WinPure Email Verification API Documentation

November 2017

The URL to pass to the API is

```
https://api.winpure.com/api/a/v1?key=yourapikey&email=test@tester.com
```

Parameter	Value	Mandatory or Optional
key	your unique API key	Mandatory
email	the email address to be validated	Mandatory
correct	**1** (this will remove certain invalid characters) **0** (this will leave the email untouched)	Optional

IMPORTANT NOTES REGARDING INTEGRATION
1. The response is a JSON data string and a JSON function must be used to decode it. Please do not rely on parsing a text string as this may cause issues.
2. We recommend that any "Unknown" results are treated as "OK" (valid). This will prevent potentially valid emails from being rejected.

The response would be :-

```
{

"status":"Bad",

"additionalStatus":"MailboxDoesNotExist",

"emailAddressProvided":"test@tester.com",

"emailAddressChecked":"test@tester.com",

"emailAddressSuggestion":""

}
```

Response	Description	Example values
status	The validation result	'Ok', 'Bad', 'Unknown'
additionalStatus	Additional information for BAD and UNKNOWN results	NoMxServersFound
emailAddressProvided	The exact email address passed to the API including obvious typos and errors	test@tester.com
emailAddressChecked	The actual email address validated	test@tester.com

Response	Description	Example values
emailAddressSuggestion	A suggested alternative if a common typo was noticed e.g. if 'fred@hotmail.cm' was provided	fred@hotmail.com

Main Status Response Codes

Response	Explanation
Ok	Verification passes all checks including Syntax, DNS, MX, Mailbox, Deep Server Configuration, Grey Listing
Bad	Verification fails checks for definitive reasons (e.g. mailbox does not exist)
Unknown	Conclusive verification result cannot be achieved due to mail server configuration or anti-spam measures. See table "Additional Status Codes".

Using the typo correction feature

Optionally, you can also use the **'correct' parameter** to remove certain invalid characters such as spaces, slashes, square brackets etc. Example using the 'correct' parameter. The user enters an email address *john99]@gmail.com* Here is the API call that would be made :-

```
https://api.winpure.com/api/a/v1?key=yourapikey&email=john99]@gmail.com&correct=1
```

The API will automatically remove the invalid character ']' and send the corrected version through for validation. Example results based on the above API call :-

```
{

"status":"Ok",

"additionalStatus":"None",

"emailAddressProvided":"john99]@gmail.com",

"emailAddressChecked":"john99@gmail.com",

"emailAddressSuggestion":""

}
```

When an email address is returned with a status of *Bad* or *Unknown* we return the detailed reason as part of the response in the *additionalStatus* value. For a full list of additional status values, please refer to the list below.

Authentication without the license key – ACL

Many situations require that the API license key is not displayed, e.g. where the API is used within a client-side application such as jQuery in a website form.

Access the API at:

```
https://api.winpure.com/api/a/v1?email=test@tester.com
```

For ACL based authentication (without using the key), it is not necessary to include the license key in the request. Other parameters (e.g. 'correct') are supported as in the key based example.

In this case the authentication will be done using the domain name hosting the files where the API is embedded. Please send us a list of your domain names that will be hosting your API integration so that we can add them to the permitted list for your account.

Additional Status Codes

None:

No additional information is available.

This status differs from a TransientNetworkFault as it should not be retried (the result will not change).

There are a few known reasons for this status code for example the target mx record uses Office 365 or a mail provider implementing custom mailbox shutdowns.

AtSignNotFound: The required '@' sign is not found in email address.

DomainIsInexistent:

The domain (i.e. the bit after the '@' character) defined in the email address does not exist, according to DNS records.

A domain that does not exist cannot have email boxes. A domain that does not exist cannot have email boxes.

DomainIsWellKnownDea:

The domain is a well known Disposable Email Address DEA.

There are many services available that permit users to use a one-time only email address. Typically, these email addresses are used by individuals wishing to gain access to content or services requiring registration of email addresses but same individuals not wishing to divulge their true identities (e.g. permanent email addresses).

DEA addresses should not be regarded as valid for email send purposes as it is unlikely that messages sent to DEA addresses will ever be read.

The mailbox is full.

Mailboxes that are full are unable to receive any further email messages until such time as the user empties the mail box or the system administrator grants extra storage quota.

MailboxFull:

Most full mailboxes usually indicate accounts that have been abandoned by users and will therefore never be looked at again.

We do not recommend sending emails to email addresses identified as *full*.

MailboxDoesNotExist:

The mailbox does not exist.

100% confidence that the mail box does not exist.

NoMxServersFound:

There are no mail servers defined for this domain, according to DNS.

Email addresses cannot be valid if there are no email servers defined in DNS for the domain.

ServerDoesNotSupportInternationalMailboxes:

The server does not support international mailboxes.

International email boxes are those that use international character sets such as Chinese / Kanji etc.

International email boxes require systems in place for Punycode translation.

Where these systems are not in place, email verification or delivery is not possible.

ServerIsCatchAll:

The server is configured for *catch all* and responds to all email verifications with a status of *Ok*.

Mail servers can be configured with a policy known as *Catch All*. Catch all redirects any email address sent to a particular domain to a central email box for manual inspection. Catch all configured servers cannot respond to requests for email address verification.

Successful verification.

Success: 100% confidence that the mail box exists.

TooManyAtSignsFound:

Too many '@' signs found in email address.

Only one '@' character is allowed in email addresses.

Unknown: The reason for the verification result is unknown.

TransientNetworkFault:

A temporary network fault occurred during verification. Please try again later.

Verification operations on remote mail servers can sometimes fail for a number of reasons such as loss of network connection, remote servers timing out etc.

One other possible cause of a temporary fault is Grey Listing (i.e. the target mail server blocks the first connection attempt and requests a delayed re-try).

These conditions are usually temporary. Retrying verification at a later time will usually result in a positive response from mail servers.

Please note that setting an infinite retry policy around this status code is inadvisable as there is no way of knowing when the issue will be resolved within the target domain or the grey listing resolved, and this may affect your daily quota.

PossibleSpamTrapDetected:

A possible spam trap email address or domain has been detected.

Spam traps are email addresses or domains deliberately placed on-line in order to capture and flag potential spam based operations.

Our advanced detection heuristics are capable of detecting likely spam trap addresses or domains known to be associated with spam trap techniques.

We do not recommend sending emails to addresses identified as associated with known spam trap behaviour.

Sending emails to known spam traps or domains will result in your ESP being subjected to email blocks from a DNS Blcok List.

An ESP cannot tolerate entries in a Block List (as it adversely affects email deliver-ability for all customers) and will actively refuse to send emails on behalf of customers with a history of generating entries in a Block List.

Code Examples

We have provided extensive code examples below.

PHP

```
<?php



// URL which should be requested

$url = 'http://api.winpure.com/api/a/v1';



$apikey = 'YOUR API KEY'; // API Key



$email = 'Email Address to Test'; // Email to test



// jSON String for request

$url .= "?email=$email&key=$apikey";



// Initializing curl

$ch = curl_init( $url );



if($ch == false) {



die ("Curl failed!");



} else {
```

```
// Configuring curl options

$options = array(

CURLOPT_RETURNTRANSFER => true,

CURLOPT_HTTPHEADER => array('Content-type: application/json')

);



// Setting curl options

curl_setopt_array( $ch, $options );



// Getting results

$result = curl_exec($ch); // Getting jSON result string



// display JSON data

echo "$result";



}



?>
```

C#

```
#region Usings
```

```
using System;

using System.IO;

using System.Net;



#endregion



/// <summary>

/// The program.

/// </summary>

internal class Program

{

        #region Constants



        /// <summary>

        /// The api url.

        /// </summary>

        private const string ApiUrl = @"http://api.winpure.com/api/a/v1";

        /// <summary>

        /// 0 = ApiUrl

        /// 1 = Email address to query

        /// 2 = API Key

        /// </summary>

        private const string QueryFormatString = @"{0}?email={1}&key={2}";
```

```
/// <summary>

/// The your api key.

/// </summary>

/// <remarks>

/// /*ENTER YOUR API KEY HERE*/

/// </remarks>

private const string YourAPIKey = @"<!-- ENTER A VALID KEY HERE-->";



#endregion



#region Methods



/// <summary>

/// The main program entry point.

/// </summary>

/// <param name="args">

/// The args.

/// </param>

private static void Main(string[] args)

{

        Console.WriteLine("Input email address to verify");
```

```
                var readLine = Console.ReadLine();



                Console.WriteLine(string.Empty);



                var requestUrl = string.Format(QueryFormatString, ApiUrl, readLine,
YourAPIKey);



                var myRequest = (HttpWebRequest)WebRequest.Create(requestUrl);



                WebResponse webResponse = null;



                try

                {

                        webResponse = myRequest.GetResponse();



                        using (var reader = new
StreamReader(webResponse.GetResponseStream()))

                        {

                                var jsonString = reader.ReadToEnd();



                                Console.ForegroundColor = ConsoleColor.Green;

                                Console.WriteLine("Result:");

                                Console.WriteLine(jsonString);

                                Console.ResetColor();
```

```
                        Console.WriteLine("Press <Enter> to continue..");
                        Console.ReadLine();
                }
        }
        catch (Exception exception)
        {
                Console.WriteLine("An error occured:");
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("Exception reported: {0}", exception.Message);
                Console.ResetColor();
                Console.WriteLine("Press <Enter> to continue..");
                Console.ReadLine();
        }
        finally
        {
                if (webResponse != null)
                {
                        webResponse.Dispose();
                }
        }
}

#endregion
```

```
}
```

VB.net

```
Imports System.IO

Imports System.Net



''' <summary>

''' The program.

''' </summary>

Friend Class Program

        #Region "Constants"



        ''' <summary>

        ''' The api url.

        ''' </summary>

        Private Const ApiUrl As String = "http://api.winpure.com/api/a/v1"



        ''' <summary>

        ''' 0 = ApiUrl

        ''' 1 = Email address to query

        ''' 2 = API Key

        ''' </summary>

        Private Const QueryFormatString As String = "{0}?email={1}&key={2}"
```

```
''' <summary>

''' The your api key.

''' </summary>

''' <remarks>

''' /*ENTER YOUR API KEY HERE*/

''' </remarks>

Private Const YourAPIKey As String = "<!-- ENTER A VALID KEY HERE-->"



#End Region



#Region "Methods"



''' <summary>

''' The main program entry point.

''' </summary>

''' <param name="args">

''' The args.

''' </param>

Private Shared Sub Main(args As String())



        Console.WriteLine("Input email address to verify")



        Dim readLine = Console.ReadLine()
```

```
                Console.WriteLine(String.Empty)

                Dim requestUrl = String.Format(QueryFormatString, ApiUrl, readLine,
YourAPIKey)

                Dim myRequest = DirectCast(WebRequest.Create(requestUrl), HttpWebRequest)

                Dim webResponse As WebResponse = Nothing

                Try
                        webResponse = myRequest.GetResponse()

                        Using reader = New StreamReader(webResponse.GetResponseStream())
                                Dim jsonString = reader.ReadToEnd()

                                Console.ForegroundColor = ConsoleColor.Green
                                Console.WriteLine("Result:")
                                Console.WriteLine(jsonString)
                                Console.ResetColor()
                                Console.WriteLine("Press <Enter> to continue..")
                                Console.ReadLine()
                        End Using
                Catch exception As Exception
```

```
                        Console.WriteLine("An error occured:")

                        Console.ForegroundColor = ConsoleColor.Red

                        Console.WriteLine("Exception reported: {0}", exception.Message)

                        Console.ResetColor()

                        Console.WriteLine("Press <Enter> to continue..")

                        Console.ReadLine()

                Finally

                        If webResponse IsNot Nothing Then

                                webResponse.Dispose()

                        End If

                End Try

        End Sub



        #End Region

End Class
```

Java

```
/*

*****************************************************************************************

* File name:

* java.v1_key_acl

*

* Version:

* 1.0.20140827.0
```

```
*
* Version control:
* - 1.0.20140827.0 - initial release
*
* Date:
* August 2014
*
* Description:
* Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1
* using java.
*
* This example requires a valid key to work correctly.
*
* License:
* Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)
*******************************************************************************
*/


import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
```

```
import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;



public class Program

{



        /*

         * The URL which should be requested

         */

        private static final String ApiUrl = "http://api.winpure.com/api/a/v1";



        /*

         * Query string for request

         * %1$s = ApiUrl

         * %2$s = Email address to query

         * %3$s = API Key

         */

        private static final String QueryFormatString = "%1$s?email=%2$s&key=%3$s";



        /*

         * Your API Key

         *
```

```
         */
        private static final String YourAPIKey = "/* ENTER A VALID KEY HERE */";


        /**
         * The main program entry point
         * @param args the command line arguments
         * @throws IOException If the server does not return a success response
         */
        public static void main(String[] args)
        {
                System.out.println("Input email address to verify");


                // Create a scanner to read in the requested email address
                Scanner in = new Scanner(System.in);
                String readLine = in.next();


                try
                {
                        // Format the request url to the correct structure for the request
                        URL requestUrl = new URL(String.format(QueryFormatString, ApiUrl,
readLine, YourAPIKey));


                        // Open a connection to the website
```

```
                        HttpURLConnection myRequest =  (HttpURLConnection)
requestUrl.openConnection();



                        // Set the type to HTTP GET

                        myRequest.setRequestMethod("GET");



                        // Create a new buffered reader to read the response back from the
server

                        BufferedReader reader = new BufferedReader(

                                new InputStreamReader(myRequest.getInputStream()));



                        String inputLine;

                        StringBuilder response = new StringBuilder();



                        // Read in the response line from the server

                        while ((inputLine = reader.readLine()) !=null )

                        {

                                response.append(inputLine);

                        }



                        // Close the reader

                        reader.close();



                        // Output the result to console
```

```
                        System.out.println(response.toString());

                }

                catch (IOException ex)

                {

                        Logger.getLogger(Program.class.getName()).log(Level.SEVERE, null,
ex);

                }

        }



}
```

Python

```
##########################################################################################################

#   Version:

#   1.0.20140827.0

#

#   Version control:

#   - 1.0.20140827.0 - initial release

#

#   Date:

#   August 2014

#

#   Description:

#   Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1
```

```
#   using Python
#
#   This example requires a valid key to work correctly.
#
#   License:
#   Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)
#############################################################################################


    # Import the module for handling the http request
    import urllib.request


    # The url for the service
    ApiUrl = "http://api.winpure.com/api/a/v1"


    # The format  of the full query string
    QueryFormatString = "{0}?email={1}&key={2}"


    # The API key provided for your account goes here
    YourAPIKey = "<!-- ENTER A VALID KEY HERE-->"


    # Read in the user input
    readLine = input("Enter Email:\n")
```

```
    # Format the full url and make the http GET request and read the response

    response = urllib.request.urlopen(QueryFormatString.format(ApiUrl, readLine,
YourAPIKey)).read()



    # Print the response

    print(response)
```

PERL

```
##############################################################################################

#   File name:

#   Program.pl

#

#   Version:

#   1.0.20140828.0

#

#   Version control:

#   - 1.0.20140828.0 - initial release

#

#   Date:

#   August 2014

#

#   Description:

#   Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1

#   using Perl using client side only calls.
```

```
#
#   This example requires a valid key to work correctly.
#
#   License:
#   Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)
################################################################################################
    use LWP;


    # The url for the service
    $ApiUrl = "http://api.winpure.com/api/a/v1";


    # The format  of the full query string
    $QueryFormatString = "%s?email=%s&key=%s";


    print "Enter Email:\n";
    $readLine = <STDIN>;


    # The API key provided for your account goes here
    $YourAPIKey = "<!-- ENTER A VALID KEY HERE-->";


    # Format the url request
    $requestUrl = sprintf $QueryFormatString, $ApiUrl, $readLine, $YourAPIKey;
```

```
    # Make the request against the REST service

    $browser = LWP::UserAgent->new;

    $response = $browser->get($requestUrl);



    # Print the response

    print $response->content;
```

jQuery (domain ACL)

Note

Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1 using jQuery, client side only calls.

```
<!DOCTYPE html>



<!--

*****************************************************************************************

*   File name:

*   v1_domain_acl.html

*

*   Version:

*   1.0.20140827.0

*

*   Version control:

*   - 1.0.20140827.0 - initial release

*

*   Date:
```

```
*   August 2014
*
*   Description:
*   Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1
*   using jQuery, client side only calls.
*
*   This example requires a domain ACL and hosting at specified domain to work correctly.
*
*   License:
*   Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)
*********************************************************************************************
-->
    <html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
            <meta charset="utf-8" />
            <title>emailverifyapp.com : Domain Access Control List Sample.</title>
            <style type="text/css">
                    .statusUnknown{ color: #c1c72c;}
                    .statusOk{ color: #009933;}
                    .statusBad,.errorMsg{ color: #ff0000;}
                    input[type='text']{ width: 300px;}
                    p label {display: inline-block; width: 60px;}
            </style>
```

```
            <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>

    </head>

            <body>

                    <h1>emailverifyapp.com : email verification demo using domain access
control list authentication with jQuery.</h1>

                    <h2>About</h2>

                    <p>This example shows how to perform email verification using just
client side scripting and invoking a domain based <abbr title="Access Control List">ACL</
abbr> RESTful endpoint at https://api.winpure.com.</p>

                    <h2>How to run this sample</h2>

                    <p>Two things are needed to run this sample:</p>

                    <ol>

                            <li>A domain <abbr title="Access Control List">ACL</abbr> is
required from your email verification API provider. Domain ACLs are applied against a
domain of your choosing (e.g. www.yourdomain.com).</li>

                            <li>Web hosting for www.yourdomain.com.</li>

                    </ol>

                    <p>Please upload this sample to your web host in order to run.</p>

                    <h2>Key features</h2>

                    <ul>

                            <li>Compatible with all modern browsers</li>

                            <li>Uses jQuery 1.11.1</li>

                            <li>No server side scripting required</li>

                    </ul>

                    <hr/>
```

```
                    <h2>Try it</h2>

                    <p>

                            <label for="email">Email:</label>

                            <input type="text" name="email" id="email" />

                            <input type="button" name="submit" id="submit" value="verify"/>

                    </p>

                    <div id="validationResult"></div> <!--Result output here-->

                    <script>

                            /*nest key logic inside document.ready to ensure functionality
only available once document has fully loaded in browser.*/

                            $(function () {

                                    console.log("ready!");



                                    $('#submit').click(function () {

                                            var emailText = $('#email').val();



                                            if (emailText.length == 0) {

                                                    $('#validationResult').html("<span
class='errorMsg'>Please enter something.</span>");

                                                    return;

                                            }



                                            $('#validationResult').html("verifying...");
```

```
                                        var emailVerifyApi = '//api.winpure.com/api/a/v1?email=' + encodeURIComponent(emailText);



                                        /*execute remote request to perform email verification. Any errors will appear in the developer console (e.g. viewable using Chrome developer tools)*/

                                        $.getJSON(emailVerifyApi, {})

                                                .done(function (data) {

                                                        reportResult(data);

                                                })

                                                .fail(function (jqxhr, textStatus, error) {

                                                        var err = textStatus + ", " + error;

                                                        console.log("Request failed: " + err);

                                                });;

                                });

                        });



                        /*Output result to the 'validationResult' div element*/

                        function reportResult(data) {

                                var status = data['status'].toLowerCase(); // get 'status' from REST response

                                var additionalStatus = data['additionalStatus']; // get 'additionalStatus' from REST response

                                var message = data['Message']; // if there is an error (e.g. license issues), a notification will appear in the 'Message" from REST response.
```

```
                                    console.log(status);
                                    console.log(additionalStatus);
                                    console.log(message);

                                    var statusHtml;

                                    // if there is an error message, show here
                                    if (message != null
                                            && message != '') {
                                            statusHtml = "<span class='errorMsg'>Error.
Message='" + message + "' .</span>";
                                    } else {
                                            // map REST response data to presentation
messages.
                                            switch (status) {
                                                    case 'ok':
                                                            statusHtml = "<span
class='statusOk'>Email address is ok.</span>";
                                                            break;
                                                    case 'bad':
                                                            statusHtml = "<span
class='statusBad'>Email address is not valid.</span>";
                                                            break;
                                                    default:
```

```
                                                        statusHtml = "<span
class='statusUnknown'>Unable to validate email. Reason=" + additionalStatus + "</span>";

                                                        break;

                                                }

                                        }



                                        console.log(statusHtml);



                                        // present the result on screen

                                        $('#validationResult').html(statusHtml);

                                }

                        </script>

                </body>

        </html>
```

jQuery (license key)

Note

Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1 using jQuery using client side only calls

```
<!DOCTYPE html>



<!--

*****************************************************************************************

*   File name:

*   v1_key_acl.html

*
```

```
*   Version:
*   1.0.20140827.0
*
*   Version control:
*   - 1.0.20140827.0 - initial release
*
*   Date:
*   August 2014
*
*   Description:
*   Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1
*   using jQuery using client side only calls.
*
*   This example requires a valid key to work correctly.
*
*   License:
*   Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)
**********************************************************************************************
-->
    <html lang="en" xmlns="http://www.w3.org/1999/xhtml">
    <head>
            <meta charset="utf-8" />
            <title>emailverifyapp.com : License Key Sample.</title>
```

```
            <style type="text/css">
                    .statusUnknown {
                            color: #c1c72c;
                    }

                    .statusOk {
                            color: #009933;
                    }

                    .statusBad, .errorMsg {
                            color: #ff0000;
                    }

                    input[type='text'] {
                            width: 300px;
                    }
                    p label {
                            display: inline-block; width: 60px;
                    }
            </style>
            <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>
    </head>
    <body>
```

```
            <h1>emailverifyapp.com : email verification demo using simple key
authentication with jQuery.</h1>

            <h2>About</h2>

            <p>This example shows how to perform email verification using just client side
scripting and invoking a simple key based RESTful endpoint at https://api.winpure.com.</p>

            <h2>How to run this sample</h2>

            <p>This page can be hosted anywhere (i.e. any web host or platform). The only
thing needed is a valid license key.</p>

            <h2>Key features</h2>

            <ul>

                    <li>Compatible with all modern browsers</li>

                    <li>Uses jQuery 1.11.1</li>

                    <li>No server side scripting required</li>

            </ul>

            <hr />

            <h2>Try it</h2>

            <p>

                    <label for="key">Key:</label>

                    <input type="text" id="key" name="key" tabindex="1" maxlength="20" />

            </p>

            <p>

                    <label for="email">Email:</label>

                    <input type="text" name="email" id="email" tabindex="2" />

                    <input type="button" name="submit" id="submit" tabindex="3"
value="verify" />
```

```
            </p>

            <div id="validationResult"></div> <!--Result output here-->

            <script>

                            /*nest key logic inside document.ready to ensure functionality
only available once document has fully loaded in browser.*/

                            $(function () {

                                    console.log("ready!");



                                    $('#submit').click(function () {

                                            var emailText = $('#email').val(); // get key
from text box entry

                                            var keyText = $('#key').val(); // get email
address to be checked from text box



                                            if (keyText.length == 0) {

                                                    $('#validationResult').html("<span
class='errorMsg'>Please enter key.</span>");

                                                    return;

                                            }



                                            if (emailText.length == 0) {

                                                    $('#validationResult').html("<span
class='errorMsg'>Please enter something for email.</span>");

                                                    return;

                                            }
```

```
                                        $('#validationResult').html("verifying...");



                                        var emailVerifyApi = '//api.winpure.com/api/a/
v1?email=' + encodeURIComponent(emailText) + '&key=' + keyText;



                                        /*execute remote request to perform email
verification. Any errors will appear in the developer console (e.g. viewable using Chrome
developer tools)*/

                                        $.getJSON(emailVerifyApi, {})

                                                .done(function (data) {

                                                        reportResult(data);

                                                })

                                                .fail(function (jqxhr, textStatus,
error) {

                                                        var err = textStatus + ", " +
error;

                                                        console.log("Request failed: "
+ err);

                                                });;

                                });

                        });



                        /*Output result to the 'validationResult' div element*/

                        function reportResult(data) {

                                var status = data['status'].toLowerCase(); // get
'status' from REST response
```

```
                                    var additionalStatus = data['additionalStatus']; // get
'additionalStatus' from REST response

                                    var message = data['Message']; // if there is an error
(e.g. license issues), a notification will appear in the 'Message" from REST response.



                                    console.log(status);

                                    console.log(additionalStatus);

                                    console.log(message);



                                    var statusHtml;



                                    // if there is an error message, show here

                                    if (message != null

                                            && message != '') {

                                            statusHtml = "<span class='errorMsg'>Error.
Message='" + message + "' .</span>";

                                    } else {

                                            // map REST response data to presentation
messages.

                                            switch (status) {

                                                    case 'ok':

                                                            statusHtml = "<span
class='statusOk'>Email address is ok.</span>";

                                                            break;

                                                    case 'bad':
```

```
                                                    statusHtml = "<span
class='statusBad'>Email address is not valid.</span>";

                                                    break;

                                                default:

                                                    statusHtml = "<span
class='statusUnknown'>Unable to validate email. Reason=" + additionalStatus + "</span>";

                                                    break;

                                            }

                                    }



                                    console.log(statusHtml);



                                    // present the result on screen

                                    $('#validationResult').html(statusHtml);

                            }

            </script>

    </body>

    </html>
```

AngularJS (license key)

Note

Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1 using AngularJS with client side only calls.

```
<!DOCTYPE html>



<!--
```

```
*****************************************************************************************

*   File name:

*   v1_key_acl.html

*

*   Version:

*   1.0.20140827.0

*

*   Version control:

*   - 1.0.20140827.0 - initial release

*

*   Date:

*   August 2014

*

*   Description:

*   Demonstrates how to call a RESTful service @ //api.winpure.com/api/a/v1

*   using AngularJS with client side only calls.

*

*   This example requires a valid key to work correctly.

*

*   License:

*   Apache 2.0 (https://www.apache.org/licenses/LICENSE-2.0)

*****************************************************************************************

-->
```

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
        <meta charset="utf-8" />
        <title>emailverifyapp.com : License Key Sample.</title>
        <style type="text/css">
                .statusUnknown {
                        color: #c1c72c;
                }
                .statusOk {
                        color: #009933;
                }
                .statusBad, .errorMsg {
                        color: #ff0000;
                }
                input[type='text'],input[type='email'] {
                        width: 300px;
                }
                p label {
                        display: inline-block;
                        width: 60px;
                }
        </style>
```

```
    </head>

    <body>



            <h1>emailverifyapp.com : email verification demo using simple key
authentication with AngularJS.</h1>

            <h2>About</h2>

            <p>This example shows how to perform email verification using just client side
scripting and invoking a simple key based RESTful endpoint at https://api.winpure.com.</p>

            <h2>How to run this sample</h2>

            <p>This page can be hosted anywhere (i.e. any web host or platform). The only
thing needed is a valid license key.</p>

            <h2>Key features</h2>

            <ul>

                    <li>Compatible with all modern browsers</li>

                    <li>Uses AngularJS 1.2.23</li>

                    <li>No server side scripting required</li>

            </ul>

            <hr />

            <h2>Try it</h2>

            <div id="MyApp" ng-app="apiDemo" ng-controller="apiDemoController">

                    <form name="verifyForm" novalidate="" ng-submit="doVerify()">

                            <p>

                                    <label for="key">Key:</label>
```

```
                                    <input type="text" id="key" name="key" tabindex="1"
maxlength="20" ng-model="query.key" required="" />

                                    <span class="errorMsg" ng-show="verifyForm.key.
$error.required">*</span>

                            </p>

                            <p>

                                    <label for="email">Email:</label>

                                    <input type="email" name="email" id="email"
tabindex="2" ng-model="query.email" required="" />

                                    <span class="errorMsg" ng-show="verifyForm.email.
$error.required">*</span>

                                    <span class="errorMsg" ng-show="verifyForm.email.
$error.email">not valid email</span>

                            </p>

                            <p>

                                    <label> </label>

                                    <input type="submit" name="submit" id="submit"
tabindex="3" value="verify" />

                            </p>

                    </form>

                    <div id="validationResult"><!--Result output here-->

                            <div ng-show="showValidating">verifying..</div>

                            <div ng-show="showOk"><span class="statusOk">Email address is
ok.</span></div>

                            <div ng-show="showBad"><span class="statusBad">Email address is
not valid.</span></div>

                            <div ng-show="showUnknown"><span class="statusUnknown">Unable
to validate email. Reason={{additionalStatusMessage}}</span></div>
```

```
                        <div ng-show="showMessage"><span class="errorMsg">Error.
Message={{errorMessage}}</span></div>

                </div>

        </div>



        <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.2.23/
angular.min.js"></script>



        <script>



                // Module

                var app = angular.module('apiDemo', []);



                // Controller

                app.controller('apiDemoController', function apiDemoController($scope,
$http) {

                        $scope.query = {

                                key: "",

                                email: ""

                        };



                        $scope.result = {

                                status: "",

                                additionalStatus: ""

                        };
```

```
                        // verification event
                        $scope.doVerify = function () {
                                resetMessage();
                                $scope.showValidating = true;
                                var emailVerifyApi = '//api.winpure.com/api/a/v1?
email=' + encodeURIComponent($scope.query.email) + '&key=' + $scope.query.key;
                                console.log(emailVerifyApi);
                                $http.get(emailVerifyApi)
                                        .success(function (response) {
                                                resetMessage();
                                                var status =
response['status'].toLowerCase();
                                                var additionalStatus =
response['additionalStatus'];
                                                var message = response['Message'];

                                                console.log(status);
                                                console.log(additionalStatus);
                                                console.log(message);

                                                // if there is an error message, show
here
                                                if (message != null
                                                        && message != '') {
```

```
                                                $scope.errorMessage = message;
                                                $scope.showMessage = true;

                                        } else {
                                                // map REST response data to
presentation messages.
                                                switch (status) {
                                                        case 'ok':
                                                                $scope.showOk =
true;
                                                                break;
                                                        case 'bad':
                                                                $scope.showBad
= true;
                                                                break;
                                                        default:

$scope.additionalStatusMessage = additionalStatus;

$scope.showUnknown = true;

                                                                break;
                                                }
                                        }
                                });
```

```
                                //

                                function resetMessage() {

                                        $scope.showValidating = false;

                                        $scope.showBad = false;

                                        $scope.showMessage = false;

                                        $scope.showOk = false;

                                        $scope.showUnknown = false;

                                        $scope.showMessage = false;

                                }

                        }

                });



        </script>



</body>

</html>
```