

Remote Availability Update Service (RAUS) Guide

Contents

Description	2
How to use RAUS	2
Add any current bookings	2
Add new bookings when added in your system	3
Cancellations	4
List all homes	5
Responses	6
Appendix	7
Code examples for Update Availability Calendar	7
C#	7
Java	8
Php	9
Code examples for List Homes	10
C#	10
Java	11
Php	12

Description

The Remote Availability Update Service enables you to maintain your property's Availability Calendar without having to log into the HL website.

Simply put, as soon as you have added the functionality to your systems, the Holiday Lettings availability calendar for your advert is synchronised, in real-time, with your own website. The advantages to you are primarily twofold:

- 1) You don't need to spend time updating two websites, leaving you free to get on with your business.
- 2) Regular updates of the availability calendar helps your adverts appear higher in holidaymakers search results, which hopefully should result in more bookings for you

How to use RAUS

As soon as the adverts are created, the availability calendars will need to be set and updated. Find here the different steps to maintain your availability calendars.

The web page link (URL) similar to the one below is used for all the different steps, but using different values for the parameters highlighted in red:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=StartDate&end=EndDate&status=Status&owner_id=OwnerID&home_id=HomeID&secret=SecretKey

Add any current bookings

Now that your availability calendar is set to all green, you can add your current bookings that you have in your system.

You will need to provide us with the following information, which should be sent to us by substituting the bold red text in the URL as follows:

- **StartDate**: start date of a current booking, for example **2012-05-14**
- **EndDate**: the date of the last full day of the booking (i.e. Departure date minus 1), for example **2012-05-21**
- **Status**: in this case, the status will be **b**, which means "Booked".
- **OwnerID**: your advertiser ID
- **HomeID**: your home ID of the home you want to add the current booking.
- **SecretKey**: your Secret Key for authentication, as provided by your Account Manager. This key is case sensitive.

Example:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=2012-05-14&end=2013-05-21&status=b&owner_id=48415&home_id=12485&secret=SDFGDFGDG45DFG44DTG5DF89IFFG8F

Your availability calendar will look like this now:

	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M					
Apr 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
May 12		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Jun 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Jul 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Aug 12			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Sep 12				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Oct 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Nov 12				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Dec 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Jan 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Feb 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28				
Mar 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Apr 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
May 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Jun 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Jul 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Aug 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Sep 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
Oct 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Nov 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Dec 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Jan 14		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Feb 14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28				
Mar 14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

Repeat this step for all your current bookings.

Add new bookings when added in your system

When a booking is added in your system, you will need to update your availability calendar.

This is the same process as described on the previous step (i.e. [Add any current bookings](#)).

You will need to provide us with the following information, which should be sent to us by substituting the bold red text in the URL as follows:

- **StartDate**: start date of the new booking, for example **2013-09-14**
- **EndDate**: the date of the last full day of the booking (i.e. Departure date minus 1), for example **2013-09-21**
- **Status**: in this case, the status will be **b**, which means "Booked".
- **OwnerID**: your advertiser ID
- **HomeID**: your home ID of the home you want to add a new booking.
- **SecretKey**: your Secret Key for authentication, as provided by your Account Manager. This key is case sensitive.

Example:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=2013-09-14&end=2013-09-21&status=b&owner_id=48415&home_id=12485&secret=SDFGDFGDG45DFG44DTG5DF89IFFG8F

You availability calendar will look like this now:

	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M						
Apr 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
May 12			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Jun 12						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Jul 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
Aug 12						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Sep 12							1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Oct 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
Nov 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Dec 12						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Jan 13			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Feb 13						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28			
Mar 13						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Apr 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
May 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Jun 13						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Jul 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
Aug 13					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Sep 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
Oct 13			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Nov 13						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Dec 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
Jan 14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
Feb 14						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28			
Mar 14						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Cancellations

When you receive a cancellation, you will need to remove the booking from your availability calendar. To do it, set the cancelled booking dates as available again.

You will need to provide us with the following information, which should be sent to us by substituting the bold red text in the URL as follows:

- **StartDate**: start date of the cancelled booking, for example **2013-09-14**
- **EndDate**: end date of the cancelled booking, for example **2013-09-21**
- **Status**: in this case, the status will be **a**, which means "Available".
- **OwnerID**: your advertiser ID
- **HomeID**: your home ID of the home you want to cancel a booking.
- **SecretKey**: your Secret Key for authentication, as provided by your Account Manager. This key is case sensitive.

Example:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=2013-09-14&end=2013-09-21&status=a&owner_id=48415&home_id=12485&secret=SDFGDFGDG45DFG44DTG5DF89IFFG8F

You availability calendar will look like this now:

	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M												
Apr 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
May 12		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Jun 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Jul 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Aug 12				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Sep 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Oct 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Nov 12				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Dec 12					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Jan 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Feb 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28				
Mar 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Apr 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
May 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Jun 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Jul 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Aug 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Sep 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
Oct 13		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Nov 13				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Dec 13	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Jan 14		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Feb 14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28				
Mar 14				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	

List all homes

If you need a list of the homes that you have on your account, you can call this web link:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=OwnerID&secret=SecretKey

With

- **OwnerID**: your advertiser ID
- **SecretKey**: your Secret Key for authentication, as provided by your Account Manager. This key is case sensitive.

Example:

https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=48415&secret=SDFGDFGDG45DFG44DTG5DF89IFFG8F

You will get an XML document back with Home IDs, Home Names and Active States, which will look like this:

```
<homes>
  <home>
    <home_id>197817</home_id>
    <hom_homename>Family holiday - 625</hom_homename>
    <active_yn>0</active_yn>
  </home>
  [...]
</homes>
```

The different fields returned are:

- **home_id**: Home ID of the advert.
- **hom_homname**: Home Name of the advert.
- **active_yn**: Active State of the advert, it can be equal to 0 (inactive or deleted) or 1 (active).

Responses

If your update is successful, the web page will return the word “OK”, otherwise it will return an explicit error messages explaining what the error is.

Examples:

- Start Date or End Date in incorrect format
- Secret Key '2L9HOI3ZS4SDFSDFGDFGENUZ8BWS' not matched for Advertiser ID 97454
- Owner 9874 not found
- Status is not a, b or r
- ...

Please check all of the details that you have provided, including:

- Start Date and End Date, which must be in yyyy-mm-dd format (e.g. 2009-02-29)
- Availability Status, which must be either a, b, u or r
- Your Owner ID and Home ID, which should be numeric
- Your Secret Key, which must be 30 characters in length

In the event that you can't find out where the issue is, please contact HL for assistance.

Appendix

Code examples for Update Availability Calendar

C#

```
///<summary>
/// In this example method, the trigger is running the Main method of the Console Application
///
/// Note we are using web page link (URL) similar to the one below:
///
/// https://agentapi.holidaylettings.co.uk/service_avail.aspx?
act=update&start=StartDate&end=EndDate&status=Status&owner_id=OwnerID&home_id=HomeID&secret=SecretKey
///</summary>
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;

namespace HLWebServices {
    public class Program {
        // Will contain all the availabilities that you want to set up
        Dictionary<string,string> availList = new Dictionary<string,string>();

        /// <summary>
        /// Update your Availability Calendar using the URL specified above
        /// </summary>
        public void setAvailabilityCalendar() {
            string startDate = "", endDate = "";
            string status = "u";
            // Counters for summary display
            int _numAvailsAttempted = 0;
            int _numAvailsSucceeded = 0;
            int _numAvailsFailed = 0;
            int _numAvailsInvalid = 0;

            // Add dates and status in your array
            availList.Add("2012-03-03,2012-03-10", "b");
            availList.Add("2012-03-10,2012-03-17", "b");
            availList.Add("2012-04-14,2012-04-28", "b");

            Console.WriteLine("[STARTING Availability Calendar Set Up] Starting processings
availabilities.");

            try {
                string ownerId = "98940";
                string key = "2L9HOI3ZS4UBY92SEDXX5ENUZ8BWSO";
                string homeId = "164677";

                // For each row in your array
                foreach (KeyValuePair<string,string> avail in availList) {
                    _numAvailsAttempted++;

                    // avail.key contains start date and end date, separated by a comma
                    string[] availTrim = avail.Key.Split(',');
                    startDate = availTrim[0];
                    endDate = availTrim[1];
                    status = avail.Value;

                    // Calling URL to update availability calendar (RAUS)
                    HttpWebRequest request =
(HttpWebRequest)WebRequest.Create("https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start
=" + startDate + "&end=" + endDate + "&status=" + status + "&owner_id=" + ownerId + "&home_id=" + homeId +
"&secret=" + key);

                    // You need to add a User Agent when sending your request
                    request.UserAgent = "sapphire";
                    using (HttpWebResponse response =
(HttpWebResponse)request.GetResponse()) {
                        StreamReader reader = new
StreamReader(response.GetResponseStream());

                        string urlText = reader.ReadToEnd();
                        Console.WriteLine("Attempting setting up Available Calendar
from " + startDate + " to " + endDate + " - RESPONSE: " + urlText);
```

```

        // Check what the response is returned to display the results at the end
        switch (urlText) {
            case "OK": _numAvailsSucceeded++;
                        break;
            case "FAILED": _numAvailsFailed++;
                        break;
            case "INVALID": _numAvailsInvalid++;
                        break;
        }
    }
}
} catch (WebException oWEx) {
    WebExceptionStatus oStatus = oWEx.Status;
    Console.WriteLine("ERROR[SetAvailCalendar] - " + oWEx.Message + " STATUS: "
+ oStatus + "\r\n");

    Console.WriteLine("\r\nPress a key to exit.");
    Console.ReadLine();
}

// Display total availabilities processed, successful, failed and invalid
Console.WriteLine(string.Format("[RESULTS]Numbers succeeded = {0}; failed = {1};
invalid = {2}; Total attempted = {3}", _numAvailsSucceeded, _numAvailsFailed, _numAvailsInvalid,
_numAvailsAttempted));

Console.WriteLine("\r\nPress a key to exit.");
Console.ReadLine();
}

static void Main(string[] args) {
    setAvailabilityCalendar();
}
}
}

```

Java

```

/**
 * In this example method, the trigger is running the Main method of the Console Application
 *
 * Note we are using web page link (URL) similar to the one below:
 * https://agentapi.holidaylettings.co.uk/service_avail.aspx?
act=update&start=StartDate&end=EndDate&status=Status&owner_id=OwnerID&home_id=HomeID&secret=SecretKey
 */
package WebServicesPackage;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;

public class Raus {
    /**
     * Will contain all the availabilities that you want to set up*/
    Map<String,String> availList = new HashMap<String,String>();

    /**
     * Update your Availability Calendar using the URL specified above
     */
    public void setAvailabilityCalendar() {
        String startDate = "", endDate = "";
        String status = "u";
        /*Counters for summary display*/
        int _numAvailsAttempted = 0;
        int _numAvailsSucceeded = 0;
        int _numAvailsFailed = 0;
        int _numAvailsInvalid = 0;

        /*Add dates and status in your array*/
        availList.put("2012-03-03,2012-03-10", "b");
        availList.put("2012-02-10,2012-02-17", "b");
        availList.put("2012-10-14,2012-10-28", "b");

        System.out.println("[STARTING Availability Calendar Set Up] Starting processings
availabilities.");

        String ownerId = "98940";
        String key = "2L9HOI3ZS4UB92SEDXX5ENUZ8BWS0";
        String homeId = "164677";
        try {
            /*For each row in your array*/
            for (Map.Entry<String,String> avail : availList.entrySet()) {
                _numAvailsAttempted++;
            }
        }
    }
}

```



```

comma*/
/*avail.getKey() contains start date and end date, separated by a
String[] availTrim = avail.getKey().split(",");
startDate = availTrim[0];
endDate = availTrim[1];
status = avail.getValue();

/*Calling URL to update availability calendar (RAUS)*/
URL iurl = new
URL("https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=" + startDate + "&end=" +
endDate + "&status=" + status + "&owner_id=" + ownerId + "&home_id=" + homeId + "&secret=" + key);
URLConnection uc;
uc = (URLConnection)iurl.openConnection();
uc.connect();

/*Get the response from the request*/
BufferedReader in = new BufferedReader(new
InputStreamReader(uc.getInputStream()));
String urlText = in.readLine();
in.close();

System.out.println("Attempting setting up Available Calendar from "
+ startDate + " to " + endDate + " - RESPONSE: " + urlText);

/*Check what the response is returned to display the results at the end*/
if (urlText.contains("OK"))
    _numAvailsSucceeded++;
if (urlText.contains("FAILED"))
    _numAvailsFailed++;
if (urlText.contains("INVALID"))
    _numAvailsInvalid++;
}
} catch (IOException e) {
    e.printStackTrace();
    System.out.println("ERROR[SetAvailCalendar] - MESSAGE: " + e.getMessage() +
" CAUSE: " + e.getCause());
}
/*Display total availabilities processed, successful, failed and invalid*/
System.out.println("[RESULTS]Numbers succeeded = " + _numAvailsSucceeded + "; failed = " +
_numAvailsFailed + "; invalid = " + _numAvailsInvalid + "; Total attempted = " + _numAvailsAttempted);
}

/**
 * Main method, display result and error messages
 */
public static void main(String[] args) throws RemoteException {
    setAvailabilityCalendar();
}
}

```

Php

```

<html>
<head>
<title></title>
</head>
<body>
<h1>Welcome to Holiday Letting Set up Availabilities (RAUS) page.</h1>
<p>
    We are calling using web page link (URL) similar to the one below to update the availability
    calendars.<br/>
    https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=StartDate&end=EndDate&status=Stat
    us&owner_id=OwnerID&home_id=HomeID&secret=SecretKey
</p>
<?php
    $ownerId = "98940";
    $key = "2L9H0I3ZS4UBY92SEDXX5ENUZ8BWSO";
    $homeId = "164677";

    // Counters for summary display
    $numAvailsAttempted = 0;
    $numAvailsSucceeded = 0;
    $numAvailsFailed = 0;
    $numAvailsInvalid = 0;

    echo "<p>[STARTING Availability Calendar Set Up] Starting processing availabilities</p>";

```

```

// Arrays containing all the availabilities that you want to set up
$startDatesArray = array('2012-03-03','2012-03-10','2012-04-14');
$endDatesArray = array('2012-03-10','2012-03-17','2012-04-28');
$statusArray = array('b','b','b');

// For each row in your arrays
foreach ($startDatesArray as $i => $value) {
    $startDate = $startDatesArray[$i];
    $endDate = $endDatesArray[$i];
    $status = $statusArray[$i];

    // Calling URL to update availability calendar (RAUS)
    $requestURL = "https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=update&start=" .
    $startDate . "&end=" . $endDate . "&status=" . $status . "&owner_id=" . $ownerId . "&home_id=" . $homeId .
    "&secret=" . $key;
    $urlText = file_get_contents($requestURL);

    echo "Attempting setting up Available Calendar from ". $startDate . " to ". $endDate . " -
    RESPONSE: " . $urlText . "<br>";

    // Check what the response is returned to display the results at the end
    if ($urlText == "OK")
        $numAvailsSucceeded++;
    if ($urlText == "FAILED")
        $numAvailsFailed++;
    if ($urlText == "INVALID")
        $numAvailsInvalid++;
}
// Display total availabilities processed, successfull, failed and invalid
echo "[RESULTS]Numbers succeeded = ". $numAvailsSucceeded ."; failed = ". $numAvailsFailed .";
invalid = ". $numAvailsInvalid ."; Total attempted = ". $numAvailsAttempted;

// Script finished to runs
echo "<p>Script complete</p>";
?>
</body>
</html>

```

Code examples for List Homes

C#

```

///<summary>
/// In this example method, the trigger is running the Main method of the Console Application
///
/// Note we are using web page link (URL) similar to the one below:
///
/// https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=OwnerID&secret=SecretKey
///</summary>
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;

namespace HLWebServices {
    public class Program {
        /// <summary>
        /// Get list of all homes for a given owner Id
        /// </summary>
        public void getListHomes() {

            Console.WriteLine("[STARTING List Homes] Starting fetching list of homes.");
            try {
                string ownerId = "98940";
                string key = "2L9H0I3ZS4UB92SEDXX5ENUZ8BWSO";

                // Calling URL to fetch homes (RAUS)
                HttpWebRequest request =
                (HttpWebRequest)WebRequest.Create("https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_i
                d=" + ownerId + "&secret=" + key);
            }
        }
    }
}

```

```

        // You need to add a User Agent when sending your request
        request.UserAgent = "sapphire";
        using (HttpWebResponse response = (HttpWebResponse)request.GetResponse()) {
            StreamReader reader = new StreamReader(response.GetResponseStream());
            string urlText = reader.ReadToEnd();
            Console.WriteLine("Attempting fetching list of homes for owner Id " +
ownerId + " - RESPONSE: " + urlText);
            Console.WriteLine("\r\nPress a key to exit.");
            Console.ReadLine();
        }
    } catch (WebException oWEx) {
        WebExceptionStatus oStatus = oWEx.Status;
        Console.WriteLine("ERROR[getListHomes] - " + oWEx.Message + " STATUS: " +
oStatus + "\r\n");

        // Save the result in a txt file
        TextWriter temp_tw = new
            StreamWriter("C:\\Documentation\\WebServices\\ListHomeRaus.txt");
        temp_tw.WriteLine(urlText);
        temp_tw.Flush();
        temp_tw.Close();
        Console.WriteLine("\r\nPress a key to exit.");
        Console.ReadLine();
    }
}

static void Main(string[] args) {
    getListHomes();
}
}
}

```

Java

```

/**
 * In this example method, the trigger is running the Main method of the Console Application
 *
 * Note we are using web page link (URL) similar to the one below:
 * https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=OwnerID&secret=SecretKey
 */
package WebServicesPackage;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.HttpURLConnection;
import java.net.URL;

public class Raus {
    /**
     * Get list of all homes for a given owner Id
     */
    public void getListHomes() {
        System.out.println("[STARTING List Homes] Starting fetching list of homes.");
        String ownerId = "98940";
        String key = "2L9H0I3ZS4UB92SEDXX5ENUZ8BWS0";
        try {
            /*Calling URL to fetch homes (RAUS)*/
            URL iurl = new
URL("https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=" + ownerId + "&secret=" +
key);

            HttpURLConnection uc;
            uc = (HttpURLConnection)iurl.openConnection();
            uc.connect();

            /*Get the response from the request*/
            BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInputStream()));

            String line = null;
            String urlText = "";
            /*For each line in the bufferedreader*/
            while ((line = in.readLine()) != null) {
                urlText += line + "\r\n";
            }
            in.close();

            System.out.println("Attempting fetching list of homes for owner Id " + ownerId +
"\r\n");

            System.out.println(urlText);

```

```

        /*Save the result in a txt file*/
        PrintWriter out = new PrintWriter("C:\\Documentation\\WebServices\\ListHomeRaus.txt");
        out.println(urlText);
        out.close();
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("ERROR[getListHomes] - MESSAGE: " + e.getMessage() + " CAUSE: " +
e.getCause());
    }
}

public static void main(String[] args) throws RemoteException {
    getListHomes();
}
}

```

Php

```

<html>
    <head>
        <title></title>
    </head>
    <body>
        <h1>Welcome to Holiday Letting Get List Homes (RAUS) page.</h1>
        <p>
            We are calling using web page link (URL) similar to the one below to get a list of all homes for a
            given owner Id.<br/>
            https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=OwnerID&secret=SecretKey
        </p>
        <?php
            $ownerId = "98940";
            $key = "2L9HOI3ZS4UB92SEDXX5ENUZ8BWSO";

            echo "<p>[STARTING List Homes] Starting fetching list of homes.</p>";

            // Calling URL to fetch homes (RAUS)
            $requestURL = "https://agentapi.holidaylettings.co.uk/service_avail.aspx?act=homelist&owner_id=" .
$ownerId . "&secret=" . $key;
            $xml = simplexml_load_file($requestURL);

            echo "Attempting fetching list of homes for owner Id ". $ownerId . "<br>";

            // If the response is successful, rebuild the XML file
            if ($xml != null){
                $resultString = "&lt;?xml version='1.0' encoding='utf-8'&gt;<br>";
                $resultString = $resultString."&lt;homes&gt;<br>";
                foreach ($xml->home as $Item) {
                    $resultString = $resultString."&lt;home&gt;<br>";
                    $resultString = $resultString."&lt;home_id&gt;". $Item->home_id . "&lt;/home_id&gt;<br>";
                    $resultString = $resultString."&lt;hom_homename&gt;". $Item->hom_homename
                    . "&lt;/hom_homename&gt;<br>";
                    $resultString = $resultString."&lt;active_yn&gt;". $Item->active_yn . "&lt;/active_yn&gt;<br>";
                    $resultString = $resultString."&lt;/home&gt;<br>";
                }
                $resultString = $resultString."&lt;/homes&gt;";
            }
            echo $resultString;

            // Save the output in a file
            file_put_contents("ListHomeRaus.xml",$resultString);

            // Script finished to runs
            echo "<p>Script complete</p>";
        ?>
    </body>
</html>

```