



e-FATURA UYGULAMASI

(Yazılım Standartları ve Nesne Yapısı)

Ağustos 2014

ANKARA

GELİR İDARESİ BAŞKANLIĞI
Uygulama ve Veri Yönetimi Daire Başkanlığı

| Versiyon | Yayım Tarihi | Eklene/Silinen/Değişen Bölüm | Açıklama |
|----------|--------------|---|--|
| 1.0 | 05.03.2010 | -- | Kılavuzun ilk yayım tarihi |
| 1.1 | 26.04.2010 | 3 XAdES | 3. XMLDsign bölümü tamamen kaldırılarak XAdES olarak değiştirildi . |
| 1.2 | 14.03.2011 | 3 XAdES 4 Sıkıştırma Formatı, 6 SSL Mekanizması 7.7 DocumentType Sınıfı Metodları 8 Dönülen Aykırı Durum Kodları | Açıklamalar genişletildi . |
| 1.3 | 19.08.2013 | 7 Kullanılan Metot ve Sınıf Yapıları | Kullanılan Metot ve Sınıf Yapıları bölümüne açıklama eklendi . |
| 1.3 | 19.08.2013 | 8 Dönülen Aykırı Durum Kodları | Yeni durum kodu eklendi . |
| 1.4 | 31.08.2014 | 7.2.2 getApplicationResponse | Metodun entegratörler tarafından nasıl gerçekleştireceği anlatımı eklendi . |
| 1.4 | 31.08.2014 | 8 Dönülen Aykırı Durum Kodları | 2007 Durum kodunun Hata Açıklaması değiştirildi . |

| | | |
|-------|--|----|
| 1 | Giriş | 5 |
| 2 | MTOM (Message Transmission Optimization Mechanism) | 5 |
| 3 | XAdES | 6 |
| 4 | Sıkıştırma Standardı | 8 |
| 5 | Mesaj Özeti Standardı..... | 8 |
| 6 | SSL Mekanizması | 8 |
| 7 | Kullanılan Metot ve Sınıf Yapıları | 8 |
| 7.1 | Metot Özetleri | 9 |
| 7.2 | Metot Detayları | 9 |
| 7.2.1 | sendDocument..... | 9 |
| 7.2.2 | getApplicationResponse | 9 |
| 7.3 | Kullanılan Sınıf Özetleri | 11 |
| 7.4 | DocumentRequest Sınıfı Metotları..... | 11 |
| 7.5 | DocumentResponse Sınıfı Metotları | 12 |
| 7.6 | DocumentReturnType Sınıfı Metotları..... | 12 |
| 7.7 | DocumentType Sınıfı Metotları | 12 |
| 7.8 | GetAppRespRequest Sınıfı Metotları | 13 |
| 7.9 | GetAppRespRequestType Sınıfı Metotları | 13 |
| 7.10 | GetAppRespResponse Sınıfı Metotları | 14 |
| 7.11 | GetAppRespResponseType Sınıfı Metotları | 14 |
| 8 | Dönülen Aykırı Durum(Exception) Kodları..... | 14 |

1 Giriş

Bu kılavuz, Elektronik Fatura Uygulamasına entegre olmak isteyenler ile uygulamaya ilişkin yazılım geliştireceklerin uymak zorunda oldukları yazılım standartları ve nesne yapısı hakkında bilgi vermek amacıyla hazırlanmıştır.

Elektronik Fatura Uygulamasının genel özellikleri aşağıda yer almaktadır.

- Uygulama içerisinde teknolojik iletişim altyapısı olarak Web servis kullanılmıştır.
- Web servis iletişim altyapısı SOAP mesajları ile sağlanmaktadır. SOAP metotları için iki tür kodlama metodu bulunmaktadır. Bunlar RPC ve DOC kodlamalarıdır. Uygulamada performans açısından DOC kodlaması benimsenmiştir. SOAP mesajlarının iletiminde büyük dosyalar söz konusu olduğundan MTOM (Message Transmission Optimization Mechanism) mekanizması kullanılmıştır.
- Web Servis belirtiminin diğer web servis gerçekleştirmeleri ile sorunsuz çalışabilmesi için WS-I ve WS-I Basic Profile standartları web servis gerçekleştirmede dikkate alınmıştır.

Kılavuzun ilk bölümünde MTOM mekanizması genel olarak tanıtılacak, sonraki bölümlerde ise tanımlanan Web Servis metotları ve metotların kullandığı parametre, dönüş değeri ve nesnelere ilgili bilgilere yer verilecektir.

2 MTOM (Message Transmission Optimization Mechanism)

MTOM ikilik verilerin etkin bir biçimde web servislerden veya web servislere gönderimini sağlayan bir metottur. İkilik veriyi göndermek için XOP(XML-binary Optimized Packaging) kullanır. MTOM büyük miktarda ikilik veriyi SOAP mesajıyla gönderme imkânı sağlamakta olup aşağıda sıralanan 3 önemli yararı bulunmaktadır.

- Güvenlik: MTOM, güvenlikle oluşur, verilerin güvenliği SOAP mesaj kadar iyi korunur. Ayrıca DIME'nin desteklemediği bazı güvenlik seviyeleri de MTOM

ile desteklenmektedir. (Örneğin: TCP analizi DIME ile yapılan transferde gerçekleştirilirse, dosyanın görüldüğü görünür ancak MTOM, dosya içine SOAP mesajına uygulanan güvenliği uygulayabilmektedir.)

- Dosya Boyutunun Korunması: MTOM sayesinde hat üzerinde ekstra yük oluşmaz ve böylece dosya boyutu korunmuş olur.
- Basit Programlama Modeli: MTOM desteği etkinleştirilerek servisteki bütün byte transferleri otomatik olarak şifrelenir.

MTOM'in web servis ile uygulamaları çeşitli web servis gerçekleştirim ürünleriyle sağlanmıştır. Bunlara örnek axis2 ve Microsoft web servisleri örnek verilebilir.

3 XAdES

XAdES, Elektronik Fatura Uygulaması kapsamında gönderilecek olan belgelerin (Fatura veya Uygulama Yanıtı) imzalanması veya onaylanmasında kullanılmaktadır. XAdES imzalamada minimum olarak XAdES-BES kullanılmaktadır. XML imzalama için "Enveloped " tekniği kullanılmalıdır. "Enveloping" ve "Detached" teknikleri kabul edilmeyecektir.

e-Fatura onaylama işleminde bir faturanın XML imza olarak minimum aşağıdaki XML etiketlerini kullanması gerekmektedir.

```
<ds:Signature Id="Signature">
  <ds:SignedInfo Id="SignedInfo">
    <ds:CanonicalizationMethod/>
    <ds:SignatureMethod/>
    <ds:Reference URI=" " >
      <ds:Transforms>
        <ds:Transform/>
      </ds:Transforms>
      <ds:DigestMethod/>
      <ds:DigestValue/>
    </ds:Reference>
    <ds:Reference URI="#SignedProperties" >
      <ds:DigestMethod />
      <ds:DigestValue />
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue/>
</ds:Signature>
```

```

<ds:KeyInfo>
  <ds:KeyValue/>
  <ds:X509Data>
    <ds:X509SubjectName/>
    <ds:X509Certificate/>
  </ds:X509Data>
</ds:KeyInfo>
<ds:Object>
  <xades:QualifyingProperties Target="Signature">
    <xades:SignedProperties Id="SignedProperties">
      <xades:SignedSignatureProperties>
        <xades:SigningTime/>
        <xades:SigningCertificate>
          <xades:Cert>
            <xades:CertDigest>
              <ds:DigestMethod/>
              <ds:DigestValue/>
            </xades:CertDigest>
            <xades:IssuerSerial>
              <ds:X509IssuerName/>
              <ds:X509SerialNumber/>
            </xades:IssuerSerial>
          </xades:Cert>
        </xades:SigningCertificate>
        <xades:SignerRole>
          <xades:ClaimedRoles>
            <xades:ClaimedRole/>
          </xades:ClaimedRoles>
        </xades:SignerRole>
      </xades:SignedSignatureProperties>
    </xades:SignedProperties>
  </xades:QualifyingProperties>
</ds:Object>
</ds:Signature>

```

Mesaj özetlerinin güvenlik açısından SHA256 olması önerilmektedir. Yukarıda verilen XAdES xml imza iskeletinde bulunan alanlar mutlaka kullanılmalıdır. İskelette verilen nitelik(attribute) değerleri örnek teşkil etmesi amacıyla verilmiştir.

XML imzanın (mali mühür) doğrulanması için kullanılabilecek web siteleri ile çeşitli yazılımlar mevcut olup, entegrasyon çalışması yapan mükellefler bu kaynaklar yardımı ile XML imza (Mali Mühür) kontrolünü yapabilirler.

4 Sıkıştırma Standardı

Hazırlanan zarflar zip formatı ile sıkıştırılmalıdır. Zip dosyası sıkıştırma algoritması standart zip formatı olmalıdır. Gzip, rar gibi formatlar kullanılmamalıdır.

5 Mesaj Özeti Standardı

Elektronik Fatura Uygulaması aracılığı ile gönderilecek olan belgelerin Zarf XML'i içerisine konulması gerekmektedir. Bu XML, zip formatında sıkıştırıldıktan sonra Web servis aracılığı ile gönderilecektir. Web servis ile bu zip belgesine ait mesaj özeti de (hash) gönderilmesi uygulama içerisinde beklenen bir durumdur.

Mesaj özeti standardı olarak MD5 standardına uyulacaktır. Alınan ve gönderilen belgelere ait özet değerleri hesaplanacak ve gelen değerler ile karşılaştırılacaktır. Özet değerleri 32 karakter uzunluğunda olacaktır.

6 SSL Mekanizması

Web servis iletişimi HTTPS protokolü üzerinden SSL mekanizması ile sağlanacaktır. Buna göre istemci uygulaması SSL sunucu uygulamasının kimliğini doğrular. Bu durumda Elektronik Fatura Uygulamasında bulunan birimlerin güvenilir sunucu sertifikası temin etmeleri ihtiyacıdır. Mükelleflerin SSL sunucu sertifikalarını Verisign, GlobalSign gibi güvenilir sertifika sağlayıcılarından almaları gerekmektedir.

e-Fatura Uygulamasına entegre olacak kurumların uygulamaları SSL sunucu sertifikası ile çalışmalıdır. Temin edilen sertifikalar ile web servis yazılımının uyumu ve çalışabilirliği sağlanmalıdır.

7 Kullanılan Metot ve Sınıf Yapıları

Bu bölümde anlatılan metot ve sınıf yapıları Java programlama dili ve axis2 gerçekleştirimi ile oluşmuş örnek yapılarıdır. Oluşacak metot ve sınıf yapıları değişik programlama dilleri ve web servis gerçekleştirimleri ile değişebilir.

Web servis için kullanılacak WSDL belgesine www.efatura.gov.tr adresinde yer alan e-Fatura Paketinden ulaşılabilir. Entegratör kurumlar geliştirecekleri web servis uygulamalarındaki hem istemci hem de sunucu yazılımlarını bu WSDL'e göre geliştirmek zorundadırlar.

7.1 Metot Özetleri

| | |
|--------------------|--|
| DocumentResponse | sendDocument (DocumentRequest request) Kullanıcının zarfların göndereceği metottur. |
| GetAppRespResponse | getApplicationResponse (GetAppRespRequest request) Kullanıcının zarfların durumunu sorguladığı metottur. |

7.2 Metot Detayları

7.2.1 sendDocument

```
public DocumentResponse sendDocument(DocumentRequest request)  
                                throws EFaturaFaultMessage
```

Kullanıcının zarfların gönderileceği metottur. İkilik veriyi(Binary Data), dosya adını ve ikilik verinin özetini içeren DocumentRequest nesnesi alır ve kendi hesapladığı özet değeri ile mesajı içeren DocumentResponse nesnesini döner.

Throws: EFaturaFaultMessage

7.2.2 getApplicationResponse

```
public GetAppRespResponse  
getApplicationResponse(GetAppRespRequest request)  
                                throws EFaturaFaultMessage
```

Kullanıcının zarfların durumunu sorguladığı metottur. Zarf ID'yi içeren GetAppRespRequest nesnesini alır ve sistem yanıtını içeren GetAppRespResponse nesnesini döner.

Throws: EFaturaFaultMessage

Bu metodu Merkez Birimi dışında, e-Fatura sistemine entegre olan entegratörler de kendi sunucu yazılımlarına gerçekleştirmelidirler. Entegratörler zarfların durumunu Merkez Biriminden sorgulayabilmektedirler. Benzer şekilde bu metod aracılığıyla Merkez Birim de entegratörlerin sistemine düşen zarfların durumunu sorgulayabilecektir.

Merkez Birimi belirli aralıklarla, entegratörlerin almış olduğu fakat sistem yanıtı dönmediği zarfların (merkezdeki durumu 1220 olan zarflar) entegratörlerdeki durumunu entegratörlerin sisteminden sorgulayacaktır. Entegratörlerden gelecek olan sistem yanıtına göre zarfın merkezdeki durumu güncellenecektir. Ardından bu sistem yanıtı zarfların göndericilerine de iletilecektir.

Merkez Birimi sorgulama işlemini `GetAppRespRequestType` sınıfını kullanarak yapacaktır. Sorgulanacak zarfın numarası `instanceIdentifier` sınıf değişkeni kullanılarak iletilecektir. Sorgulama sonucu `GetAppRespResponseType` tipinde nesne dönülmelidir. Dönülen bu nesnenin `applicationResponse` sınıf değişkeni, sorgulanan zarfa ait sistem yanıtı içeren zarfın XML'i (String tipinde) olmalıdır. Dönülmesi gereken `applicationResponse` değerinin bir XML formatındaki bir örneği <http://www.efatura.gov.tr/efaturamevzuat.html> adresindeki e-Fatura Paketi içerisinde mevcuttur. (1_SISTEM_YANITI_POSTA_KUTUSU.xml)

Bu metodun entegratörler tarafından geliştirimi şu şekilde olmalıdır:

```
public GetAppRespResponseType getApplicationResponse(GetAppRespRequestType request) throws
EFaturaFaultMessage {

String instanceIdentifier = request.getInstanceIdentifier();
String appResponseStr = "<?xml version='1.0' encoding='UTF-8'?>"+
    "<sh:StandardBusinessDocumentHeader>"+
        "<sh:HeaderVersion>1.0</sh:HeaderVersion>"+
        "<sh:Sender>"+
            ....
            "<cac:Response>"+
                "<cbc:ReferenceID>bd0a35bd-2578-4aaa-9f94-e7906e196771</cbc:ReferenceID>"+
                "<cbc:ResponseCode>1200</cbc:ResponseCode>"+
                "<cbc:Description>.....</cbc:Description>"+
            "</cac:Response>"+
            ....
            "</ElementList>"+
        "</Elements>"+
    "</ef:Package>"+
    "</sh:StandardBusinessDocument>";

GetAppRespResponseType t = new GetAppRespResponseType();
t.setApplicationResponse(appResponseStr);
return t;
}
```

Sorgulama sonucu dönülen `applicationResponse` değerinin sistem yanıtlarından bir farkı yoktur. Entegratörler sistem yanıtlarını hazırlarken dikkat ettikleri kurallara bu metodun geliştirimini yaparken de dikkat etmelidir.

7.3 Kullanılan Sınıf Özetleri

| | |
|-------------------------------|--|
| DocumentRequest | sendDocument metoduna parametre olarak verilen sınıftır. |
| DocumentResponse | sendDocument metodundan dönen sınıftır. |
| DocumentReturntype | Mesaj ve özet niteliklerini içeren sınıftır. |
| DocumentType | İkili veri, dosya adı ve özet niteliklerini içeren sınıftır. |
| GetAppRespRequest | getApplicationResponse metoduna parametre olarak verilen sınıftır. |
| GetAppRespRequestType | getApplicationResponse metodundan dönen sınıftır. |
| GetAppRespResponse | Zarf id niteliğini içeren sınıftır. |
| GetAppRespResponseType | Sistem Uygulama Yanıtını içeren sınıftır. |

7.4 DocumentRequest Sınıfı Metotları

| | |
|--------------|--|
| DocumentType | getDocumentRequest () DocumentType nesnesini döner. |
| void | setDocumentRequest (DocumentType param) DocumentType nesnesine parametre ile verilen nesneyi atar. |

7.5 DocumentResponse Sınıfı Metotları

| | |
|----------------------|---|
| DocumentResponseType | getDocumentResponse() DocumentResponseType nesnesini döner. |
| Void | setDocumentResponse (DocumentResponseType param) DocumentResponseType nesnesine parametre ile verilen nesneyi atar. |

7.6 DocumentReturnType Sınıfı Metotları

| | |
|--------|--|
| String | getHash() İkili verinin özet değerini döner. |
| String | getMsg() Mesajı döner. |
| void | setHash (String param) param ile verilen değeri Hash'e atar. |
| void | setMsg (String param) param ile verilen değeri Msg'a atar. |

7.7 DocumentType Sınıfı Metotları

| | |
|--------------|---|
| Base64Binary | getBinaryData() İkili (binary) veriyi döner. |
| String | getFileName() Gönderilen zarfa ait dosya adını döner. |
| String | getHash() İkili verinin özet değerini döner. |

| | |
|------|--|
| void | setBinaryData (Base64Binary param) Bu kısma XML formatında oluşturmuş zarfın, ZIP formatı ile sıkıştırılmış halinin verisi yazılacaktır. |
| void | setFileName (String param) Bu kısma XML formatında oluşturulmuş ve daha sonra ZIP formatında sıkıştırılmış zarfın adı yazılacaktır. Zarf adının uzantısının “zip” olması gerekmektedir. Zarf adının ise e-Fatura Uygulaması Teknik Kılavuzlar “Ek-1 e-Fatura Uygulaması Zarf Şema Yapısı” dokümanında belirtilen “InstanceIdentifier” değeri ile aynı olmalıdır. |
| void | setHash (String param) Bu kısma XML formatında oluşturulmuş ve daha sonra ZIP formatı ile sıkıştırılmış zarfın MD5 özeti yazılacaktır. |

7.8 GetAppRespRequest Sınıfı Metotları

| | |
|-----------------------|--|
| GetAppRespRequestType | getGetAppRespRequest () GetAppRespRequestType nesnesini döner |
| Void | setGetAppRespRequest (GetAppRespRequestType param) GetAppRespRequestType nesnesine parametre ile verilen nesneyi atar. |

7.9 GetAppRespRequestType Sınıfı Metotları

| | |
|--------|--|
| String | getInstanceIdentifier () Zarf id'yi döner. |
| void | setInstanceIdentifier (String param) Zarf id'ye parametre ile verilen değeri atar. |

7.10 GetAppRespResponse Sınıfı Metotları

| | |
|------------------------|--|
| GetAppRespResponseType | getGetAppRespResponse () GetAppRespResponseType nesnesini döner. |
| Void | setGetAppRespResponse (GetAppRespResponseType param) GetAppRespResponseType nesnesine paramere ile verilen nesneyi atar. |

7.11 GetAppRespResponseType Sınıfı Metotları

| | |
|--------|--|
| String | getApplicationResponse () Uygulama Yanıtını döner. |
| void | setApplicationResponse (String param) Uygulama Yanıtına parametre ile verilen değeri atar. |

8 Dönülen Aykırı Durum(Exception) Kodları

Aykırı durum kodları için WSDL'de belirtilen EFaturaFaultMessage yapısı kullanılmalıdır. EFaturaFaultMessage yapısının alabileceği aykırı durum kodları ve açıklamaları aşağıdaki tabloda belirtilmiştir.

| Hata Kodu | Hata Açıklaması |
|-----------|--------------------------|
| 2000 | OZET DEGERLER ESIT DEGIL |
| 2001 | ZARF ID SISTEMDE MEVCUT |
| 2002 | ZARF ARSIVE EKLENEMEDI |
| 2003 | ZARF KUYRUGA EKLENEMEDI |
| 2004 | ZARF ID BULUNAMADI |
| 2005 | SISTEM HATASI |
| 2006 | GECERSIZ ZARF ADI |

| Hata Kodu | Hata Açıklaması |
|-----------|--|
| 2007 | PAKET GÖNDERMEYE VE SORGULAMAYA YETKİNİZ GEÇİCİ OLARAK KALDIRILMIŞTIR. |

Örnek:

```
EFaturaFaultType faultType=new EFaturaFaultType();
EFaturaFault fault=new EFaturaFault();
EFaturaFaultMessage faultMessage=new EFaturaFaultMessage();

faultType.setCode(code);
faultType.setMsg(msg);
fault.setEFaturaFault(faultType);
faultMessage.setFaultMessage(fault);

throw faultMessage;
```

Not:

WSDL hata yapısını(WSDL:fault) direk desteklemeyen programlama dilleri/araçları için de standartlaşma açısından EFaturaFaultMessage üretilmelidir.

Döndürülen SOAP mesajı yapılandırılarak EFaturaFaultMessage üretmek mümkündür. Örneğin .Net için WSDL:fault araçlar ile üretilmiyorsa SOAPException'ın metotları kullanılarak üretilbilmektedir. SoapFault detay(detail) kısmı aşağıdaki mesaja benzemelidir.

```
<soapenv:Detail>
  <ns2:EFaturaFault xmlns:ns2="http://gib.gov.tr/vedop3/eFatura">
    <code>2000</code>
    <msg>OZET DEGERLER ESIT DEGIL</msg>
  </ns2:EFaturaFault>
</soapenv:Detail>
```