AdFalcon Mobile Web API Integration Developer's Guide

AdFalcon Mobile Ad Network

Product of

Noqoush Mobile Media Group

Table Of Contents

1 Overview

This document aims at providing developers with instructions and guidance on how to access AdFalcon API from within their Mobile Websites.

Through using JavaScript, the recommended integration mechanism for mobile web sites, which sends Ad Requests with user and device parameters to the AdFalcon network, our system will be able to detect your web site, the device its running on and its location as well as other useful information that will allow AdFalcon to deliver a highly targeted ad for users of your mobile web site.

2 Ad Request Parameters

Mandatory parameters

There are a collection of parameters that are mandatory in all ad requests. If you do not fill the required parameters in a correct way, AdFalcon network will not be able to respond back with an appropriate ad.

For Mobile Web Sites, the mandatory parameters are:

- **Site ID (R_SID)**
  A string conataining a unique site ID of your application or web site.

Optional Parameters

The optional parameters are extra information related to the user and device which helps AdFalcon's ads selector engine to get the most relevant ads for the user.

- **Test Mode (R_TM)**
  A boolean intended to inform AdFalcon network about application or web is under the development or maintenance mode. Ex. R_TM: true

  Note: *Ensure Test Mode Parameter is set to false before releasing your application to the public.*

- **Ad Unit Size (R_AS)**
  An integer identifies the ad unit size to be delivered by AdFalcon.

  Below are the supported sizes:

Name	Size	Value
Standard	320x48	1
X-Large Banner	300x50	2
Large Banner	216x36	3
Medium Banner	168x28	4
Small Banner	120x20	5
Full Banner	480x60	6
Leaderboard	728x90	7
Medium Rectangle	300x250	8
Skyscraper	120x600	9

- **User ID (R_UUID)**
  A string used to get the unique ID for the user. You can get this value in two ways:

  1. If you have your own unique ID for each user you can always pass this ID.
  2. If you do not have a unique ID for each user then you can do the following:
     A – Send the first ad request without R_UUID and D_UDID parameters.

B – When AdFalcon receives the request without a unique ID, the system will generate a new unique user id and send it back as follows:
- Set the HTTP header "X-ADFALCON-UUID"
- Set a persistent cookie
- Set UUID field in JSON responses.

C – Get the generated user unique id from X-ADFALCON-UUID HTTP header or from the UUID JSON field.
D – Thereafter, pass the user unique id in the R_UUID in all ad requests that are submitted on behalf of the same user.

- **Language (U_LN)**
  A two chars containing the language of the requested ad in ISO 639-1 language codes format (Two Letters code); for arabic language, pass **ar** code for language parameter U_LN: ar.

- **Age or birthdate (U_AGE or U_BD)**
  You can add one of these two parameters to determine the age of the user. In the case you pass a birthdate you must send it in the dd.MM.yyyy format

- **Gender (U_G)**
  A char containing the gender of the user

  The possible values are:

Gender	Value
Male	m
Female	f

- **Keywords (U_KW)**
  A string is used to list a collection of keywords in comma separated format. These keywords can be useful since AdFalcon's ads selector engine will search for ads containing these keywords.

- **Country Code (U_CC)**
  A string containing county code of the end user in ISO 3166-1 alpha-2 format code (two-letter code).

- **Area Code (U_AC)**
  A string containing the user's area code.

- **Postal Code (U_PC)**
  A string containing the user's postal/ZIP code.

- **Geolocation (D_LA, D_LO)**
  A double containing the geolocation information of the device. The location information is divided into two values; latitude (D_LA) and longitude (D_LO).
  Ex. D_LA: 35.54454548, D_LO: 58.981

3 Client-Side Web Integration - JavaScript

AdFalcon offers Javascript solution for smartphone websites. If you are sure that your site will only be viewed over smartphone devices, you can use the JavaScript integration solution. Otherwise we recommend integrating with AdFalcon network using the server side web integration method explain in the sections below.

Contents

Function	Description
adFalcon(**key, parameters, enableAlertFeedback**) **key** A string containing unique key for an instance in whole page. **parameters** An object containing AdFalcon required and optional parameters **enableAlertFeedback** A boolean enables alert messaging for the AdFalcon server's feedback	this function containing a set of member variables and functions these members used to load and render the ads. you do not use these members directly; you just create an instance of AdFalcon function, then use the coming member function to load and render the ads.
loadAd()	this function is a member function for AdFalcon that is used to load and render the ads.

How to use

To use AdFalcon JavaScript you will need to create three simple script sections as follows:

- The first script defining a variable object "adFalconParameters " that holds the required and optional parameters. The variable name can be named after your naming conventions.
- The second script is responsible for loading AdFalcon JavaScript file from this path: http://www.adfalcon.com/static/js/adfalcon-web-all-1.0.0.js

Note: *We highly recommend setting the previous two scripts in the header of HTML page.*

- Third script is responsible for:
  - Creating a new instance of **adFalcon(key, parameters, enableAlertFeedback).**
  - Loading and rendering the ad using **loadAd()** member method.

Code

```
<html>
<header>

<!-- this section declares the parameters for adFalcon JavaScript, you find these parameters in AdFalcon Mobile Web API document in Appendix A -->

<script type="text/javascript">
var adFalconParameters =
{
   R_SID: "set your Site ID"
```

```
    , R_TM: true
    //Extra optional parameters
    //, R_AS:1 //If particular Ad size is requested; otherwise ignore it.
    //, R_UUID:"UUID" // If you maintain a unique user id for each user, then provide it here.
    //, U_LN: "ar"
    //,U_AGE: 27
    //,U_G: m
    //,U_KW: "sport, music"
    //,U_CC: "JO"
    //,U_AC: "962"
    //,U_PC: "11121"
    //,D_LA:35.654
    //,D_LO:34.054


}
</script>

<!--this section loads JavaScript of AdFalcon -->

<script type="text/javascript" src="http://www.adfalcon.com/static/js/adfalcon-web-all-1.0.0.js">
</script>

</header>
<body>

<!--Creating new instance of adFalcon function and load the ad -->

<script type="text/javascript">
    var adFalconInstance1 = new adFalcon("set unique key for this instance", adFalconParameters, false);
    adFalconInstance1.loadAd();
</script>

</body>
</html>
```

JQuery Mobile/JQTouch Frameworks

AdFalcon JavaScript components supports JQuery Mobile and JQTouch Frameworks, in addition to any similar frameworks which implement multiple pages whiten one HTML/JS file.

In order to load a new ad in each of the pages implemented in such frameworks, you will need to call "adFalcon.loadAd" method whenever a page is shown by registering with the "pageshow" event.

Ex. for Mobile jQuery

```
<script type="text/javascript">

    var adFalconInstance1 = new adFalcon("page1_id");

        $("body").delegate("#page1", "pageshow", function(){

                adFalconInstance1.loadAd();

        });

</script>
```

4 Server-Side Web Integration

AdFalcon offers web site integration code for the most well-known programing languages to integrate with AdFalcon. You simply need to paste the code below where you want the ad to appear in the page, and follow the instructions provided in the comments.

Mandatory Parameters

There are a collection of parameters that are mandatory in all ad requests. If you do not fill the required parameters in a correct way, AdFalcon network will not be able to respond back with an appropriate ad.

- **Site ID (R_SID)**
  This parameter is the unique ID for your application or web site in AdFalcon system.

- **Ad Format (R_F)**
  AdFalcon supports the following types of ad formats related to mobile web sites integration which are

Format	Description
BASIC_HTML	Simple HTML without JavaScript
XHTML	XHTML 1.1 with JavaScript format.

- **IP (R_IP)**
  This parameter is the IP Address of the mobile devices to which the Ad will be delivered. The IP Address helps AdFalcon to identify the country and carrier of the device.

  You can get the IP Address of the device making the request from the HTTP header **REMOTE_ADDR**.

  If you have an intermediate server (proxy) between the device and your server you can get the device IP address from HTTP header **X_FORWARDED_FOR**.

- **User agent (D_UA)**
  The User agent of the device making the request. This helps AdFalcon to identify the device's specifications and capabilities.

  You can get the user-agent of the device making the request from HTTP header **USER-AGENT**.

- **Version (R_V)**
  The version of the AdFalcon Server API on which the integration has been implemented. The current version is: **api-all-1.0.0**

- **User ID (R_UUID)**
  This parameter is used to get the unique ID for the device user. you can get this value in two ways:

  3. If you have your own unique ID for each user you can always pass this ID.
  4. If you do not have a unique ID for each user then you can do the following:
     A – Send the first ad request without R_UUID and D_UDID parameters.
     B – When AdFalcon receive the request without a unique ID, the system will generate a new unique user id and send it back as follows:
     - Set the HTTP header "X-ADFALCON-UUID"
     - Set a persistent cookie
     - Set UUID field in JSON responses.

     C – Get the generated user unique id from X-ADFALCON-UUID HTTP header or from the UUID JSON field.
     D – Thereafter, pass the user unique id in the R_UUID in all ad requests that are submitted on behalf of the same user.

- **HTTP Headers of end-user browser (R_HH_HEADERNAME)**
  HTTP headers sent by the user's device. In order to pass header values use the following format: R_HH_HEADERNAME = header value. Header name should be in capital letters only.

  Example

  R_HH_REMOTE-ADDR=94.249.59.107&R_HH_USER-AGENT=keep-alive…etc

Optional Parameters

The optional parameters are extra information related to the user and device which helps AdFalcon's ads selector engine to get the most relevant ads for the user.

The most important optional parameters are:

- **Test Mode (R_TM)**
  A boolean intended to inform AdFalcon network about application or web is under the development or maintenance mode. Ex. R_TM: true

  Note: *Ensure Test Mode Parameter is set to false before releasing your application to the public.*

- **Ad Unit size (R_AS)**
  This parameter identifies the ad unit size to be delivered by AdFalcon. Below are the supported sizes.

Ad size	value
320x48	1
300x50	2
216x36	3
168x28	4

120x20	5
480x60	6
728x90	7
300x250	8
120x600	9

Note: *If this Ad Unit Size parameter is not specified, AdFalcon server will choose the best ad size that is suitable for the device's screen resolution.*

- **Language (U_LN)**
  The language of the requested ad in ISO 639-1 language codes format (Two Letters code); for Arabic language, pass **ar** code for language parameter U_LN=ar.

- **Age or birthdate (U_AGE or U_BD)**
  You can add one of these two parameters to determine the age of the user.

  In the case you pass a birthdate you must send it in the dd.MM.yyyy format

- **Gender (U_G)**
  This parameter represents the Gender of the user

Gender	Value
Male	m
Female	f

- **Keywords (U_KW)**
  This parameter is used to list a collection of keywords in comma separated format. These keywords can be useful since AdFalcon's ads selector engine will search for ads containing these keywords.

- **Country Code (U_CC)**
  County code of the end user in ISO 3166-1 alpha-2 format code (two-letter code)

- **Area Code (U_AC)**
  A parameter containing the user's area code

- **Postal Code (U_PC)**
  A parameter containing the user's postal/ZIP code

- **Geolocation (D_LA, D_LO)**
  A parameter containing the geolocation information of the device. The location information is divided into two double values; latitude (D_LA) and longitude (D_LO).

  Example

  D_LA=35.54454548&D_LO=58.981

JSP

```
<%@ page import="java.util.*,java.io.*,java.net.*,java.net.InetAddress;"%>

<%
//AdFalcon URL which used to get ads
 String adf_adFalconURL ="http://api.adfalcon.com/AdRequest/GetAd";
//Parameters is a variable that contains "get method request" paramerts
 String adf_parameters ="";
/*
 * You must fill the following required parameters because
 * adfalcon will not send ads without having those values
 */
String adf_siteId ="Site ID";

/*
* Ensure this is set to false once you release your web site to the public
*/
boolean adf_testMode =true;
/*
* Fill only in case specific ad sizes are required; otherwise remove this line of code, and AdFalcon will
determine the best ad size for the device
*/
int adf_adUnitSize =1;

/*
* Unique User Identification:
* AdFalcon will need to uniquely identify each user, therefore if you have
* a User unique ID for your web site users, then provide it here
* Otherwise, replies on AdFalcon to generate a unique ID for the which should be * save it in a persistent
cookie/session with following properties
* - Cookie value: AdFalcon's UUID or any unique value for user
* - Expires : maximum as possible
* - Path : /
*/

/*
* If you have your own user unique value, assign your value to userUniqueId variable
* note: remove all codes related to cookie when using your own user unique value
*/
String adf_userUniqueId ="";

/*
* this will get the user unique id generated by AdFalcon from the cookie in case this  was saved in a
previous ad request
*/
for (Cookie cookie : request.getCookies())
{
   if (cookie.getName().equalsIgnoreCase("ADFALCON-UUID"))
   {
      adf_userUniqueId =cookie.getValue();
   }
}

/*
 * You can help adfalcon to get better targeting ads by filling optional parameters, * these parameters ar
e demographics and device information.
 * you will find name of parameters in AdFalcon Server API integration guide at appendix A
 */
Hashtable<String, String>adf_optionalParams =null;
//adf_optionalParams = new Hashtable<String, String>();
 //adf_optionalParams.put("U_LN", "ar");
 //adf_optionalParams.put("U_KW", "sport, mobile");
 /*
 * Collects all needed information and builds "GET method request" URL
 */
adf_parameters +="R_SID=" +adf_siteId;
adf_parameters +="&R_IP=" +URLEncoder.encode(request.getRemoteAddr(), "UTF-8");
adf_parameters +="&R_F=XHTML";
adf_parameters +="&R_V=api-all-1.0.0";
adf_parameters +="&D_UA=" +URLEncoder.encode(request.getHeader("User-Agent"), "UTF-8");
//adf_parameters +="&R_AS=" +Integer.toString(adf_adUnitSize);
adf_parameters +="&R_TM=" +Boolean.toString(adf_testMode);
adf_parameters +="&R_UUID=" +URLEncoder.encode(adf_userUniqueId, "UTF-8");
```

```
if (adf_optionalParams !=null)
{
   for (String key : adf_optionalParams.keySet())
   {
      adf_parameters +="&" +"" +URLEncoder.encode(key, "UTF-
8") +"=" +URLEncoder.encode(adf_optionalParams.get(key), "UTF-8");
   }
}
//get end user's browser header names and add them to the request adf_parameters
 Enumeration adf_headerNames =request.getHeaderNames();
ArrayList adf_ignoreHeaders =new ArrayList();
adf_ignoreHeaders.add("CACHE-CONTROL");
adf_ignoreHeaders.add("CONNECTION");
adf_ignoreHeaders.add("COOKIE");
adf_ignoreHeaders.add("PRAGMA");
adf_ignoreHeaders.add("USER-AGENT");
while (adf_headerNames.hasMoreElements())
{
   String name =adf_headerNames.nextElement().toString();
   if (!adf_ignoreHeaders.contains(name.toUpperCase()))
   {
      String value =request.getHeader(name);
      if (name !=null &&value !=null)
      {
         adf_parameters +="&" +"R_HH_" +URLEncoder.encode(name.toUpperCase(), "UTF-
8") +"=" +URLEncoder.encode(value, "UTF-8");
      }
   }
}
//fill page query string
 adf_parameters +=(request.getQueryString() !=null ? "&" +request.getQueryString() : "");
//create GET method URL
 URL adf_url =new URL(adf_adFalconURL +"?" +adf_parameters);
/*
 *Open http connection with adfalcon server and get new ad
 */
HttpURLConnection adf_httpURLConnection =null;
InputStream adf_inputStream =null;
InputStreamReader adf_inputStreamReader =null;
BufferedReader adf_bufferedReader =null;
try
{
   adf_httpURLConnection =(HttpURLConnection) adf_url.openConnection();
   adf_httpURLConnection.setConnectTimeout(10000);
   adf_httpURLConnection.setReadTimeout(10000);
   adf_httpURLConnection.setDoInput(true);
   adf_httpURLConnection.setDoOutput(true);
   adf_httpURLConnection.setUseCaches(false);
   adf_httpURLConnection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
   adf_httpURLConnection.setRequestProperty("charset", "UTF-8");
   adf_httpURLConnection.setRequestMethod("GET");
   if (adf_httpURLConnection.getResponseCode() ==HttpURLConnection.HTTP_OK)
   {
      /*
 * in case a request was not completed successfully;
 * adfalcon will send two header's parameters
 * - X-ADFALCON-ERROR-CODE: status code of the request
 * - X-ADFALCON-ERROR-MESSAGE: message descripes the status code
 */
      if (adf_httpURLConnection.getHeaderField("X-ADFALCON-ERROR-CODE") !=null)
      {
         out.println(
         adf_httpURLConnection.getHeaderField("X-ADFALCON-ERROR-CODE") +":"
         +adf_httpURLConnection.getHeaderField("X-ADFALCON-ERROR-MESSAGE"));
      }else
      {
         /*
 * Receive an ad from adfalcon server and fill it in adf_adFalconResponse variable
 */
         adf_inputStream =adf_httpURLConnection.getInputStream();
         adf_inputStreamReader =new InputStreamReader(adf_inputStream);
         adf_bufferedReader =new BufferedReader(adf_inputStreamReader);
         String adf_adFalconResponse ="";
         while (adf_bufferedReader.ready())
```

```
            {
                adf_adFalconResponse +=adf_bufferedReader.readLine();
            }
            /*
 * If adf_adFalconResponse variable is not null or empty, print it at page
 */
            if (adf_adFalconResponse !=null &&!adf_adFalconResponse.trim().isEmpty())
            {
                out.append(adf_adFalconResponse);
            }
            /*
 * Check if adfalcon server has sent UUID, * if yes, you should save it in presistent cookie/session
 *
 */
            if (adf_httpURLConnection.getHeaderField("X-ADFALCON-UUID") !=null)
            {
                /*
 * if you want to use the adfalcon UUID and set it to cookie, still activate the following code
 */
                Cookie cookie =new Cookie("ADFALCON-UUID", adf_httpURLConnection.getHeaderField("X-
ADFALCON-UUID"));
                cookie.setMaxAge(10 *365 *24 *60 *60);
                cookie.setPath("/");
                response.addCookie(cookie);
                /*
 * else set adfalcon UUID wherever you want
 */
            }
        }
    }
}catch (Exception ex)
{}
finally
{
    if (adf_inputStream !=null)
    {
        adf_inputStream.close();
    }
    if (adf_inputStreamReader !=null)
    {
        adf_inputStreamReader.close();
    }
    if (adf_bufferedReader !=null)
    {
        adf_bufferedReader.close();
    }
    if (adf_httpURLConnection !=null)
    {
        adf_httpURLConnection.disconnect();
    }
}
%>
```

PHP-CURL

```
<?php
//AdFalcon URL that used to get ads
$adf_adFalconURL ="http://api.adfalcon.com/AdRequest/GetAd";
//this variable used to hold request's parameters
$adf_parameters ="";
/*
 * You must fill the following required parameters because
 * adfalcon will not send ads without having those values
 */
$adf_siteId ="Site ID";
/*
* Ensure this is set to false once you release your web site to the public
*/
```

```
$adf_testMode =true;
/*
* Fill only in case specific ad sizes are required; otherwise remove this line of code, and AdFalcon will
determine the best ad size for the device
*/
$adf_adUnitSize =1;

/*
* Unique User Identification:
* AdFalcon will need to uniquely identify each user, therefore if you have
* a User unique ID for your web site users, then provide it here
* Otherwise, replies on AdFalcon to generate a unique ID for the which should be * save it in a persistent
cookie/session with following properties
* - Cookie value: AdFalcon's UUID or any unique value for user
* - Expires : maximum as possible
* - Path : /
*/

/*
* If you have your own user unique value, assign your value to userUniqueId variable
* note: remove all codes related to cookie when using your own user unique value
*/
$adf_userUniqueId ="";

/*
* this will get the user unique id generated by AdFalcon from the cookie in case this  was saved in a
previous ad request
*/
$adf_cookie_UUID =$_COOKIE["ADFALCON-UUID"];
if ($adf_cookie_UUID !="" and $userUniqueId =="")
{
   $adf_userUniqueId =$adf_cookie_UUID;
}

/*
 * You can help adfalcon to get better targeting ads by filling optional parameters, * these parameters ar
e demographics and device information.
 * you will find name of parameters in AdFalcon Server API integration guide at appendix A
 */
$adf_optionalParams =array();
$adf_optionalParams["U_LN"] ="ar";
$adf_optionalParams["U_KW"] ="mobile, sport";
/*
 * Collects all needed information and builds URL for GET request
 */
$adf_parameters .="R_SID=" . $adf_siteId;
$adf_parameters .="&R_IP=" . rawurlencode($_SERVER['REMOTE_ADDR']);
$adf_parameters .="&R_F=XHTML";
$adf_parameters .="&R_V=api-all-1.0.0";
$adf_parameters .="&D_UA=" . rawurlencode($_SERVER['HTTP_USER_AGENT']);
//$adf_parameters .="&R_AS=" . $adf_adUnitSize;
$adf_parameters .="&R_TM=" . ($adf_testMode ? 'true' : 'false');
$adf_parameters .="&R_UUID=" . rawurlencode($adf_userUniqueId);
if ($adf_optionalParams !=null)
{
   $adf_count =count($adf_optionalParams);
   foreach ($adf_optionalParams as $adf_key =>$adf_value)
   {
      $adf_parameters .="&" . "" . rawurlencode($adf_key) . "=" . rawurlencode($adf_value);
   }
}
//get end user's browser header names and add them to the request parameters
$adf_ignore =array(
'HTTP_PRAGMA' =>'ignore', 'HTTP_CACHE_CONTROL' =>'ignore', 'HTTP_CONNECTION' =>'ignore', 'HTT
P_USER_AGENT' =>'ignore', 'HTTP_COOKIE' =>'ignore'
);
foreach ($_SERVER as $adf_key =>$adf_val)
{
   if (0 ===strpos($adf_key, 'HTTP_') &&!isset($adf_ignore[$adf_key]) &&isset($adf_val))
   {
      $adf_key =str_replace('HTTP_', '', $adf_key);
      $adf_parameters .="&" . "R_HH_" . rawurlencode(strtoupper($adf_key)) . "=" . rawurlencode($adf
_val);
   }
```

```
}
//fill page query string
//$parameters .= (request.getQueryString() != null ? "&" + request.getQueryString() : "");
//create GET method URL
$adf_url_get =$adf_adFalconURL . "?" . $adf_parameters;
/*
 * Open http connection with adfalcon server and get new ad
 */
$adf_copt =array(
CURLOPT_URL =>$adf_url_get
, CURLOPT_RETURNTRANSFER =>true
, CURLOPT_HEADER =>true
, CURLOPT_HTTPPROXYTUNNEL =>true
, CURLOPT_CONNECTTIMEOUT =>20
, CURLOPT_TIMEOUT =>20
, CURLOPT_HTTPHEADER =>array('Content-Type: application/x-www-form-urlencoded')
);
$adf_ch =curl_init();
curl_setopt_array($adf_ch, $adf_copt);
$adf_response =curl_exec($adf_ch);
$adf_responseInfo =curl_getinfo($adf_ch);
curl_close($adf_ch);
echo $adf_responseInfo['CURLOPT_NOBODY'];
if ($adf_responseInfo['http_code'] ===200)
{
   //check header values
    list($adf_header, $adf_body) =explode("\r\n\r\n", $adf_response, 2);
   $adf_status =strtok($adf_header, "\r\n");
   $adf_tok =strtok("\r\n");
   while ($adf_tok !==false)
   {
      if (strpos($adf_tok, ":") >=0)
      {
         list($adf_name, $adf_value) =explode(":", $adf_tok);
         if (strpos(trim($adf_name), "X-ADFALCON-UUID") !==false)
         {
            $adf_adfalcon_header_UUID =$adf_value;
         }else if (strpos(trim($adf_name), "X-ADFALCON-ERROR-CODE") !==false)
         {
            $adf_adfalcon_header_error_code =$adf_value;
         }else if (strpos(trim($adf_name), "X-ADFALCON-ERROR-MESSAGE") !==false)
         {
            $adf_adfalcon_header_error_message =$adf_value;
         }
      }
      $adf_tok =strtok("\n\r");
   }
   /*
 * in case a request was not completed successfully;
 * adfalcon will send two header's parameters
 * - X-ADFALCON-ERROR-CODE: status code of the request
 * - X-ADFALCON-ERROR-MESSAGE: message descripes the status code
 */
   if (isset($adf_adfalcon_header_error_code))
   {
      echo $adf_adfalcon_header_error_message;
   }else
   {
      /*
 * Check if adfalcon sends UUID, * and save it in a presistent cookie/session.
 *
 */
      if (isset($adf_adfalcon_header_UUID) &&strlen($adf_userUniqueId) >0)
      {
         /*
 * save X-ADFALCON-UUID feild to end user's browser cookie
 */
         setcookie("ADFALCON-
UUID",  $adf_adfalcon_header_UUID, time() +10 *365 *24 *60 *60 *3600, "/");
      }
      /*
 * If $adf_body variable is not null or empty, print it at page
 */
      if (null !==$adf_body)
```

```
        echo( $adf_body );
    }
}
?>
```

C#

```
<%//AdFalcon URL which used to get ads
String adf_adFalconURL ="http://api.adfalcon.com/AdRequest/GetAd";
//Parameters is a variable that contains "get method request" paramerts
String adf_parameters ="";
/*
* You must fill the following required parameters because
* adfalcon will not send ads without having those values
*/
String adf_siteId ="Site ID";
/*
* Ensure this is set to false once you release your web site to the public
*/
bool adf_testMode =true;
/*
* Fill only in case specific ad sizes are required; otherwise remove this line of code, and AdFalcon will
determine the best ad size for the device
*/
int adf_adUnitSize =1;

/*
* Unique User Identification:
* AdFalcon will need to uniquely identify each user, therefore if you have
* a User unique ID for your web site users, then provide it here
* Otherwise, replies on AdFalcon to generate a unique ID for the which should be
* save it in a persistent cookie/session with following properties
* - Cookie value: AdFalcon's UUID or any unique value for user
* - Expires : maximum as possible
* - Path : /
*/

/*
* If you have your own user unique value, assign your value to userUniqueId variable
* note: remove all codes related to cookie when using your own user unique value
*/
String adf_userUniqueId ="";

/*
* this will get the user unique id generated by AdFalcon from the cookie in case this  was saved in a
previous ad request
*/
HttpCookie adf_cookie =Request.Cookies.Get("ADFALCON-UUID");
if (adf_cookie !=null)
{
   adf_userUniqueId =adf_cookie.Value;
}


/*
* You can help adfalcon to get better targeting ads by filling optional parameters, * these parameters are
 demographics and device information.
* you will find name of parameters in AdFalcon Server API integration guide at appendix A
*/
Dictionary<string, string> adf_optionalParams =null;
// adf_optionalParams = new Dictionary<string, string>();
// adf_optionalParams.put("U_LN", "ar");
// adf_optionalParams.put("U_KW", "sport, mobile");
/*
* Collects all needed information and builds "GET method request" URL
*/
adf_parameters +="R_SID=" +Server.UrlEncode(adf_siteId);
adf_parameters +="&R_IP=" +Server.UrlEncode(Request.UserHostAddress);
adf_parameters +="&R_F=XHTML";
```

```
adf_parameters +="&R_V=api-all-1.0.0";
adf_parameters +="&D_UA=" +Server.UrlEncode(Request.UserAgent);
//adf_parameters +="&R_AS=" +adf_adUnitSize;
adf_parameters +="&R_TM=" +adf_testMode;
adf_parameters +="&R_UUID=" +Server.UrlEncode(adf_userUniqueId);
if (adf_optionalParams !=null)
{
   foreach (String key in adf_optionalParams.Keys)
   {
      adf_parameters +="&" +"" +key +"=" +Server.UrlEncode((String) adf_optionalParams[key]);
   }
}//get end user's browser header names and add them to the request parameters
List<string> adf_ignoreHeaders =new List<string>();
adf_ignoreHeaders.Add("CACHE-CONTROL");
adf_ignoreHeaders.Add("CONNECTION");
adf_ignoreHeaders.Add("COOKIE");
adf_ignoreHeaders.Add("PRAGMA");
adf_ignoreHeaders.Add("USER-AGENT");
foreach (string name in Request.Headers)
{
   if (!adf_ignoreHeaders.Contains(name.ToUpper()))
   {
      String value =Request.Headers[name];
      if (name !=null &&value !=null)
      {
         adf_parameters +="&" +"R_HH_" +name.ToUpper() +"=" +Server.UrlEncode(value);
      }
   }
}/*
*Open http connection with adfalcon server and get new ad
*/
System.Net.WebRequest adf_webRequest =null;
System.Net.WebResponse adf_webResponse =null;
System.IO.StreamReader adf_StreamReader =null;
try
{
   adf_webRequest =System.Net.WebRequest.Create(adf_adFalconURL +"?" +adf_parameters);
   adf_webRequest.ContentType ="application/x-www-form-urlencoded";
   adf_webRequest.Method ="GET";
   adf_webResponse =adf_webRequest.GetResponse();
   adf_StreamReader =new System.IO.StreamReader(adf_webResponse.GetResponseStream());
   /*
* in case a request was not completed successfully;
* adfalcon will send two header's parameters
* - X-ADFALCON-ERROR-CODE: status code of the request
* - X-ADFALCON-ERROR-MESSAGE: message descripes the status code
*/
   if (adf_webResponse .Headers.Get("X-ADFALCON-ERROR-CODE") !=null)
   {
      /*
      * Handle error…
      */
      Response.Write(
      adf_webResponse .Headers.Get("X-ADFALCON-ERROR-CODE") +":"
      + adf_webResponse .Headers.Get("X-ADFALCON-ERROR-MESSAGE"));
   }else
   {
      /*
* Check if adfalcon server has sent UUID, * if yes, you should save it in presistent cookie/session
*
*/
      if (adf_webResponse .Headers.Get("X-ADFALCON-UUID") !=null)
      {
         /*
* if you want to use the adfalcon UUID and set it to cookie, still activate the following code
*/
         HttpCookie adf_newCookie =new HttpCookie("ADFALCON-
UUID", (String) adf_webResponse .Headers.Get("X-ADFALCON-UUID"));
         adf_newCookie.Expires = DateTime().Now.AddHours(10 *365 *24);
         adf_newCookie.Path ="/";
         Response.Cookies.Add(adf_newCookie);
         /*
* else set adfalcon UUID wherever you want
*/
```

```
        }/*
* Receive an ad from adfalcon server and fill it in adFalconResponse variable
*/
        String adf_adFalconResponse =adf_StreamReader.ReadToEnd();
        /*
* If adFalconResponse variable is not null or empty, print it at page
*/
        if (adf_adFalconResponse !=null &&adf_adFalconResponse.Trim().Length !=0)
        {
            Response.Write(adf_adFalconResponse);
        }
    }
}catch (Exception ex)
{
    Response.Write(ex.Message);
}
finally
{
    if (adf_StreamReader !=null)
    {
        adf_StreamReader.Close();
    }
    if (adf_webResponse !=null)
    {
        adf_webResponse.Close();
    }
}
%>
```