkFeDy7EGYzp
H8pQHr56MD6qUX+hEYBNl5/CrJJVfMsY2wJvyTOwLnmIrevgKlaEI5CWuHnfp2IA=</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</InitUpload>
```


Zwracane dane

Odpowiedzi

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

Odpowiedź 200 - OK:

Nazwa	Opis	Typ
ReferenceNumber	Identyfikator rozpoczętej sesji	String
TimeoutInSec	Czas życia (w sekundach) klucza uwierzytelniającego do wysłania dokumentów	Int
RequestToUploadFileList	Lista metadanych wykorzystywanych do zbudowania żądania wysłania plików do Azure Storage	Lista obiektów typu RequestToUploadFile
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	String
Method	Metoda przesłania żądania	String
HeaderList	Lista nagłówków wymaganych do utworzenia żądania	Lista kluczy i wartości
Key	Klucz nagłówka	String
Value	Wartość nagłówka	String

Przykład odpowiedzi:


```
{
 "ReferenceNumber": "ba96951d00635700000001726b6ec621",
 "TimeoutInSec": "7200",
 "RequestToUploadFileList": [
  {
   "BlobName": "a8b6f7db-e5f4-4541-b232-0e6a9017ca3f",
   "FileName": "jpkfile01.xml",
   "Url": "https://jpkstorageaccount03dev.blob.core.windows.net/container-004/a8b6f7db-e5f4-4541-b232-0e6a9017ca3f",
   "Method": "PUT",
   "HeaderList": [
    {
     "Key": "x-ms-date",
     "Value": "Mon, 16 May 2016 17:21:51 GMT"
    },
    {
     "Key": "x-ms-version",
     "Value": "2015-04-05"
    },
    {
     "Key": "Content-MD5",
     "Value": "eu/k4pzvymH+SYVs1F8MAg=="
    },
    {
     "Key": "x-ms-blob-type",
     "Value": "BlockBlob"
    },
    {
```

```
        "Key": "Content-Type",
        "Value": "application/xml"
      },
      {
        "Key": "Authorization",
        "Value": "SharedKey jpkstorageaccount03dev:Gf565UNo7q7ymIw2rGdg4LDM4z+M3BbTbXedg+Xt7Mk="
      }
    ]
  },
  {
    "BlobName": "2a3bfb5d-e817-404c-9e7a-5d819fdd4df7",
    "FileName": "jpkfile02.txt",
    "Url": "https://jpkstorageaccount03dev.blob.core.windows.net/container-004/2a3bfb5d-e817-404c-9e7a-5d819fdd4df7",
    "Method": "PUT",
    "HeaderList": [
      {
        "Key": "x-ms-date",
        "Value": "Mon, 16 May 2016 17:21:51 GMT"
      },
      {
        "Key": "x-ms-version",
        "Value": "2015-04-05"
      },
      {
        "Key": "Content-MD5",
        "Value": "eu/PE54vymH+SYVs238MAg=="
      },
      {
        "Key": "x-ms-blob-type",
```

```
        "Value": "BlockBlob"
      },
      {
        "Key": "Content-Type",
        "Value": "application/xml"
      },
      {
        "Key": "Authorization",
        "Value": "SharedKey jpkstorageaccount03dev:Tz7EqAl6OszIxGjBUk2qcxs82Af4Xq9CxyFx6u34LEI="
      }
    ]
  }
 ]
}
```


Odpowiedź 400 – Bad Request:

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

Code	Komunikat	Opis
100	Invalid xml	Podany dokument nie jest dokumentem xml
110	Unsigned xml document	Podany dokument jest niepodpisany zgodnie ze specyfikacją

120	Signature verification failed	Nie udało się poprawnie zweryfikować podpisu
130	Signed data was modified	Podpisane dane zostały zmodyfikowane
140	Schema validation failed	Nie udało się zweryfikować dokumentu zgodnie ze schematem xsd

Przykład odpowiedzi:

```
{
  "Message": " Signature verification failed",
  "Code": 120
}
```

Odpowiedź 500 – Internal Server Error

Nazwa	Opis	Typ
Message	Komunikat błędu	String

```
{
  "Message": "Internal server error ",
}
```

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

Schemat żądania http:

https://<nazwa_konta_storage>.blob.core.windows.net/<nazwa_kontenera>/<nazwa _blobu>

Dla przypomnienia – pełny adres, do którego klient ma wysłać dokumenty JPK jest zwracany przez metodę InitUpload. Częścią zwracanego adresu jest Shared Access Signature (SAS), jednorazowy klucz, umożliwiający klientowi na umieszczenie dokumentów we wskazanym kontenerze. Klucz SAS jest generowany jednorazowo i jest ważny tylko dla konkretnego przesyłanego pliku (weryfikacja wartości funkcji skrótu), w zadanych ramach czasowych i w zadanym fragmencie przestrzeni Azure Storage – zapewnia więc wysoki poziom bezpieczeństwa i gwarantuje, że wysłane zostaną pliki, dla których klucz SAS został wygenerowany

Nagłówek żądania

Wykorzystywane nagłówki żądań:

Nagłówek żądania	Opis
Authorization	Wymagany. Określa schemat uwierzytelniania, nazwę konta i podpis. Więcej informacji: Authentication for the Azure Storage Services.
Date* or *x-ms-date	Wymagany. Określa Coordinated Universal Time (UTC) dla żądania. Więcej informacji: Authentication for the Azure Storage Services.
x-ms-version	Wymagany dla wszystkich uwierzytelnionych żądań. Określa wersję interfejsu po stronie Azure dla operacji. Więcej informacji: Versioning for the Azure Storage Services.
x-ms-blob-type:	Wymagany. Określa rodzaj bloba. Dopuszczalna wartość to BlockBlob.

BlockBlob	
Content-MD5	Wymagany. 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).
Content-Type	MIME typ przesyłanego pliku

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

Treść żądania

W treści żądania zawarty jest wysyłany plik.

2.2.3 FinishUpload

Metoda kończąca sesję. Jej wywołanie jest warunkiem koniecznym prawidłowego zakończenia procedury wysyłania dokumentów. Brak jej wywołania jest tożsamy z uznaniem, że sesja została przerwana.

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

Opis treści (body) żądania:

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

Zwracane dane

Odpowiedzi

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

Odpowiedź 200 – Ok

Pusta zawartość odpowiedzi

Odpowiedz 400 – Bad Request:

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

Przykład:

```
{
  "Message": "The request is invalid."
  "Errors": "[‘Reference number jest wymagany’]"
}
```


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
Maksymalny rozmiar żądania	100KB
Format	Status/ba96951d00635700000001726b6ec621

Opis przesyłanego parametru

Nazwa	Opis	Typ	Walidacja
ReferenceNumber	ReferenceNumber - Identyfikator sesji	String	Wymagany

Odpowiedzi

Kod odpowiedzi	Opis

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

Odpowiedz 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


```
{
  "Code": 300,
  "Description": "Nieprawidłowy numer referencyjny",
  "Upo": ""
  "Details": ""
  "Timestamp": "2016-06-17T09:37:40.773976+00:00"
}
```

Odpowiedź 400 – Bad Request:

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

Przykład:

```
{
  "Message": "The request is invalid.",
}
```

Lista statusów:

Poniższa tabela prezentuje kody statusów wraz z ich opisami.

Statusy są pogrupowane w poniższy sposób:

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

2xx – Kody określające sytuacje, w których przetwarzanie dokumentów zakończyło się powodzeniem

3xx – Kody informujące o fazie przetwarzania dokumentu

4xx- 5xx Kody określające sytuacje, w których proces przetwarzania dokumentów zakończył się błędem..

Kod status	Opis
100	Rozpoczęto sesję przesyłania plików.
101	Odebrano X z Y zadeklarowanych dokumentów.
102	Proszę o ponowne przesłanie żądania UPO.
110	Sesja wygasła, nie przesłano zadeklarowanej liczby plików.
120	Sesja została poprawnie zakończona. Dane zostały poprawnie zapisane. Trwa

	weryfikacja dokumentu.
200	Przetwarzanie dokumentu zakończone poprawnie, pobierz UPO.
300	Nieprawidłowy numer referencyjny.
301	Dokument w trakcie przetwarzania, sprawdź wynik następnej weryfikacji dokumentu.
302	Dokument wstępnie przetworzony, sprawdź wynik następnej weryfikacji dokumentu.
303	Dokument w trakcie weryfikacji podpisu, sprawdź wynik następnej weryfikacji dokumentu.
401	Weryfikacja negatywna – dokument niezgodny ze schematem XSD.
403	Dokument z niepoprawnym podpisem.
404	Dokument z nieważnym certyfikatem.
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.
409	Dokument zawiera niewłaściwą ilość i/lub rodzaj elementów.
410	Przesłane pliki nie są prawidłowym archiwum ZIP.
411	Błąd podczas scalania dokumentu (dokument nieprawidłowo podzielony).
412	Dokument nieprawidłowo zaszyfrowany.

413	Suma kontrolna dokumentu niezgodna z deklarowana wartością.
414	Suma kontrolna części dokumentu (pliku …….) niezgodna z deklarowaną wartością.
415	Przesłany rodzaj dokumentu nie jest obsługiwany w systemie.

Odpowiedź 500 – Internal Server Error

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

```
{
  "Message": "Internal server error ",
}
```