

# Journeys Web Service User Guide

For use with Journeys XML Web Services

Version 3.3



<http://journeystravel.co.uk>

## Copyright

© PKC Associates Limited, trading as, Journeys Travel Insurance. All rights reserved. No part of this document may be scanned, reproduced, stored in a retrieval system or transmitted in any form or by any means whether electronic, mechanical or otherwise without the prior written permission of PKC Associates Limited.

## Disclaimer

This document is for informational purposes only. PKC Associates Limited make no warranties, express or implied, through the distribution of this document. No warranty of accuracy is given concerning the contents of the information contained in this publication. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by PKC Associates Limited, its subsidiaries or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

PKC Associates Limited reserves the right to revise the content without obligation to notify any person of such changes.

## About Us

PKC Associates Ltd t/a Journeys Travel Insurance is authorised and regulated by the Financial Services Authority (Reference 312339). We are permitted to arrange with respect to non-investment insurance policies. You can check this on the FSA's Register by visiting the website <http://www.fsa.gov.uk/Pages/register> or by contacting the FSA on 0845 606 1234.

For more information, please visit our web site <http://journeystravel.co.uk>

This document is prepared and written by Joe Lee <[joel@journeystravel.co.uk](mailto:joel@journeystravel.co.uk)>



## Table of Contents

1 Introduction.....	4
2 Prerequisites.....	5
3 Connecting to JWS.....	5
3.1 Using HTTPS GET method.....	5
3.2 Using HTTPS POST method.....	6
3.2.1 PHP.....	7
3.2.2 Perl.....	8
3.2.3 Java.....	9
3.2.4 C#.....	11
4 Getting Help.....	12

# 1 Introduction

Thank you for choosing to use Journeys Web Service (JWS).

JWS is a platform for Travel Agents and their technology provider to integrate transactions, that normally are done on Journeys Extranet, on their bespoke software or web site.

JWS is designed to work with a wide range of platform using proven non-proprietary technology such as XML and SSL. Travel Agents can build a system communicate to JWS using the HTTPS protocol and exchanging messages in the XML format. HTTPS is a secure protocol used by web browsers to communicate with web sites. XML is a message format that are commonly use to exchange data between systems.

The purpose of this document is to provide you with a guide on how to communicate to JWS with your software development platform. It does not provide the transactions available on JWS (for that, please refer to a complement document “**Journeys API Specification**”).

## 2 Prerequisites

JWS is known to work on a variety of operating systems like Windows, Unix/Linux/BSD and Mac OS X. It is also known to work on the following software development platform:

- Java
- .Net
- PHP
- Perl
- Python
- Ruby

JWS will work for you if you are using one of the platform above. You will also need a permanent connection to the Internet and a static IP address.

## 3 Connecting to JWS

You can connect to JWS with the following URL:

<https://www.journeystravel.co.uk/Extranet/ws/>

JWS accepts request and response in XML message format. See “**Journeys API Specification**” for more information on the various XML messages supported by JWS.

Your requests in XML format should pass through the URL above using either the GET or POST method.

### 3.1 Using HTTPS GET method

If you wish to use the GET method, send your encoded XML request on the 'request' field as follows:

[https://journeystravel.co.uk/Extranet/ws/?request=\*your\\_encoded\\_xml\\_here\*](https://journeystravel.co.uk/Extranet/ws/?request=<i>your_encoded_xml_here</i>)

The following steps are an example of communication with JWS using the GET method:

To send the following XML message:

```
<Request service="echo">
  <String>Testing Echo...</String>
</Request>
```

You will need to encode the XML message into a string to be accepted by the query part of a URL. The encode string will look as follows:

```
%3CRequest+service%3D%22echo%22%3E%3CString%3ETesting+Echo...%3C%2FString%3E%3C%2FRequest%3E
```

Then you append the encoded string onto the URL with the `request` field as follows:

```
https://journeystravel.co.uk/Extranet/ws/?request=%3CRequest+service%3D%22echo%22%3E%3CString%3ETesting+Echo...%3C%2FString%3E%3C%2FRequest%3E
```

When you cut and paste the above URL into your browser, you will receive the following response from JWS in XML format:

```
<?xml version="1.0" encoding="utf-8" ?>
<Response class="echo">
  <String>Testing Echo...</String>
</Response>
```

## 3.2 Using HTTPS POST method

The following sub-sections contain examples of communication with JWS using various languages via the POST method.

### 3.2.1 PHP

```
<?php
function sendRequest( $xml_request ) {
    $url = "https://www.journeystravel.co.uk/Extranet/ws/";
    $ch = curl_init( $url );

    $encoded = 'request='.urlencode( $xml_request );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $encoded);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

$xml_request = <<<EOT
<?xml version="1.0" encoding="utf-8" ?>
<Request class="getKey">
    <Agent>OB999</Agent>
    <Password>Password</Password>
</Request>
EOT;

header("Content-type: text/xml");
echo sendRequest( trim($xml_request) );
```

### 3.2.2 Perl

```
sub sendRequest {  
  
    my $xml = shift;  
    my $url = 'https://www.journeystravel.co.uk/Extranet/ws/';  
  
    use HTTP::Request::Common qw(POST);  
    use LWP::UserAgent;  
  
    $ua = LWP::UserAgent->new;  
  
    my $req = POST $url,  
                [ request => $xml ];  
  
    print $ua->request($req)->as_string;  
  
}  
  
sendRequest '<?xml version="1.0" encoding="utf-8" ?>  
<Request class="getKey">  
    <Agent>OB999</Agent>  
    <Password>Password</Password>  
</Request>';
```



### 3.2.3 Java

```
...
public static String sendRequest(String xmlData)
{

    HttpURLConnection connection = null;
    String urlParams = "request=" + URLEncoder.encode(xmlData, "UTF-8");

    try {

        //Create connection
        URL url = new URL("https://www.journeystravel.co.uk/Extranet/ws/");
        connection = (HttpURLConnection)url.openConnection();
        connection.setRequestMethod("POST");
        connection.setRequestProperty("Content-Type",
            "application/x-www-form-urlencoded");
        connection.setRequestProperty("Content-Length", "" +
            Integer.toString(urlParams.getBytes().length));
        connection.setUseCaches (false);
        connection.setDoInput(true);
        connection.setDoOutput(true);

        //Send request
        DataOutputStream wr = new DataOutputStream (
            connection.getOutputStream ());
        wr.writeBytes (urlParams);
        wr.flush ();
        wr.close ();

    } catch (Exception e) {
        // continue next page...
    }
}
```

```
//Get Response
InputStream is = connection.getInputStream();
BufferedReader rd = new BufferedReader(new InputStreamReader(is));
String line;
StringBuffer response = new StringBuffer();
while((line = rd.readLine()) != null) {
    response.append(line);
    response.append('\r');
}
rd.close();
return response.toString();

} catch (Exception e) {

    e.printStackTrace();
    return null;

} finally {

    if(connection != null) {
        connection.disconnect();
    }
}
}
...

```

### 3.2.4 C#

```
...
public string sendRequest ( string xmlData )
{
    string url = "https://www.journeystravel.co.uk/Extranet/ws/";
    WebRequest request = WebRequest.Create (url);
    request.Method = "POST";

    string postData = "request=" + Uri.EscapeDataString ( xmlData );
    byte[] byteArray = Encoding.UTF8.GetBytes (postData);

    request.ContentType = "application/x-www-form-urlencoded";
    request.ContentLength = byteArray.Length;
    Stream dataStream = request.GetRequestStream ();
    dataStream.Write (byteArray, 0, byteArray.Length);
    dataStream.Close ();

    WebResponse response = request.GetResponse ();

    dataStream = response.GetResponseStream ();
    StreamReader reader = new StreamReader (dataStream);
    string responseFromServer = reader.ReadToEnd ();

    reader.Close ();
    dataStream.Close ();
    response.Close ();

    return responseFromServer;
}
// ...
```

## 4 Getting Help

If you have questions related to the Travel Insurance scheme, its terms and conditions, and anything about rates & accounts, please contact our Sales and Operations department on **0844 567 8628** or email [operations@journeystravel.co.uk](mailto:operations@journeystravel.co.uk)

For any technical help and questions related to this User Guide, please contact me:

**Joe Lee**

Email : [joel@journeystravel.co.uk](mailto:joel@journeystravel.co.uk)

Skype : [joelee.org](https://www.skype.com/people/joelee.org)

DDI : 0208 917 3338