



REPUBLIC OF TURKEY  
**MINISTRY OF HEALTH**  
Medicines & Medical Devices Agency

---

# PTS WORKING PRINCIPLES

September 2012

---



Turkish Medicines & Medical Devices Agency, Department of Pharmaceutical Track & Trace System  
Söğütözü Mahallesi 2176 Sok. No: 5 P.K.06520 Çankaya, Ankara  
**Phone:** (312) 218 30 00 **Help Desk:** (312) 218 34 50 **E-mail:** [its@titck.gov.tr](mailto:its@titck.gov.tr)



---

## LEGAL NOTICE

---

*All rights reserved. This document cannot be changed without written permission from Turkish Medicines & Medical Devices Agency.*

VERSION	DATE	NOTES
2.0	17.09.2012	PTS Version 2

---

## TABLE OF CONTENTS

---

<b>1.</b>	<b>OBJECTIVE, SCOPE, GENERAL OPERATION OF WEB SERVICES</b>	<b>4</b>
1.1.	Objective	4
1.2.	Scope	4
1.3.	General Operation of Web Services	4
<b>2.</b>	<b>DEVELOPMENT OF PTS</b>	<b>5</b>
<b>3.</b>	<b>PTS PACKAGE AND PTS STRUCTURE</b>	<b>6</b>
<b>4.</b>	<b>PTS WEB SERVICES</b>	<b>7</b>
4.1.	Package Sending Service	7
4.1.1.	“asFile” Operation Request Message Example and Data Fields in the Message	7
4.1.2.	“asStream” Operation Request Message Example and Data Fields in the Message	8
4.1.3.	Package Sending Service Response Message Example and Data Fields in the Message	9
4.1.4.	Fault Message Structure	10
4.2.	Package Receiving Service	11
4.2.1.	“asFile” Operation Request Message Example and Data Fields in the Message	11
4.2.2.	“asStream” Operation Request Message Example and Data Fields in the Message	12
4.2.3.	“asFile” Operation Response Message Example and Data Fields in the Message	12
4.2.4.	“asStream” Operation Response Message Example and Data Fields in the Message	13
4.2.5.	Fault Message Structure	13

## 1. OBJECTIVE, SCOPE, GENERAL OPERATION OF WEB SERVICES

### 1.1. Objective

The notifications of the Turkish Pharmaceutical Track and Trace System are made on the basis of units. However the pharmaceuticals are transferred as a whole in the sector. To ease the ITS notifications without reading the DataMatrix information of every single drug unit in the physical transfers, the “Package Transfer System” (PTS) has been developed.

This guideline has been prepared for the purpose to share the structure and usage of PTS web services with our stakeholders.

### 1.2. Scope

This document explains the communication structure, messaging structure and the data structures which will be used in the messages of the PTS web services provided by the Turkish Medicines and Medical Devices Agency. The inner processes of the Package Transfer System and of our stakeholders will not be mentioned.

Sample codes will be shared on site called github : <https://github.com/ilactakipsistemi>

### 1.3. General Operation of Web Services

The Turkish Pharmaceutical Track and Trace System web services operate in two way. The “Request” message of the client is transferred to the system, the system processes the messages and transfers the “Response” related to its request to the client. The errors which occur during the transfer of data or during the processing of the data is fed back to the client as a Fault message. The client needs to evaluate the response and fault messages, and it is assumed that it does so. The request, response and fault messages are transferred using HTTP protocol within a SOAP envelope. Clients access the web services with the HTTP Basic Authentication method. Communication can be encoded with SSL. In addition the packages can be sent and received as Mime Attachments as well as codified as Base4 Binary in the SOAP envelope.

## 2. DEVELOPMENT OF PTS

After ITS was put into use the problem of managing transfers of products occurred in the sector especially between manufacturers and pharmaceutical wholesalers. ITS operating with notifications on the basis of drug units made it obligatory to establish a relationship between transportation units (bundles, cases, pallets etc.) and the products within these units in the management of the supply chain among system stakeholders.

While the standardization studies on this issue within the sector continued, the demand that this transfer operation is to be provided by a central structure which every stakeholder could rely on and accept jointly occurred. With the conveyance of these demands to the Turkish Medicines and Medical Devices Agency (TITCK, former General Directorate of Pharmaceutical and Pharmacy) a workshop on 13.12.2010 was made by TITCK with the participation of the related stakeholders. At the end of the workshop it was shared with the stakeholders that this central structure is to be provided by TITCK; and the XML data format prepared by TITCK and planned to be used in this structure was determined with the participation of the stakeholders.

A platform in which the files created in the determined XML data format has been provided and the files which are sent through this platform are called packages. To provide the reliable transfer among stakeholders the authentication system of ITS has been used in PTS which paved the way for ITS stakeholders using PTS web services without the need for additional information.

PTS gained importance especially with the transition to Phase 2 in ITS on January 1, 2012 and PTS notifications have become mandatory in cases when they are requested by the recipient. PTS has been adopted by our stakeholders in a short span of time and started to be used.

PTS is being changed by our development team from time to time according to the requests and needs coming from our stakeholders.

### 3. PTS PACKAGE AND PTS STRUCTURE

The information of physical transfers are shared in the in the format determined in the PTS XML Standard Guideline apart from the bilateral agreements among stakeholders. To actualize this sharing through PTS web services the related XML documents must be sent as zipped files. The mentioned zipped files are called as packages. The **deflate** method should be used when the XML document within the package is zipped.

There are three important reasons for the usage of package in the zipped file web services:

- ▲ Since the SOAP envelopes used in web services are XML, there is a great possibility that an XML that is wanted to be sent as its content may damage the envelope,
- ▲ Reducing the transfers in size and thus shortening their transfer time,
- ▲ Zip files to be able to compensate the loss which can occur during transfers.

## 4. PTS WEB SERVICES

PTS notifications are independent of ITS notifications. However, PTS uses the authentication system of ITS. Thus all stakeholders can invoke PTS web services without needing additional information. There are two PTS web services; Package Sending Service and Package Receiving Service.

### 4.1. Package Sending Service

Is the web service through which a package prepared for a physical transfer can be sent to any stakeholder. There are two operations called “asFile” and “asStream” which carry out the same duty within this service. The only difference between these operations is the method of sending the packages. It is under the initiative of the stakeholder through which method the package will be sent.

Even though the request types sent in the operations are different from each other, the same response type is returned. Things that should be taken into consideration in this service are listed below:

- ▲ The package that will be sent should be a zipped file,
- ▲ If the “asFile” operation is being used the package should be sent as a MIME attachment and sending only one package in every notification,
- ▲ Being sure that the GLN of the recipient is correct,
- ▲ The stakeholder should not send packages to itself,
- ▲ Confirming that the package has reached the ITS servers without any losses by comparing the MD5 sum of the sent package and the “md5Checksum” field in the received response.

#### 4.1.1. “asFile” Operation Request Message Example and Data Fields in the Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:send="http://pts.titck.gov.tr/send">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <send:sendFileParameters>  
      <receiver>8680002800017</receiver>
```

```
</send:sendFileParameters>  
</soapenv:Body>  
</soapenv:Envelope>
```

The explanations of the data fields in the request message sent in this operation are as stated below:

#### ▲ <receiver> Field

This field contains the GLN of the recipient. For the request to be successful, the sent recipient information must exist in the system.

**Warning!** The package to be sent does not exist in the request message of the asFile operation. The reason for that is that the MIME attachment is being sent in the HTTP request. In addition only one package should be sent in every request when using this operation. Otherwise the request will not be successful.

### 4.1.2. “asStream” Operation Request Message Example and Data Fields in the Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:send="http://pts.titck.gov.tr/send">  
  <soapenv:Header />  
  <soapenv:Body>  
    <send:sendStreamParameters>  
      <parameters>  
        <receiver>8680002800017</receiver>  
      </parameters>  
      <fileStream>UESDBBQAAAAAAKSGA0F6FiIQBgAAAAAYAAAAIAAAAAdGVzdC50eHRpc21haWxQSWECEFAAU  
AAAAAACKhgNBehYpUAYAAAAGAAAAACAkAAAAAABACAAAAAAdGVzdC50eHQKACAAAAAAAE  
AGABhmusQe3HNAWGa6xB7cc0BYZrrEHtxzQFQSwUGAAAAAAEAAQBaAAAAALAAAAAA</fileStream>  
    </send:sendStreamParameters>  
  </soapenv:Body>  
</soapenv:Envelope>
```

The explanations of the data fields in the request message sent in this operation are as stated below:

#### ▲ <parameters> Field



This field is in the request type sent in the “asFile” operation. It has been used so that the data types do not differ from each other in both operations.

▲ **<receiver> Field**

This field contains the GLN of the recipient. For the request to be successful, the sent recipient information must exist in the system.

▲ **<fileStream> Field**

This field contains the package, which is wanted to be sent. Packages are coded as Binary Base64 to this field and are put into this field. The reason Base64 is used, is to prevent that the non-printable characters which the binary data could bring damage the XML envelope.

### 4.1.3. Package Sending Service Response Message Example and Data Fields in the Message

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:response xmlns:ns2="http://pts.titck.gov.tr/send">
      <transferId>379537</transferId>
      <md5Checksum>3c3944c80bda1c631dcaaa87dcaa998e</md5Checksum>
    </ns2:response>
  </S:Body>
</S:Envelope>
```

The explanations about the data types in the message are as stated below:

▲ **<transferId> Field**

transferId expresses the transfers made and is unique. The recipient stakeholder obtains the package using this value. The stakeholder who invokes the service for the recipient can see this value on its own and the recipient stakeholder can obtain the transfer id's of the packages sent to him using the related Reference Service.

▲ **<md5Checksum> Field**

This field contains the binary MD5 sum (MD5 checksum) of the sent packages with the transfer. This value is acquired from the copy of the package the stakeholder has sent in the ITS servers. This way the sender stakeholder can compare the MD5 sum of the package sent with the value received in the response, and confirm whether the package is sent to the recipient without any losses.

#### 4.1.4. Fault Message Structure

The system notifies the user with this fault message in errors which occur during the communication with web services or during the control of the message content. The clients should make the necessary warnings to the users by processing the received fault messages. The fields <faultCode> and <faultMessage> are in the detail information of the fault message.

##### ▲ <faultCode> Field

Holds the code of the fault. It is in the length of five characters. It is published in the online services logged in by our stakeholders in the Turkish Pharmaceutical Track and Trace System web site with the warning codes. In addition fault codes can be obtained using Reference Web Services.

##### ▲ <faultMessage> Field

Is the legible text corresponding to the fault code.

An example of a fault message of the package sending notification is shown below:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>Alıcı GLN formatı uygun değildir! GLN 13 haneli olmalı ve rakamlardan oluşmalıdır.</faultstring>
      <detail>
        <ns2:transferError xmlns:ns2="http://pts.titck.gov.tr/send">
          <faultCode>21002</faultCode>
          <faultMessage>Alıcı GLN formatı uygun değildir! GLN 13 haneli olmalı ve rakamlardan oluşmalıdır.</faultMessage>
        </ns2:transferError>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

```
</detail>  
</S:Fault>  
</S:Body>  
</S:Envelope>
```

## 4.2. Package Receiving Service

Is a web service through which a package of a physical transfer can be received by the recipient. There are two operations called “asFile” and “asStream” which carry out the same duty within this service. The only difference between these operations is the method of sending the packages. It is under the initiative of the stakeholder through which method the package will be sent.

Things that should be considered when using this service are as stated below:

- ▲ That the transfer id of the package sent to the recipient is correct,
- ▲ Packages should not be obtained over and over using the service,
- ▲ Confirming that the package has reached the ITS servers without any losses by comparing the MD5 sum of the sent package and the “md5Checksum” field in the received response.

### 4.2.1. “asFile” Operation Request Message Example and Data Fields in the Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:get="http://pts.titck.gov.tr/get">  
  <soapenv:Header />  
  <soapenv:Body>  
    <get:getFileParameters>  
      <transferId>379542</transferId>  
    </get:getFileParameters>  
  </soapenv:Body>  
</soapenv:Envelope>
```

The explanations of the data fields of the message are as stated below:

#### ▲ <transferId> Field

TransferId expresses the transfers made and is unique. The recipient stakeholder obtains the package using this value. The stakeholder who invokes the service for the recipient can

see this value on its own and the recipient stakeholder can obtain the transfer id's of the packages sent to him using the related Reference Service.

#### 4.2.2. "asStream" Operation Request Message Example and Data Fields in the Message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:get="http://pts.titck.gov.tr/get">  
<soapenv:Header/>  
<soapenv:Body>  
<get:getStreamParameters>  
<transferId>379542</transferId>  
</get:getStreamParameters>  
</soapenv:Body>  
</soapenv:Envelope>
```

The explanations of the data fields of the message are as stated below:

##### ▲ <transferId> Field

TransferId expresses the transfers made and is unique. The recipient stakeholder obtains the package using this value. The stakeholder who invokes the service for the recipient can see this value on its own and the recipient stakeholder can obtain the transfer id's of the packages sent to him using the related Reference Service.

#### 4.2.3. "asFile" Operation Response Message Example and Data Fields in the Message

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">  
<S:Body>  
<ns2:fileMD5checksum xmlns:ns2="http://pts.titck.gov.tr/get">  
3c3944c80bda1c631dcaa87dcaa998e</ns2:fileMD5checksum>  
</S:Body>  
</S:Envelope>
```

##### ▲ <fileMD5Checksum> Field

This field contains the MD5 sum of the package. The value in this field and the MD5 sum of the package is compared by the recipient stakeholder. This way whether the package is with or without losses is confirmed by the recipient stakeholder.

*Warning! The package of the related transfer id is attached along this response as a MIME attachment. However this package comes with the HTTP Request. Thus no information related to the package is to be found in the received SOAP envelope.*

#### 4.2.4. “asStream” Operation Response Message Example and Data Fields in the Message

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:stream xmlns:ns2="http://pts.titck.gov.tr/get">
      <streamMD5checksum>3c3944c80bda1c631dcaaa87dcaa998e</streamMD5checksum>
      <fileStream>UEsDBBQAAAAAAKSGA0F6FiIQBgAAAAAYAAAAIAAAAdGVzdC50eHRpc21haWxQSwECFAAU
      AAAAAACKhgNBehYpUAYAAAAGAAAACAkAAAAAABACAAAAAAAAAdGVzdC50eHQKACAAAAAAAE
      AGABhmusQe3HNAWGa6xB7cc0BYZrrEHTxzQFQSwUGAAAAAAEAAQBaAAAAALAAAAAAAA</fileStream>
    </ns2:stream>
  </S:Body>
</S:Envelope>
```

The explanations of the data fields of the message are as stated below:

##### ▲ <streamMD5Checksum> Field

Contains the MD5 sum of the received package. The value in this field and the MD5 sum of the package is compared by the recipient stakeholder. This way whether the package is with or without losses is confirmed by the recipient stakeholder.

##### ▲ <fileStream> Field

This field contains the package wanted to be received. The content of this field is the package coded as Binary Base64. The reason Base64 is used, is to prevent that the non-printable characters which the binary data could bring damage the XML envelope.

#### 4.2.5. Fault Message Structure

The system notifies the user with this fault message in errors which occur during the communication with web services or during the control of the message content. The clients should make the necessary warnings to the users by processing the received fault messages. The fields <faultCode> and <faultMessage> are in the detail information of the fault message.

### ▲ <faultCode> Field

Holds the code of the fault. It is in the length of five characters. It is published in the online services logged in by our stakeholders in the Turkish Pharmaceutical Track and Trace System web site with the warning codes. In addition fault codes can be obtained using Reference Web Services.

### ▲ <faultMessage> Field

Is the legible text corresponding to the fault code.

An example of a fault message of the package sending notification is shown below:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>Alınmak istenen paket 100 defa alınmaya çalışılmış. Bir paket en fazla 100 defa alınabilir, Lütfen süreçleriniz kontrol ediniz.</faultstring>
      <detail>
        <ns2:transferError xmlns:ns2="http://pts.titck.gov.tr/get">
          <faultCode>21012</faultCode>
          <faultMessage>Alınmak istenen paket 100 defa alınmaya çalışılmış. Bir paket en fazla 100 defa alınabilir, Lütfen süreçleriniz kontrol ediniz.</faultMessage>
        </ns2:transferError>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```