

*T.C. SAĞLIK BAKANLIĐI*

*İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĐÜ*

*İLAÇ TAKİP SİSTEMİ ŞUBE MÜDÜRLÜĐÜ*

# *PAKET TRANSFER SİSTEMİ*

Kullanıcı Dökümanı

Version 1.2



T.C. SAĞLIK BAKANLIĐI

**İlaç ve Eczacılık**  
Genel Müdürlüğü



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



<b>1. Revizyonlar</b> .....	<b>1</b>
<b>2. Giriş</b> .....	<b>1</b>
2.1 Kapsam.....	1
2.2 Hedef Kitle.....	1
<b>3. Gereksinimler</b> .....	<b>1</b>
3.1 Yazılım Gereksinimleri.....	1
<b>4. Paket Gönderme ve Paket Alma Servisleri</b> .....	<b>1</b>
4.1 Paket Gönderme Servisi.....	1
4.1.1 Paket Gönderme İşlemi.....	1
4.2 Paket Alma Servisi.....	2
<b>5. Çalışma Ortamı</b> .....	<b>3</b>
5.1 Paket Alma Servisi Çalışma Ortamı.....	3
5.1.1 Genel Bakış.....	3
5.2 Paket Alma Servisi Çalışma Ortamı.....	3
5.2.1 Genel Bakış.....	3
<b>6. Örnek İstemci (Client) Kodları</b> .....	<b>3</b>
6.1 Java Programlama Dili.....	3
6.1.1 Paket Gönderme Servisi.....	3
6.1.2 Paket Alma Servisi.....	4
6.2 C# Programlama Dili.....	4
6.2.1 Paket Gönderme Servisi.....	4
6.2.2 Paket Alma Servisi.....	4
<b>7. EK 1</b> .....	<b>4</b>
<b>8. EK 2</b> .....	<b>7</b>
<b>9. EK 3</b> .....	<b>9</b>
<b>10. EK 4</b> .....	<b>10</b>



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



## 1. Revizyonlar

No	Tarih	Revizör	Değişiklikler
1.0	12.09.2011	Musa ÖĞÜTLÜ	İlk Sürüm
1.1	27.09.2011	Musa ÖĞÜTLÜ	6.2, Ek 3, Ek 4
1.2	30.10.2011	İsmail Taha AYKAÇ Mehmet Elgin AKPINAR	Yeni servis metodlarına göre güncelleme yapıldı.



## **2. Giriş**

### **2.1 Kapsam**

Paket Transfer Sistemi paydaşlar arasında gerçekleşen transferlerin sanal ortamda kolay bir şekilde takip edilmesini amaçlamaktadır. Bunun için Paket Gönderme Servisi ve Paket Alma Servisi olmak üzere Java tabanlı iki web servisi hazırlanmıştır. Bu Web Servislerin kullanılabilmesi için istemci (client) tarafının geliştirilmesi gerekmektedir.

### **2.2 Hedef Kitle**

Bu doküman paydaşların yazılım geliştirme birimlerine hitap etmektedir.

## **3. Gereksinimler**

### **3.1 Yazılım Gereksinimleri**

Paket Transfer Sistemi için geliştirilecek web servis istemcileri aşağıdaki standartlara uygun olmalıdır:

- SOAP 1.1
- Web Services Description Language (WSDL) 1.1



## 4. Paket Gönderme ve Paket Alma Servisleri

Paket Transfer Sistemi; Paket Gönderme Servisi ve Paket Alma Servisi arayüzleri için Web Servisi Tanımlama Dili (Web Service Definition Language - WSDL) kullanılır. WSDL'lerin en son sürümüne aşağıdaki linklerden ulaşabilirsiniz.

*Paket Gönderme Servisi* :<http://pts.saglik.gov.tr/PTS/PackageSenderWebService?wsdl>

*Paket Alma Servisi*:<http://pts.saglik.gov.tr/PTS/PackageReceiverWebService?wsdl>

Bu WSDL'ler beklenen mesaj yapısını, mesajı oluşturan elemanların tiplerini gösterir. Detaylı açıklama için daha önceden yayımlanmış *PAKET TRANSFER SİSTEMİ - Çalışma Prensipleri* dökümanına başvurabilirsiniz.

Paket Transfer Sistemi kapsamındaki servisleri kullanacak istemciler **HTTP Basic Authentication** yöntemi ile kimliklerini bildirmelidirler.

### 4.1 Paket Gönderme Servisi

#### 4.1.1 Paket Gönderme İşlemi

Paket gönderme işlemi gerçekleştirilmek için **sendFile** veya **sendFileStream** metotları kullanılabilir. Bu metotlar paket dosyasının gönderim yöntemi dışında bir fark içermemektedir. Detaylı bilgiler aşağıda verilmiştir.

**sendFile** metodu kaynak GLN, hedef GLN ve gönderilecek paket(dosya) parametrelerine ihtiyaç duyar.

1. Paket Zip dosyası olmalıdır ve tek seferde yalnız bir dosya gönderilebilir.
2. Paket MIME eklentisi olarak gelmelidir.
3. GLN'ler, GLN formatına uygun olmalıdır (*bkz: PAKET TRANSFER SİSTEMİ - Çalışma Prensipleri dökümanı*)
4. Servisi çağıran kullanıcı, kaynak GLN adına işlem yapabilecek yetkiye sahip değildir.
5. Paydaş kendi kendine paket gönderemez.



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



**sendFileStream** metodu diğer metotta olduğu gibi kaynak GLN, hedef GLN ve gönderilecek paket(dosya) parametrelerine ihtiyaç duyar.

1. Paket Zip dosyası olmalıdır ve tek seferde yalnız bir dosya gönderilebilir.
2. Paket Binary Stream olarak gelmelidir.
3. GLN'ler, GLN formatına uygun olmalıdır (*bkz: PAKET TRANSFER SİSTEMİ - Çalışma Prensipleri dökümanı*)
4. Servisi çağırın kullanıcı, kaynak GLN adına işlem yapabilecek yetkiye sahip değildir.
5. Paydaş kendi kendine paket gönderemez.

Her iki metot da işlem sonucunda transfer id döner. Bu id'yi kullanarak, işlem esnasında gönderilen hedef GLN adına işlem yapma yetkisine sahip olan paydaş, gönderilen paketi (Paket Alma Servisini kullanarak) alabilir.

1. Transfer id ile yalnızca bir pakete ulaşılabilir.
2. Transfer id paketi gönderen istemci tarafından kayıt altına alınmalıdır.
3. Transfer id paketin gönderildiği paydaşa bildirilmelidir.

**sendFile** ve **sendFileStream** metotlarının doğru çalışabilmesi için GLN'ler formata uygun ve geçerli (sistemde kayıtlı bir paydaşa ait) olmalıdır. Gönderilen dosya Zip formatında olmalıdır, aksi takdirde servis hata mesajı döner.

## 4.2 Paket Alma Servisi

Paket alma işlemini gerçekleştirmek için **receiveFile** veya **recieveFileStream** metotları kullanılabilir. Bu metotlar paket dosyasının alım yöntemi dışında bir fark içermemektedir. Detaylı bilgiler aşağıda verilmiştir.

**receiveFile** ve **receiveFileStream** metotları kaynak GLN ve Transfer id parametrelerine ihtiyaç duyar.

1. Kaynak GLN, GLN formatına uygun olmalıdır (*bkz: PAKET TRANSFER SİSTEMİ - Çalışma Prensipleri dökümanı*)
2. Servisi çağırın kullanıcı, kaynak GLN adına işlem yapabilecek yetkiye sahip değildir.

**receiveFile** ve **receiveFileStream** metotları sonuç olarak paket(dosya) döner.

1. Sadece bir dosya beklenmelidir.
2. Dosya **receiveFile** metodunda MIME eklentisi, **receiveFileStream** metodunda ise Binary Stream olarak gelecektir.
3. Eğer dosya kayıt altına alınacaksa, kayıt yeri ve dosya ismi istemci(client) tarafından belirlenmelidir.

**receiveFile** ve **receiveFileStream** metotlarının doğru çalışabilmesi için kaynak GLN formata uygun ve geçerli (sistemde kayıtlı bir paydaşa ait) olmalıdır. Transfer id, servisi çağırın paydaşa gönderilmiş bir pakete ait olmalıdır; aksi takdirde servis hata mesajı döner.



## 5. Çalışma Ortamı

Paket Transfer Sistemi Çalışma Ortamı kullanılarak;

- Paket Transfer Sistemi çalışma ortamına olan ağ bağlantısı kontrol edilebilecek.
- Web Servisleri için oluşturulan veri formatlarının doğruluğu kontrol edilebilecek.
- Hatalı gönderilen veri formatları için karşılaşılabilecek hata mesajları görülebilecek.

Paydaşlar için Paket Gönderme ve Paket Alma servisleri için gerçek ortam kullanıcı hesaplarını kullanacaklar.

### 5.1 Paket Alma Servisi Çalışma Ortamı

#### 5.1.1 Genel Bakış

- Formatlara uygun parametrelerin gönderilmesi durumunda gönderilen paket(dosya) kayıt altına alınır ve istemciye transfer id bilgisi dönülür.
- Herhangi bir parametre uygun olmadığı zaman paket(dosya) kayıt edilmez, hata mesajı dönülür.

### 5.2 Paket Alma Servisi Çalışma Ortamı

#### 5.2.1 Genel Bakış

- Paydaş kendisi için gönderilmiş olan paketi(dosyayı) transfer id ve GLN parametreleriyle ister.Bu transfer id ile kendisine gönderilmiş bir paket varsa servis bunu mesaj eklentisi olarak döner.
- Paydaş parametre olarak gönderdiği transfer id ile kendisine gönderilmiş bir dosya yoksa yada paydaş dosya almaya yetkili olmadığı bir GLN'yi parametre olarak göndermişse hata mesajı dönülür.

## 6. Örnek İstemci (Client) Kodları

### 6.1 Java Programlama Dili

Bu bölümde Java programlama dili ile servislere ulaşımı sağlayan örnek kodlara yer verilmiştir.

Kodlar Java SE Development Kit (JDK) 6 kütüphanesi kullanılarak geliştirilmiştir.Eklerde Paket Gönderme ve Alma Servisleri için metotlar verilmiştir.

#### 6.1.1 Paket Gönderme Servisi



sendFile metodu için örnek metot Ek 1'de mevcuttur.

### 6.1.2 Paket Alma Servisi

receiveFile metodu için örnek metot Ek 2'de mevcuttur.

## 6.2 C# Programlama Dili

Bu bölümde C# programlama dili ile servislere ulaşımı sağlayan örnek kodlara yer verilmiştir. Örnek kodlar sendFile ve receiveFile metodları için yazılmıştır bu metodlarda eklentiler mime attachment olarak gönderildiğinden 3. parti araçlar kullanmak gerekmektedir.

sendFile ve receiveFile metodlarının aksine paket dosyalarının Binary Stream olarak SOAP zarfı içerisinde gönderilmesine olanak sağlayan sendFileStream ve receiveFileStream metodları mevcuttur. **Bu metodlar kullanmak için herhangi 3. parti araç kullanmaya ihtiyaç yoktur!** Servis Web Referans olarak eklendiğinde oluşan nesnelere kolaylıkla çağırılabilir.

Ekteki kodlarda sendFile ve receiveFile metodları **AÇIK KAYNAK** kodlu **PocketSOAP** bileşeni kullanılarak çağırılmıştır.

### 6.2.1 Paket Gönderme Servisi

sendFile metodu için örnek metot Ek 3'de mevcuttur.

### 6.2.2 Paket Alma Servisi

receiveFile metodu için örnek metot Ek 4'de mevcuttur.

## 7. EK 1

```
private String sendPackage(  
    String username,  
    String pwd,  
    String filePath,  
    String frGln,  
    String toGln) throws SOAPException, MalformedURLException {  
  
    MessageFactory factory = MessageFactory.newInstance();
```





**T.C. SAĞLIK BAKANLIĞI  
İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



```
SOAPMessage message = factory.createMessage();

MimeHeaders hd = message.getMimeHeaders();
String authorization = new BASE64Encoder().encode((username + ":" + pwd).getBytes());
hd.addHeader("Authorization", "Basic " + authorization);

SOAPBody body = message.getSOAPBody();

QName sendFileParamBodyName = new QName("http://its.iegm.gov.tr/pts/sendpackage",
"sendFileParameters", "sendFile");
SOAPBodyElement sendFileParamBodyElement =
body.addBodyElement(sendFileParamBodyName);

QName frNane = new QName("sourceGLN");
SOAPElement frElement = sendFileParamBodyElement.addChildElement(frNane);
frElement.addTextNode(frGln);

QName toName = new QName("destinationGLN");
SOAPElement toElement = sendFileParamBodyElement.addChildElement(toName);
toElement.addTextNode(toGln);

SOAPConnectionFactory soapConnectionFactory = SOAPConnectionFactory.newInstance();
SOAPConnection connection = soapConnectionFactory.createConnection();

java.net.URL endpoint = new URL("http://pts.saglik.gov.tr/PTS/PackageSenderWebService");

File myfile = new File(filePath);
DataSource ds = new FileDataSource(myfile);
DataHandler dataHandler = new DataHandler(ds);

AttachmentPart attachment = message.createAttachmentPart(dataHandler);
attachment.setContentId("<filePart=12@pts.org>");

message.addAttachmentPart(attachment);

SOAPMessage soapResponse = connection.call(message, endpoint);

SOAPBody soapResponseBody = soapResponse.getSOAPBody();

if (soapResponseBody.hasFault()) {
    SOAPFault newFault = soapResponseBody.getFault();

    QName faultDetailCode = new QName("faultcode");
    java.util.Iterator itCode = newFault.getChildElements(faultDetailCode);
    SOAPBodyElement faultDetailCodeElement = (SOAPBodyElement) itCode.next();

    QName faultDetailString = new QName("faultstring");
```



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



```
java.util.Iterator itString = newFault.getChildElements(faultDetailString);
SOAPBodyElement faultDetailStringElement = (SOAPBodyElement) itString.next();

QName faultDetailBodyName = new QName("detail");
java.util.Iterator iterato = newFault.getChildElements(faultDetailBodyName);
SOAPBodyElement faultBodyElement = (SOAPBodyElement) iterato.next();

QName faultBodyName = new QName("http://its.iegm.gov.tr/pts/sendpackage",
"packageTransferError", "ns2");
java.util.Iterator iterator = faultBodyElement.getChildElements(faultBodyName);
SOAPBodyElement faultdetailElement = (SOAPBodyElement) iterator.next();

QName responseFaultCodeName = new QName("faultCode");
java.util.Iterator iterator1 = faultdetailElement.getChildElements(responseFaultCodeName);
SOAPBodyElement faultCodeElement = (SOAPBodyElement) iterator1.next();

//Fault Code
String faultCode = faultCodeElement.getValue();

QName responseFaultMessageName = new QName("faultMessage");
java.util.Iterator iterator2 =
faultdetailElement.getChildElements(responseFaultMessageName);
SOAPBodyElement faultMessageElement = (SOAPBodyElement) iterator2.next();

//Fault Message
String faultMessage = faultMessageElement.getValue();
return faultCode + " " + faultMessage ;
}

QName responseBodyName = new QName("http://its.iegm.gov.tr/pts/sendpackage",
"sendFileResponse", "ns2");
java.util.Iterator iterator = soapResponseBody.getChildElements(responseBodyName);
SOAPBodyElement responseBodyElement = (SOAPBodyElement) iterator.next();

QName responseTransferIdName = new QName("transferId");
java.util.Iterator iterator1 =
responseBodyElement.getChildElements(responseTransferIdName);
SOAPBodyElement responseTransferIdElement = (SOAPBodyElement) iterator1.next();

//Tranfer Id
String transferId = responseTransferIdElement.getValue();

connection.close();
return transferId;
}
```



## 8. EK 2

```
private String receivePackage(  
    String username,  
    String pwd,  
    String frGln,  
    String transferId,  
    String filePathWithFileName) throws SOAPException, MalformedURLException {  
  
    MessageFactory factory = MessageFactory.newInstance();  
    SOAPMessage message = factory.createMessage();  
  
    SOAPHeader header = message.getSOAPHeader();  
    MimeHeaders hd = message.getMimeHeaders();  
    String authorization = new BASE64Encoder().encode((username + ":" + pwd).getBytes());  
    hd.addHeader("Authorization", "Basic " + authorization);  
  
    SOAPBody body = message.getSOAPBody();  
  
    QName getFileParamBodyName = new  
    QName("http://its.iegm.gov.tr/pts/receivepackage", "receiveFileParameters", "receiveFile");  
    SOAPBodyElement getFileParamBodyElement =  
    body.addBodyElement(getFileParamBodyName);  
  
    QName frName = new QName("sourceGLN");  
    SOAPElement frElement = getFileParamBodyElement.addChildElement(frName);  
    frElement.addTextNode(frGln);  
  
    QName transferIdName = new QName("transferId");  
    SOAPElement transferIdElement =  
    getFileParamBodyElement.addChildElement(transferIdName);  
    transferIdElement.addTextNode(transferId);  
  
    SOAPConnectionFactory soapConnectionFactory =  
    SOAPConnectionFactory.newInstance();  
  
    SOAPConnection connection = soapConnectionFactory.createConnection();  
  
    java.net.URL endpoint = new  
    URL("http://pts.saglik.gov.tr/PTS/PackageReceiverWebService");  
  
    SOAPMessage soapResponse = connection.call(message, endpoint);  
  
    SOAPBody soapResponseBody = soapResponse.getSOAPBody();  
  
    if (soapResponseBody.hasFault()) {  
  
        SOAPFault newFault = soapResponseBody.getFault();  
        QName faultDetailBodyName = new QName("detail");
```



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



```
java.util.Iterator iterato = new Fault.getChildElements(faultDetailBodyName);
SOAPBodyElement faultBodyElement = (SOAPBodyElement) iterato.next();

QName faultBodyName = new QName("http://its.iegm.gov.tr/pts/receivepackage",
"packageTransferError", "ns2");
java.util.Iterator iterator = faultBodyElement.getChildElements(faultBodyName);
SOAPBodyElement faultdetailElement = (SOAPBodyElement) iterator.next();

QName responseFaultCodeName = new QName("faultCode");
java.util.Iterator iterator1 =
faultdetailElement.getChildElements(responseFaultCodeName);
SOAPBodyElement faultCodeElement = (SOAPBodyElement) iterator1.next();
String faultCode = faultCodeElement.getValue();

QName responseFaultMessageName = new QName("faultMessage");
java.util.Iterator iterator2 =
faultdetailElement.getChildElements(responseFaultMessageName);
SOAPBodyElement faultMessageElement = (SOAPBodyElement) iterator2.next();
String faultMessage = faultMessageElement.getValue();

return faultCode + " : " + faultMessage;
}

FileOutputStream outputFile = null;
Iterator it = soapResponse.getAttachments();
while (it.hasNext()) {
    AttachmentPart part = (AttachmentPart) it.next();
    try {

        outputFile = new FileOutputStream(filePathWithFileName);
        part.getDataHandler().writeTo(outputFile);
        return "Dosya kaydedildi: " + filePathWithFileName;
    } catch (IOException ex) {
        Logger.getLogger(ReceiverServlet.class.getName()).log(Level.SEVERE, null, ex);
        return ex.getMessage();
    } finally {
        try {
            outputFile.flush();
            outputFile.close();
            connection.close();
        } catch (IOException ex) {
            Logger.getLogger(ReceiverServlet.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
return null;
}
```



## 9. EK 3

```
private static string sendFile(string sourceGLN, string destinationGLN, string userName,
string pwd, string filePath)
{
    string result = "";
    // create the envelope
    CoEnvelope e = new CoEnvelopeClass();
    e.SetMethod("sendFileParameters", "http://its.iegm.gov.tr/pts/sendpackage");

    // create attachments manager
    CoAttachmentManager mgr = new CoAttachmentManagerClass();
    mgr.Format = AttachmentFormat.formatMime;

    // create the parameter, and point it at the attachment
    e.Parameters.Create("sourceGLN", sourceGLN, "", null, null);
    e.Parameters.Create("destinationGLN", destinationGLN, "", null, null);

    // create and configure the transport
    IHTTPTransport t = new HTTPTransportClass();
    ((HTTPTransport)t).Authentication(userName, pwd);

    CoSoapAttachment att = mgr.Request.Create(@filePath,
    TypeNameFormat.tnfAbsoluteURI, "application/zip");
    att.ContentId = "filePart=123@swa.pocketsoap.com";

    CoSoapNode n = e.Parameters.Create("filePart", null, "", null, null);
    n.href = att.Uri;

    t.SOAPAction = "";
    mgr.Transport = t;
    ISOAPTransport st = mgr as ISOAPTransport;

    st.Send("http://pts.saglik.gov.tr/PTS/PackageSenderWebService", e.Serialize());
    string enc = "";
    try
    {
        e.Parse(st, enc);
        CoSoapNode response = e.Parameters.get_Item(0);
        long transferId = long.Parse(response.Value.ToString());
        result = "Transfer Id : " + transferId;
    }
    catch (Exception ex)
    {
        try
        {
            CoSoapNode detailFault = e.Parameters.get_ItemByName("detail", "");
            CoSoapNode packageTransferError =
            detailFault.Nodes.get_ItemByName("packageTransferError", "");
            CoSoapNode faultCode =
            packageTransferError.Nodes.get_ItemByName("faultCode", "");
            CoSoapNode faultMsg =
            packageTransferError.Nodes.get_ItemByName("faultMessage", "");
        }
    }
}
```



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



```
        result = "Hata : " + faultCode.Value + " " + faultMsg.Value;
    }
    catch (Exception ex1)
    {
        result = ex.Message;
    }
}
return result;
}
```

## 10. EK 4

```
private static string getAndSaveFile(string sourceGLN, long transferId, string userName,
string pwd, string filePathWithFileName)
{
    string result = "";
    // create the envelope
    CoEnvelope e = new CoEnvelopeClass();
    e.SetMethod("receiveFileParameters",
"http://its.iegm.gov.tr/pts/receivepackage");

    // create attachments manager
    CoAttachmentManager mgr = new CoAttachmentManagerClass();
    mgr.Format = AttachmentFormat.formatMime;

    // create the parameter, and point it at the attachment
    e.Parameters.Create("sourceGLN", sourceGLN, "", null, null);
    e.Parameters.Create("transferId", transferId, "", null, null);

    // create and configure the transport
    IHTTPTransport t = new HTTPTransportClass();
    ((HTTPTransport)t).Authentication(userName, pwd);

    t.SOAPAction = "";
    mgr.Transport = t;
    ISOAPTransport st = mgr as ISOAPTransport;

    st.Send("http://pts.saglik.gov.tr/PTS/PackageReceiverWebService", e.Serialize());
    string enc = "";

    try
    {
        e.Parse(st, enc);
        // get the returned attachment and dump some info about it
        CoSoapAttachment att = mgr.Response.get_Item(1);
        if ("application/zip".Equals(att.TypeName))
        {
            att.Save(filePathWithFileName + ".zip");
        }
    }
}
```



**T.C. SAĞLIK BAKANLIĞI**  
**İLAÇ VE ECZACILIK GENEL MÜDÜRLÜĞÜ**



```
else
{
    att.Save(filePathWithFileName);
}
result = "File : " + filePathWithFileName;
}
catch (Exception ex)
{
    try
    {
        CoSoapNode detailFault = e.Parameters.get_ItemByName("detail", "");
        CoSoapNode packageTransferError =
detailFault.Nodes.get_ItemByName("packageTransferError", "");
        CoSoapNode faultCode =
packageTransferError.Nodes.get_ItemByName("faultCode", "");
        CoSoapNode faultMsg =
packageTransferError.Nodes.get_ItemByName("faultMessage", "");
        result = "Hata : " + faultCode.Value + " " + faultMsg.Value;
    }
    catch (Exception ex1)
    {
        result = ex.Message;
    }
}
return result;
}
```