API Developer Notes

Galileo Web Services Host Sessions

11 October 2011

Version 1.1

THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO TRAVELPORT

Copyright

Travelport provides this document for information purposes only and does not promise that the information contained in this document is accurate, current or complete. This document is subject to change without notice.. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the licensee’s personal use without the prior written permission of Travelport and/or its subsidiaries.

Trademarks

Travelport and/or its subsidiaries may have registered or unregistered patents or pending patent applications, trademarks copyright, or other intellectual property rights in respect of the subject matter of this document. The furnishing of this document does not confer any right or licence to or in respect of these patents, trademarks, copyright, or other intellectual property rights.

All other companies and product names are trademarks or registered trademarks of their respective holders.

Contents

Overview

The history of all Computer Reservation Systems (CRS) dates back to the days of large networks of dumb terminals connected to a mainframe computer, with each terminal maintaining a session to the host mainframe. The user (a travel agent) signed into the mainframe with a user ID and a password, which initiated the session, and all transactions were accomplished within the session.

Modern APIs allow many system requests without a host session. However, some requests operate on information stored within a session, such as a PNR/Booking File or a booked itinerary, and therefore must be submitted within a host session. This document describes how to initiate and work within a session.

Session Timeout Limitation

Galileo Web Services (GWS) sessions time out after three minutes of inactivity. The process flow of your application should be designed so that the application is never waiting for human input while a session is being maintained with the host. It is generally better to use a “shopping basket” approach, which collects all of the itinerary and traveler information before beginning a booking session.

Host session operations are exclusive to the XML Select Service (XWS) in the GWS API. All of the functions described within this document use XWS methods.

Starting a Session

Start a host session by calling the *BeginSession* method of the XML Select Web Services (XWS). Your Host Access Profile (HAP) is required to access XWS.

The *BeginSession* method returns a session token, which is then used for the subsequent session requests. The SOAP body for the *BeginSession* request looks similar to:

```
<soap:Body>

    <BeginSession xmlns="http://webservices.galileo.com">

        <Profile>yourHAPhere</Profile>

    </BeginSession>

</soap:Body>
```

If you have established a Web Services proxy stub by using SOAP tools for your environment, then the call to that stub usually looks similar to:

```
        yourStubName.BeginSession(yourHAPhere);
```

The preceding request returns a very long string, which is the session token. The SOAP body of the response looks similar to:

```
<soap:Body>

    <BeginSessionResponse xmlns="http://webservices.galileo.com">

        <BeginSessionResult>veryLongTokenStringHere</BeginSessionResult>

    </BeginSessionResponse>

</soap:Body>
```

Your application must save this token for use in additional session requests.

Session Requests

After initiating a host session, you can work with data that is stored in a session, such as a PNR/BF that you are creating or that you have retrieved.

It is possible to create a PNR/BF without a session by including all of the PNR/BF information in a single PNRBFManagement_# request. However, it can be simpler to open a session and build the PNR/BF incrementally, while verifying each response as the PNR/BF is built.

For other operations such as PNR/BF modification, client files, fare quotes on a booked itinerary, ticketing, and retrieving ticket numbers, a session is required. All terminal transactions must be made within a session. As an example, to retrieve a PNR/BF into a session, use the following XML request format:

```
<PNRBFManagement_#>

    <PNRBFRetrieveMods>

        <PNRAddr>

            <FileAddr/>

            <CodeCheck/>

            <RecLoc>recordLocator</RecLoc>

        </PNRAddr>

    </PNRBFRetrieveMods>
```

Retrieves the PNR/BF.

```
    <FareRedisplayMods>

        <DisplayAction>

            <Action>D</Action>

        </DisplayAction>

    </FareRedisplayMods>

</PNRBFManagement_#>
```

Optional. Retrieves fare information in the PNR/BF.

The PNRBFManagement_# request uses the *SubmitXmlOnSession* method of XWS (not the non-sessioned *SubmitXML*). The SOAP body for this request is:

```
<soap:Body>

    <SubmitXmlOnSession xmlns="http://webservices.galileo.com">

        <Token> veryLongTokenStringHere </Token>

        <Request>PNRBFManagement_# XmlRequestFormatShowAbove</Request>

        <Filter><_ xmlns=""/></Filter>

    </SubmitXmlOnSession>

</soap:Body>
```

The PNRBFManagement <PNRBFRetrieveMods> request returns the XML representation of the PNR/BF. The PNR/BF now resides in the session and can be operated on.

If you are using a proxy stub, your method call looks similar to:

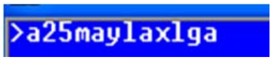
```
yourStubName.SubmitXmlOnSession ( veryLongTokenStringHere,

                                  XmlRequestFormatShowAbove,

                                  “<_ xmlns=""/>” );
```

The response is generally an XML element containing the XML representation of the PNR/BF. All requests used to operate on the PNR/BF also use the *SubmitXmlOnSession* call.

Terminal Requests

All terminal commands are also known as Terminal Emulation commands, TE queries, or Apollo/Galileo formats. These commands must be submitted within a session. A terminal command request sends the same cryptic request strings that an agent uses within the Focalpoint agency product, and returns the same screen text responses, all within simple XML tags.

Therefore, a simple air availability request for May 25 from LAX to LGA can be entered by an agent as:



```
>a25maylaxlga
```

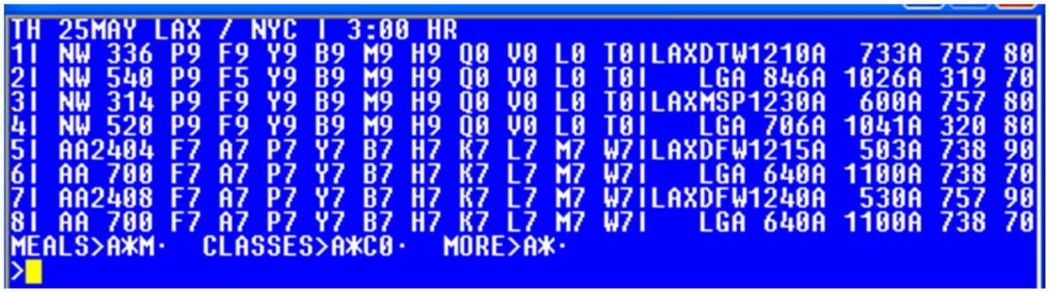
The SOAP body of the equivalent request via Web services looks similar to:

```
<soap:Body>
    <SubmitTerminalTransaction xmlns="http://webservices.galileo.com">
    <Token> veryLongTokenStringHere =</Token>
        <Request>A25MAYLAXLGA</Request>
        <IntermediateResponse/>
    </SubmitTerminalTransaction>
</soap:Body>
```

This request uses the *SubmitTerminalTransaction* method of XWS. The call to a proxy stub looks similar to:

```
         yourStubName.SubmitTerminalTransaction(“A25MAYLAXLGA”);
```

The response that an agent sees can be:



```
TH  25MAY LAX / NYC | 3:00 HR
1|  NW 336 P9 F9 Y9 B9 M9 H9 Q0 V0 L0 T0|LAXDTW1210A  733A 757 80
2|  NW 540 P9 F5 Y9 B9 M9 H9 Q0 V0 L0 T0|   LGA 846A 1026A 319 70
3|  NW 314 P9 F9 Y9 B9 M9 H9 Q0 V0 L0 T0|LAXMSP1230A  600A 757 80
4|  NW 520 P9 F9 Y9 B9 M9 H9 Q0 V0 L0 T0|   LGA 706A 1041A 320 80
5|  AA2404 F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|LAXDFW1215A  503A 738 90
6|  AA 700 F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|   LGA 640A 1100A 738 70
7|  AA2408 F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|LAXDFW1240A  530A 757 90
8|  AA 700 F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|   LGA 640A 1100A 738 70
MEALS>A*M·  CLASSES>A*C0·  MORE>A*·
>
```

Likewise, the response from Web Services is:

```
<soapenv:Body>
    <SubmitTerminalTransactionResponse xmlns="http://webservices.galileo.com">
        <SubmitTerminalTransactionResult USM="false">TH 25MAY LAX / NYC | 3:00 HR
```

```
1| NW 336 P9 F9 Y9 B9 M9 H9 Q0 V0 L0 T0|LAXDTW1230A  750A 320 802| NW 540 P9 F5 Y9 B9 M9 H9 Q0 V0 L0
T0|   LGA 850A 1031A D9S 603| DL 560 F9 A0 Y0 B0 M0 H0 Q0 K0 L0 U0|LAXATL1230A  729A 757 604| DL 506
F6 A0 Y9 B9 M9 H9 Q0 K0 L0 U0|   LGA 830A 1042A M80 705| NW 314 P9 F9 Y9 B9 M9 H9 Q0 V0 L0
T0|LAXMSP1225A  603A 757 606| NW 520 P9 F9 Y9 B9 M9 H9 Q0 V0 L0 T0|   LGA 700A 1046A 320 507| AA2408
F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|LAXDFW 100A  551A M80 908| AA 700 F7 A7 P7 Y7 B7 H7 K7 L7 M7 W7|   LGA
648A 1111A 738 70MEALS&gt;A*M;  CLASSES&gt;A*C0;  MORE&gt;A*;

----------------------------------------------------------------JT*NEW CODE JT*ESSEX HOUSE*8971*15 PCT COMM*FR
279USD    &gt;AH*1;                                                    JT*NEW CODE JT*ESSEX HOUSE*8971*15 PCT
COMM*FR 279USD    &gt;AH*2; &gt;&lt;</SubmitTerminalTransactionResult>

    </SubmitTerminalTransactionResponse>

</soapenv:Body>
```

The response is returned just as an agent would see it, with 64 characters to a line. Note that some characters are XML entity references, such as *&gt* for the > character.

Ending a Session

Make sure you end the session, also known as End Transact or ET, when all the session transactions are complete. Ending the session releases the GTID that was reserved when the session was started.

If modifications were made to a PNR/BF during the session, be sure to send an *End Transact* request before ending the session. To end a session, use the *EndSession* method of XWS with the session token. The SOAP body for this method is:

```
<soap:Body>
    <EndSession xmlns="http://webservices.galileo.com">
        <Token> veryLongTokenStringHere </Token>
    </EndSession>
</soap:Body>
```

The response is:

```
<soap:Body>
    <EndSessionResponse xmlns="http://webservices.galileo.com"/>
</soap:Body>
```

Summary

Host sessions are an important feature in Galileo Web Services, especially for PNR/BF modification and terminal transactions. Sessions can be very effective, as long they are not kept waiting for human responses. The proper use of host sessions can aid in the development of robust travel applications.