



Safenames RESTful Web API

User Guide

Document Classification:	External Use
Document Ref.	
Version:	1.2
Dated:	24 January 2019
Document Author:	Mark Gilkes
Document Owner:	Jon Stock

Revision History

Version	Date	Revision Author	Summary of Changes
0.1	1 Feb 2016	Draft	
1.0	17 March 2016	Release	Approved for release
1.1	10 August 2018	Release	DomainInfo - added Status, OrderStatus, RenewStatus and ModifyDNSStatus. ClientAccount/{customerid} - added AutoRenewal, Profilenameservers & ProfileNSIPaddress
1.2	24 January 2019	Release	MoveDomain - added POST method. MoveDomain - added GET method Register - added GET method Register - Force TargetCustomerID. Added Local Presence validation.

Distribution

Name	Title
Mark Gilkes	Development Manger
Jon Stock	CTO

Approval

Name	Position	Signature	Date
Jon Stock	CTO		17 March 2016

Contents

1	INTRODUCTION.....	4
2	RESOURCE REFERENCE.....	5
2.1	SECURITY TOKEN.....	5
2.2	GET CLIENT ACCOUNT TREE	5
2.3	GET CLIENT ACCOUNT DETAILS	6
2.4	CREATE CLIENT SUB ACCOUNT	7
2.5	GET REGISTER REQUIREMENTS.....	8
2.6	REGISTER DOMAIN NAME.....	8
2.7	TRANSFER IN DOMAIN NAME	9
2.8	GET NAMESERVERS	9
2.9	MODIFY NAMESERVERS.....	10
2.10	GET DOMAIN RENEWALS	10
2.11	RENEW DOMAIN NAME	11
2.12	GET DOMAIN NAMES	11
2.13	GET DOMAIN INFO	12
2.14	GET DOMAIN AVAILABILITY	13
2.15	GET COUNTRY CODES.....	13
2.16	GET MOVE DOMAINS	14
2.17	MOVE DOMAIN	14
3	ERROR CODES	16
4	EXAMPLES	18
4.1	HTTP CLIENT - FIDDLER	18
4.2	C# SAMPLE CLIENT.....	22
4.3	PHP CLIENT EXAMPLE (WITHOUT CURL)	25
4.4	PHP CLIENT EXAMPLE (WITH CURL).....	28

1 Introduction

The Safenames Web API 2 is a RESTful web service built using ASP.NET Web API framework and implements an OAuth 2.0 Authorization server using OWIN middleware to provide token based authorization. To obtain an access token and gain access to the Safenames Web API a user must follow two steps;

1. Request a token from the OAuth server endpoint.
2. Send the token (as well as token_type) in the Authorization header for all subsequent requests.

e.g. Authorization: Bearer {access token}

For POST requests a Content-Type header must be sent to initiate content negotiation. i.e.

Content-Type: application/json (OR application/xml)

For all requests an Accept header can be sent to ensure the returned data format, again, this can be either json or xml;

Accept: application/json (OR application/xml)

Please note, the default Accept type is json.

All requests have common response properties, as well as any specific information that may be returned with each endpoint (if applicable). The common response properties are;

Code (integer)	1 (Success) -1 (Fail)
Message (string)	The response code followed by an error message.
Success (boolean)	True when response code = 1 False when response code = -1

An HTTP 200 response code will be returned when a request has been successful and an HTTP 400 Bad Request will be returned when an error is encountered. The client should then consult the error code and message.

An OTE endpoint is available for testing commands and code development, where no actual changes or domain registrations will occur. The OTE environment is reset monthly.

2 Resource Reference

2.1 Security Token

Request	Security Token
End point	https://api.safenames.com/token
OTE end point	https://dev-api.safenames.com/token
Method	POST
Headers	None
Parameters	grant_type=password&username= <i>your_username</i> &password= <i>your_password</i>
Response	<pre>{ "access_token": "<i>base64 encoded access token</i>", "token_type": "bearer", "expires_in": 86399 }</pre>

2.2 Get Client Account Tree

Request	Get Client Account Tree
End point	https://api.safenames.com/clientaccount
OTE end point	https://dev-api.safenames.com/clientaccount
Method	GET
Headers	Authorization Accept (optional)
Parameters	None
Response	<pre>{ "CompanyName": "<i>company name</i>", "CustomerID": <i>customer_id</i>, "ParentID": <i>parent_id</i> }</pre>

2.3 Get Client Account Details

Request	Get Client Account Details
End point	https://api.safenames.com/clientaccount/{customer_id}
OTE end point	https://dev-api.safenames.com/clientaccount/{customer_id}
Method	GET
Headers	Authorization Accept (optional)
Parameters	CustomerID
Response	<pre> { "CustomerID": (integer) "CompanyName": (string) "Registrant": (string) "ContactName": (string) "Address1": (string) "Address2": (string) "City": (string) "State": (string) "PostalCode": (string) "CountryCode": (string) "TelephoneNumber": (string) "FaxNumber": (string) "EmailAddress": (string) "AutoRenewal": (boolean) "ProfileNameserver1": (string) "ProfileNS1IPAddress": (string) "ProfileNameserver2": (string) "ProfileNS2IPAddress": (string) "ProfileNameserver3": (string) "ProfileNS3IPAddress": (string) "ProfileNameserver4": (string) "ProfileNS4IPAddress": (string) "ProfileNameserver5": (string) "ProfileNS5IPAddress": (string) "ProfileNameserver6": (string) "ProfileNS6IPAddress": (string) } </pre>

2.4 Create Client Sub Account

Request	Create Client Sub Account
End point	https://api.safenames.com/clientaccount
OTE end point	https://dev-api.safenames.com/clientaccount
Method	POST
Headers	Authorization Accept (optional)
Parameters	<pre>{ "AccountProfile" : { "OrgName": (string) * "ContactName": (string) * "Address1": (string) * "Address2": (string) "StateProvince": (string) * "City": "city" (string) * "CountryCode": (string) * (max length (2)) "PostalCode": (string) * "EmailAddress": (string) * "TelNo": (string) * "FaxNo": (string) }, "TargetCustomerID": (int) }</pre>
Response	<pre>{ "CustomerID": (integer) }</pre>
Additional Info	If the optional TargetCustomerID is omitted the sub-account will be created under the currently logged in account. Otherwise it will be created under the supplied target customer id.

2.5 Get Register Requirements

Request	Get Register Requirements		
End point	https://api.safenames.com/register		
OTE end point	https://dev-api.safenames.com/register		
Method	GET		
Headers	Authorization Accept (optional)		
Parameters	?DomainName= *		
Response	<pre>{ "CountryCode" : (string) "Country" : (string) "LPRequirement" : (string) "DefaultRegistrationTerm" : (integer) "MinRegistrationTerm" : (integer) "MaxRegistrationTerm" : (integer) }</pre>		
Additional Info	Local Presence Requirements	nr	no restrictions
		lc	local contact
		cf	company formation

2.6 Register Domain Name

Request	Register Domain Name		
End point	https://api.safenames.com/register		
OTE end point	https://dev-api.safenames.com/register		
Method	POST		
Headers	Authorization Accept (optional)		
Parameters	<pre>{ "DomainName": (string) * "RegistrationTerm": (integer) "TargetCustomerID": (integer) * }</pre>		
Response	None		

2.7 Transfer In Domain Name

Request	Transfer In Domain Name
End point	https://api.safenames.com/transferin
OTE end point	https://dev-api.safenames.com/transferin
Method	POST
Headers	Authorization Accept (optional)
Parameters	{ "DomainName": (<i>string</i>) * "AuthCode": (<i>string</i>) }
Response	None

2.8 Get Nameservers

Request	Get Nameservers
End point	https://api.safenames.com/nameservers
OTE end point	https://dev-api.safenames.com/nameservers
Method	GET
Headers	Authorization Accept (optional)
Parameters	?DomainName= *
Response	{ }

2.9 Modify Nameservers

Request	Modify Nameservers
End point	https://api.safenames.com/nameservers
OTE end point	https://dev-api.safenames.com/nameservers
Method	POST
Headers	Authorization Accept (optional)
Parameters	<pre>{ "DomainName": "domain_name", "UseIDPDNS": use_idp_dns, (Boolean) * "UseProfileDNS": use_profile_nameservers, (Boolean) * "Nameserver1": {"Name": (string) *, "Address": (string) *}, (Nameserver) * "Nameserver2": {"Name": (string) *, "Address": (string) *}, (Nameserver) * "Nameserver3": {"Name": (string) *, "Address": (string) *}, (Nameserver) "Nameserver4": {"Name": (string) *, "Address": (string) *}, (Nameserver) "Nameserver5": {"Name": (string) *, "Address": (string) *}, (Nameserver) "Nameserver6": {"Name": (string) *, "Address": (string) *}, (Nameserver) }</pre>
Response	None

2.10 Get Domain Renewals

Request	Get Domain Renewals
End point	https://api.safenames.com/domainrenewals
OTE end point	https://dev-api.safenames.com/domainrenewals
Method	GET
Headers	Authorization Accept (optional)
Parameters	?TargetCustomerID= ?FromDate= ?ToDate=
Response	<pre>{ "Registrant": (string) "DomainName": (string) "ExpiryDate": (datetime) "ExpiryDueDays": (integer) }</pre>

2.11 Renew Domain Name

Request	Renewal Domain Name		
End point	https://api.safenames.com/domainrenewals		
OTE end point	https://dev-api.safenames.com/domainrenewals		
Method	POST		
Headers	Authorization Accept (optional)		
Parameters	<pre>{ "DomainName": (string) * "Action": (integer) * "RenewalTerm": (integer) * }</pre>		
Response	<pre>{ "Registrant": (string) "DomainName": (string) "ExpiryDate": (datetime) "ExpiryDueDays": (integer) }</pre>		
Additional Info	Action values	1	Renew
		2	No not renew
		3	Restore & renew

2.12 Get Domain Names

Request	Get Domain Names		
End point	https://api.safenames.com/domainnames		
OTE end point	https://dev-api.safenames.com/domainnames		
Method	GET		
Headers	Authorization Accept (optional)		
Parameters	None		
Response	<pre>{ "DomainName": (string) "CustomerID": (int) }</pre>		

2.13 Get Domain Info

Request	Get Domain Info			
End point	https://api.safenames.com/domaininfo			
OTE end point	https://dev-api.safenames.com/domaininfo			
Method	GET			
Headers	Authorization Accept (optional)			
Parameters	?DomainName= (string) *			
Response	<pre>{ "DomainName": (string) "RegistrantName": (string) "ContactName": (string) "Address1": (string) "Address2": (string) "City": (string) "State": (string) "PostalCode": (string) "Country": (string) "TelephoneNumber": (string) "EmailAddress": (string) "Status": (integer) "OrderedStatus": (integer) "RegisteredDate": (string) "ExpiryDate": (datetime) "LastUpdateDate": (datetime) "RenewStatus": (integer) "ModifyDNSStatus": (integer) }</pre>			
Additional Info	Status	Declined	Declined at registry	
		Ordered	Please refer to OrderedStatus	
		Pre-Order	For NTLD domains where the TLD is in the pre-order phase	
		Pre-Transfer	Preparing DNS changes before Transfer	
		Registered	Completed Register/Transfer process	
		Reserved	Domain is reserved at registry	
		Transfer	Domain is being Transferred	
	OrderedStatus	0	Not in 'Ordered' status	
		1	Awaiting Customer Information	
		2	Safenames Processing	
3		Application at Registry		
RenewStatus	1	Renew requested		
	2	Do not renew requested		

	ModifyDNSStatus	0	Not requested
		1	In Progress
		2	Complete

2.14 Get Domain Availability

Request	Get Domain Availability
End point	https://api.safenames.com/domainavailability
OTE end point	https://dev-api.safenames.com/domainavailability
Method	GET
Headers	Authorization Accept (optional)
Parameters	?DomainNames=[&DomainNames=] (<i>string</i>) *
Response	{ "DomainName": (<i>string</i>) "status": (<i>string</i>) }

2.15 Get Country Codes

Request	Get Country Codes
End point	https://api.safenames.com/countrycodes
OTE end point	https://dev-api.safenames.com/countrycodes
Method	GET
Headers	Authorization Accept (optional)
Parameters	
Response	{ "Country": (<i>string</i>) "CountryCode": (<i>string</i>) }

2.16 Get Move Domains

Request	Get Move Domains
End point	https://api.safenames.com/movedomain
OTE end point	https://dev-api.safenames.com/movedomain
Method	GET
Headers	Authorization Accept (optional)
Parameters	?TargetCustomerID= (<i>int</i>)
Response	[{ "CustomerID" : (<i>int</i>) "DomainName" : (<i>string</i>) }]
Additional Info	If the TargetCustomerID is omitted then the resulting domain list will show all domains valid for moving under the currently logged-in client's account tree.

2.17 Move Domain

Request	Move Domain
End point	https://api.safenames.com/movedomain
OTE end point	https://dev-api.safenames.com/movedomain
Method	POST
Headers	Authorization Accept (optional)
Parameters	{ "DomainName" : (<i>string</i>) * "CustomerID" : (<i>int</i>) * "IRTPOptOut" : (<i>bool</i>) "ChangeRegistrant" : (<i>bool</i>) "TargetCustomerID" : (<i>int</i>) * }
Response	None

Additional Info	IRTP Opt Out (Inter-Registrar Transfer Policy) an opt in/out flag is required for generic domains. Please refer to the ICANN Inter-Registrar Transfer Policy
-----------------	--

3 Error Codes

Code	Message	Description
1000	Invalid request data	Malformed or missing values in request
1001	Domain name not found	The domain name supplied has not been found in your account.
1002	Invalid domain name	The format of the domain name supplied is invalid
1003	Invalid top level domain name	An invalid TLD has been supplied
1004	Invalid target customer id	The customer identifier supplied has not been found in your account structure
1005	Invalid country code	The country code supplied has not been recognised as an ISO 3166-1 alpha-2 country code
1006	Invalid registration term	The supplied registration term is invalid for the supplied top level domain name
1007	Domain name unavailable	The supplied domain name was found to be registered
1008	Invalid renewal term	The renewal term supplied is not valid for the top level domain name
1009	Invalid auth code	The supplied auth code is invalid for this transfer
1010	Domain is not pending transfer	The supplied domain name was found not to be in a transferable state
1011	At least 2 nameservers are required	An insufficient amount of nameservers has been supplied for a modify DNS request.
1012	Invalid name in Nameserver #	An invalid nameserver name or no value has been supplied
1013	Invalid IP address in Nameserver #	An invalid or no nameserver IP address has been supplied
1014	Transfer out already initiated	The domain name supplied is already in the transfer out status
1015	Invalid transfer out status	There is either a dispute lock or no paid transfer out invoice for the domain.
1016	Ivalid IPS tag	An IPS tag was expected or is invalid for this .uk domain
1017	{service} failed. Cashpot balance low	The service request failed due to insufficient funds
1018	Invalid renewal action	The renewal action is not valid for this domain name. Refer to DomainInfo.RenewStatus
1019	Not a valid domain status for modify nameservers	Modify nameserver requests are only valid for domains with a 'Registered' status
1020	IRTP property expected for TLD {tld}	You must supply a value for the Inter-registrar Transfer Policy setting which applies to the supplied TLD

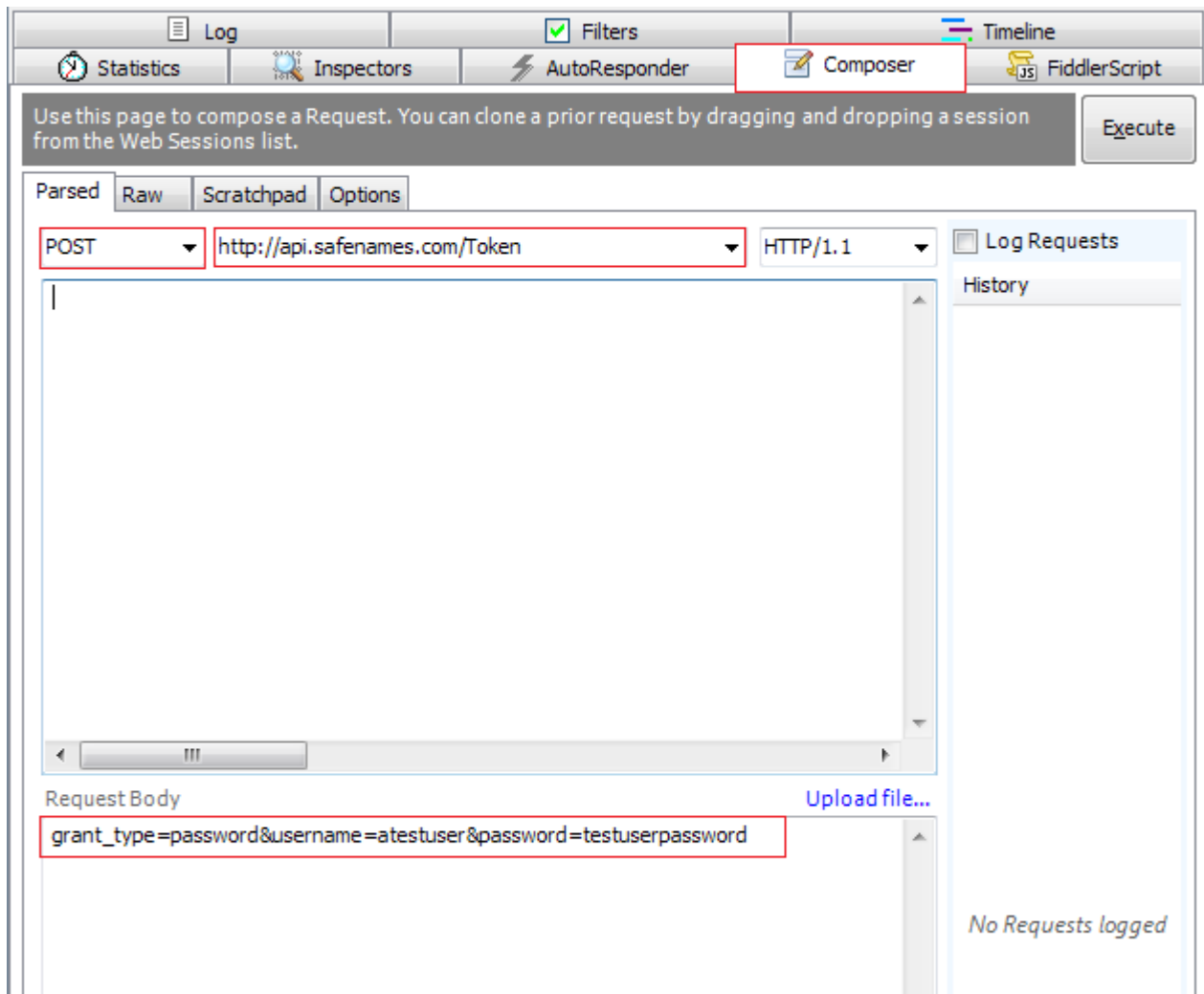
1021	Invalid move domain	The domain name supplied is either not found or not a valid domain for moving. Use GetMoveDomains to get a list of valid domain names.
1022	Nothing to move in request	The Target Client ID is the current Client ID

4 Examples

4.1 HTTP Client - Fiddler

You can test the API endpoints using your favourite HTTP client. The following example demonstrates acquiring an access token and requesting the country code list using Fiddler.

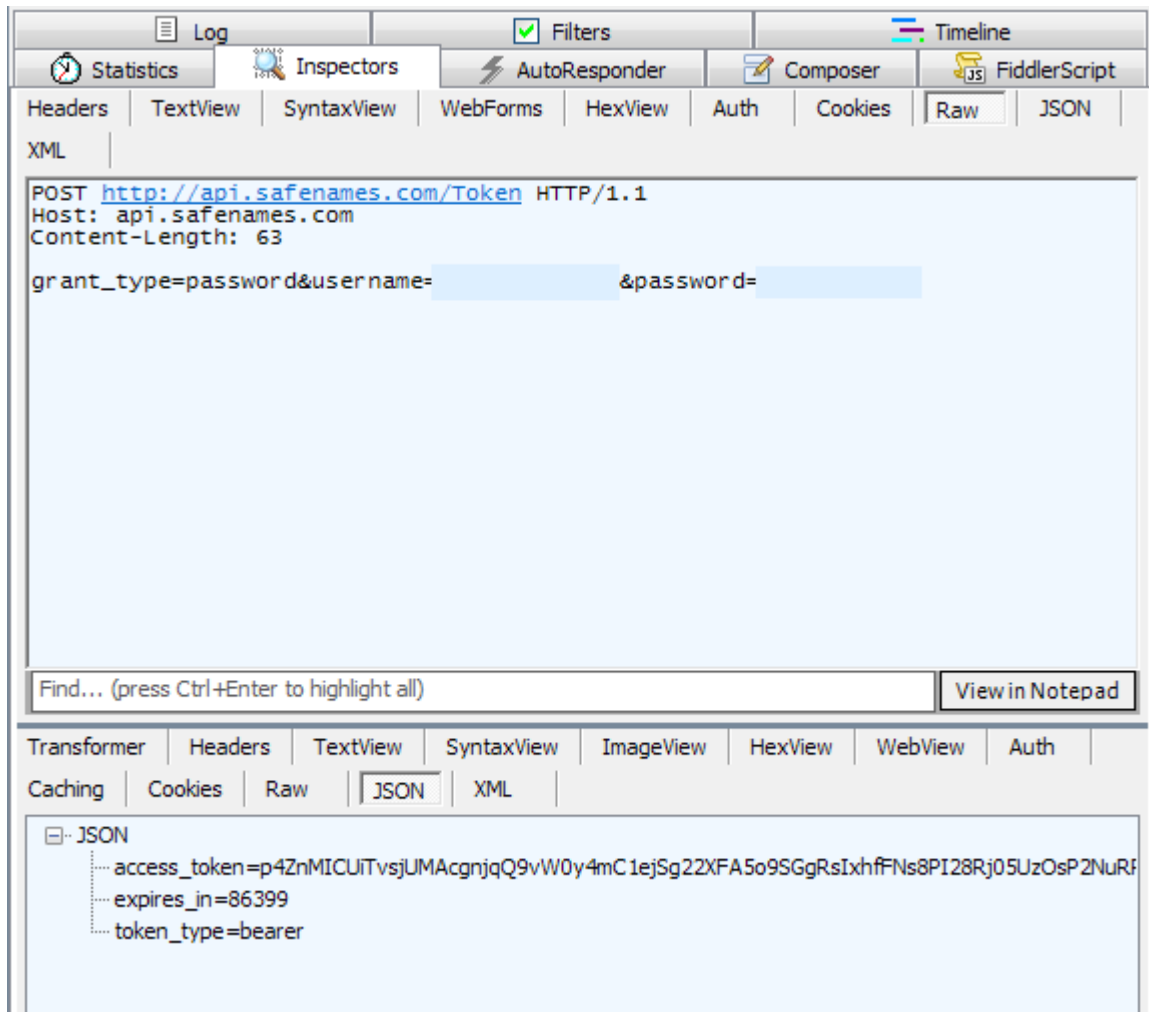
1. Open Fiddler and select the 'Composer' tab in the right part of the UI. Select the 'POST' request method and enter the URL for access token. In the request body enter the grant_type of password, your username and password as shown. Click the 'Execute' button.



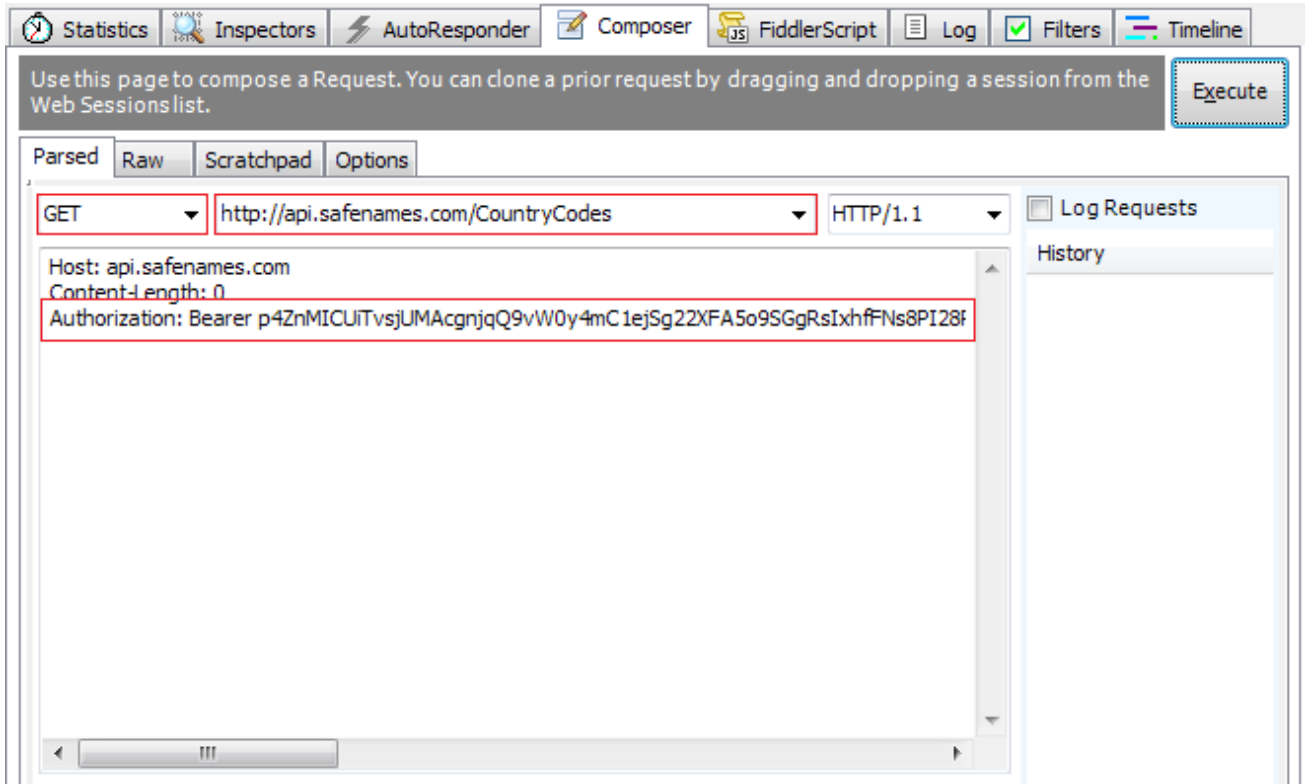
2. You should see a 200 response in the session list (left panel) as shown below. Double click the response and head over to the inspector window.

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
5	200	HTTP	api.safenames.com	/Token	295	no-cache; Expires: -1	application/json; charset=UTF-8

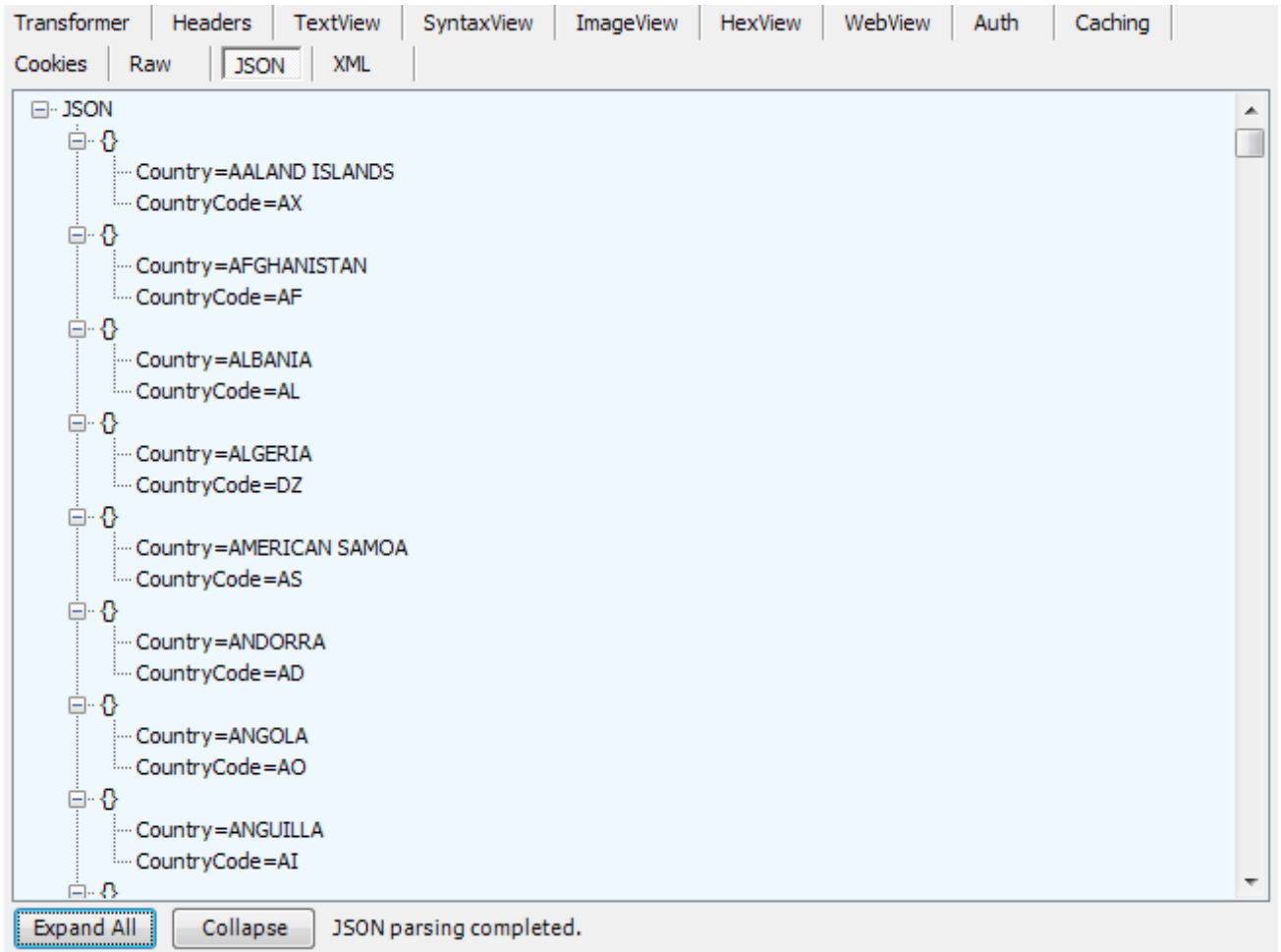
3. Right click the access_token response and copy the value to the clipboard. Click the 'Composer' tab.



4. Change the Url endpoint to CountryCodes and the method to 'GET'. Create an 'Authorization' header and paste the value of the access token as shown including the 'Bearer' authentication type. Click 'Execute'.



5. The response to the GET CountryCodes should be seen in the response pane as below;



4.2 C# Sample Client

The following console application code listing is in c# 5.0 and is the equivalent to the above calls to the Token and CountryCodes endpoints using Fiddler. The code snippet uses the new HttpClient found in the System.Net.Http namespace.

```
class Program
{
    static void Main(string[] args)
    {
        using (var httpClient = new HttpClient())
        {
            httpClient.BaseAddress = new Uri("http://dev-api.safenames.com");
            httpClient.DefaultRequestHeaders.Accept.Clear();
            httpClient.DefaultRequestHeaders.Accept.Add(new
                MediaTypeWithQualityHeaderValue("application/json"));

            string loginContent =
                String.Format("grant_type=password&username={0}&password={1}", "at
                    estuser", "testuserpassword");

            var content = new StringContent(loginContent);

            HttpResponseMessage response = httpClient.PostAsync("Token",
                content).Result;

            if (response.IsSuccessStatusCode) {

                var loginResp =
                    response.Content.ReadAsAsync<LoginResponse>().Result;

                // Add the Authorization token header
                httpClient.DefaultRequestHeaders.Authorization = new
                    AuthenticationHeaderValue("Bearer",
                    loginResp.AccessToken);

                response = httpClient.GetAsync("CountryCodes").Result;

                var countryCodeResponse =
                    response.Content.ReadAsAsync<List<CountryCodeResponse>>().
                    Result;
                countryCodeResponse.ForEach(r =>
                    Console.WriteLine("Country (Code) : {0} ({1})", r.Country,
                    r.CountryCode));

                Console.WriteLine();
                Console.WriteLine("Complete!");

            } else {

                var resp =
                    response.Content.ReadAsAsync<LoginErrorResponse>().Result;

                Console.WriteLine("Error : {0}", resp.Error);
                Console.WriteLine("Error Message : {0}",
                    resp.ErrorDescription);
            }
        }
    }
}
```

```
        Console.WriteLine();
        Console.WriteLine("Press any key to continue.");
        var input = Console.ReadKey();

    }

}

public class LoginResponse
{
    [JsonProperty("access_token")]
    public string AccessToken { get; set; }

    [JsonProperty("token_type")]
    public string TokenType { get; set; }

    [JsonProperty("expires_in")]
    public int ExpiresIn { get; set; }
}

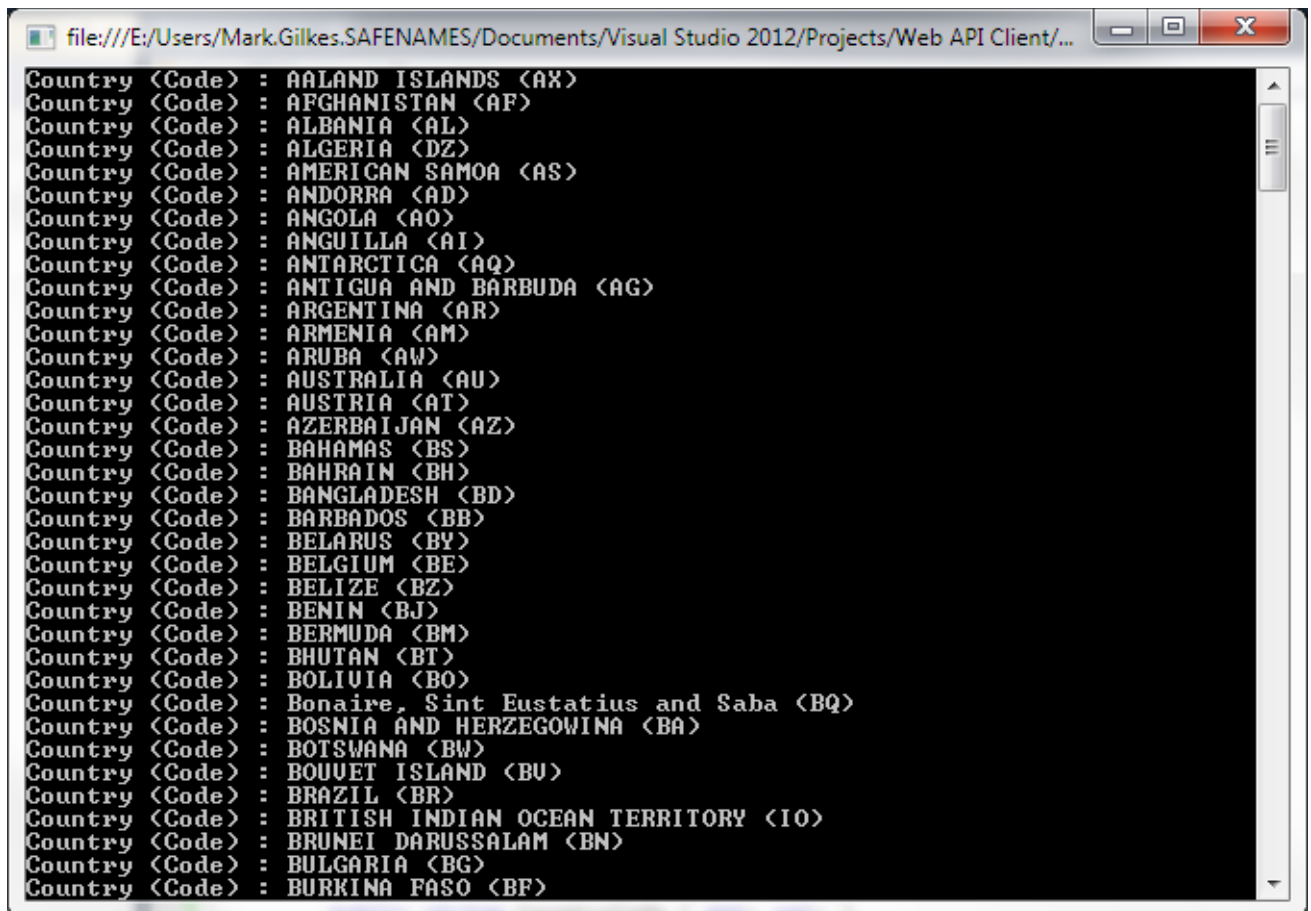
public class CountryCodeResponse
{
    [JsonProperty("Country")]
    public string Country { get; set; }

    [JsonProperty("CountryCode")]
    public string CountryCode { get; set; }
}

public class LoginErrorResponse
{
    [JsonProperty("error")]
    public string Error { get; set; }

    [JsonProperty("error_description")]
    public string ErrorDescription { get; set; }
}
```

Output;



```
file:///E:/Users/Mark.Gilkes.SAFENAMES/Documents/Visual Studio 2012/Projects/Web API Client/...
Country (Code) : AALAND ISLANDS (AX)
Country (Code) : AFGHANISTAN (AF)
Country (Code) : ALBANIA (AL)
Country (Code) : ALGERIA (DZ)
Country (Code) : AMERICAN SAMOA (AS)
Country (Code) : ANDORRA (AD)
Country (Code) : ANGOLA (AO)
Country (Code) : ANGUILLA (AI)
Country (Code) : ANTARCTICA (AQ)
Country (Code) : ANTIGUA AND BARBUDA (AG)
Country (Code) : ARGENTINA (AR)
Country (Code) : ARMENIA (AM)
Country (Code) : ARUBA (AW)
Country (Code) : AUSTRALIA (AU)
Country (Code) : AUSTRIA (AT)
Country (Code) : AZERBAIJAN (AZ)
Country (Code) : BAHAMAS (BS)
Country (Code) : BAHRAIN (BH)
Country (Code) : BANGLADESH (BD)
Country (Code) : BARBADOS (BB)
Country (Code) : BELARUS (BY)
Country (Code) : BELGIUM (BE)
Country (Code) : BELIZE (BZ)
Country (Code) : BENIN (BJ)
Country (Code) : BERMUDA (BM)
Country (Code) : BHUTAN (BT)
Country (Code) : BOLIVIA (BO)
Country (Code) : Bonaire, Sint Eustatius and Saba (BQ)
Country (Code) : BOSNIA AND HERZEGOWINA (BA)
Country (Code) : BOTSWANA (BW)
Country (Code) : BOUVET ISLAND (BU)
Country (Code) : BRAZIL (BR)
Country (Code) : BRITISH INDIAN OCEAN TERRITORY (IO)
Country (Code) : BRUNEI DARUSSALAM (BN)
Country (Code) : BULGARIA (BG)
Country (Code) : BURKINA FASO (BF)
```


4.3 PHP Client Example (without curl)

```
echo "PHP Client Test\r\n";
echo "-----\r\n";

///// print Token
// print_r(get_token_context());

country_code_request();

function country_code_request()
{
    $tokenContent = get_token_context();

    // Check if successfully logged in
    if($tokenContent->{"error"} == "invalid_grant")
    {
        echo $tokenContent->{"error"}.", ".$tokenContent-
        >{"error_description"}."\r\n";
        return ;
    }
    else if ($tokenContent->{"access_token"}=="")
    {
        echo "Invlaid token\r\n";
        return ;
    }

    $access_token = $tokenContent->{"access_token"};

    $datacontext = array('http' =>
        array(
            'method' => 'GET',
            'header' => "Content-Type: applicatrion/json\r\n".
            "Authorization: bearer ".$access_token."\r\n"
        )
    );

    $APIBaseURL = "https://dev-api.safenames.com";
    $APIAction = "/CountryCodes";
    $response = PostToApi($APIBaseURL,$APIAction,$datacontext);

    if($response === false)
    {
        // if the curl_exec() fails for some reason, it means it could not even
        reach the aWhere server
        // and the function returns FALSE
        echo 'cURL Transport Error (HTTP request failed)';
        die();
    }
    else
    {
        //transform the API response into a native PHP object
        $response = json_decode($response);

        if($response->{"Message"} == "Authorization has been denied for this
        request.")
```

```
        {
            echo $response->{"Message"};
            return;
        }
        else
        {
            while (list($key, $value) = each($response))
            {
                echo $value->{"CountryCode"}." : ".$value->{"Country"};
                echo "\r\n";
            }
        }
    }
}

function get_token_context()
{
    $user="{username}";
    $password="{password}";

    $APIBaseURL = "https://api.safenames.com";
    $APIAction = "/Token";
    $data = "grant_type=password&username=".$user."&password=".$password ;

    $datacontext = array ('http' =>
        array (
            'method' => 'GET',
            'header' => "Content-Type: application/json\r\n".
                "Authorization:
Basic".base64_encode("$user:$password")."\r\n",
            'content' => $data)
        );

    $response = PostToApi($APIBaseURL,$APIAction,$datacontext);

    $result = json_decode($response);

    return $result;
}

function PostToApi($baseURL,$action,$datacontext)
{
    $servicePath = $baseURL.$action;

    $context = stream_context_create($datacontext);
    $responseData = file_get_contents($servicePath, false, $context);

    if($responseData=="")
        echo "No response from ".$servicePath."\r\n";

    return $responseData;
}
```

Get-token output

The screenshot shows a PHP script in an IDE. The code includes a test echo statement, a separator line, and three functions: `country_code_request()`, `get_token_context()`, and `PostToApi($baseUrl, $action, $datacontext)`. The `get_token_context()` function is currently selected. The console output shows the result of the `get_token_context()` function call, which returns an `stdClass Object` with the following properties:

```

[access_token] => uDskZI-32CVwhzkzScX0yB08yad4ve5yhqoseAT9NyECqmxpEd82nv4EUVmA2hyqxtloNgvpoE8zDJYdfTSs4PwSuCktxUnYRy6VCYUgf-PY3gexB7K0xmo-frMS1kzI
[token_type] => bearer
[expires_in] => 86399
    
```

Get Country Code output

The screenshot shows the same PHP script as above, but with the `country_code_request()` function selected. The console output displays a list of country codes and their corresponding names:

```

TM : TURKMENISTAN
TC : TURKS AND CAICOS ISLANDS
TV : TUVALU
UG : UGANDA
UA : UKRAINE
AE : UNITED ARAB EMIRATES
GB : UNITED KINGDOM
US : UNITED STATES
UM : UNITED STATES MINOR OUTLYING ISLANDS
UY : URUGUAY
UZ : UZBEKISTAN
VU : VANUATU
VA : VATICAN CITY STATE (HOLY SEE)
VE : VENEZUELA
VN : VIET NAM
VG : VIRGIN ISLANDS (BRITISH)
VI : VIRGIN ISLANDS (U.S.)
WF : WALLIS AND FUTUNA ISLANDS
EH : WESTERN SAHARA
YE : YEMEN
ZM : ZAMBIA
ZW : ZIMBABWE
    
```

4.4 PHP Client Example (with curl)

```
<?php

get_token_context();

country_code_request();

function get_token_context()
{
    $user="{username}";
    $password="{password}";
    $data_string = "grant_type=password&username=".$user."&password=".$password ;

    $host = $GLOBALS['host'];
    $verb = $GLOBALS['verb'];

    $uri = "/Token";

    $ch = curl_init($host.$uri);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, $verb);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Content-Length: ' . strlen($data_string)
    ));
    curl_setopt($ch, CURLOPT_TIMEOUT, 5);
    curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 5);

    //execute post
    $result = curl_exec($ch);
    curl_close($ch);

    $result = json_decode($result);

    return $result;
}

function country_code_request()
{
    $host = $GLOBALS['host'];
    $verb = $GLOBALS['verb'];

    $tokenContent = get_token_context();

    //check if success logged in
    if($tokenContent->{"error"} == "invalid_grant")
    {
        echo $tokenContent->{"error"}.", ".$tokenContent
->{"error_description"}."<br>";
        return ;
    }
}
```

```
}
else if ($tokenContent->{"access_token"}=="")
{
    echo "Invlaid token<hr>";
    return ;
}

$uri = "/CountryCodes";
$access_token = $tokenContent->{"access_token"};
$headers = array("Authorization: Bearer $access_token");

// Set up the cURL request
$curl = curl_init($host.$uri);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
//curl_setopt($curl, CURLOPT_HEADER, true);

curl_setopt($curl, CURLOPT_TIMEOUT, 20);
curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 20);

// Normally you should not use these cURL options. They disable the SSL Cert
// verification. But many local development environments are not built with
// the standard chain authority certificates, and so cannot verify the Cert.
// If you have troubles making cURL requests, you can uncomment the next two
// lines, but don't put them into production.

curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);

// This fires the cURL request
$response = curl_exec($curl);
curl_close($curl);

if($response === false){
    // if the curl_exec() fails for some reason, it means it could not even
reach the aWhere server
    // and the function returns FALSE
    echo 'cURL Transport Error (HTTP request failed): '.curl_error($curl);
    die();
}
else
{

    //transform the API response into a native PHP object
$response = json_decode($response);

if($response->{"Message"} == "Authorization has been denied for this
request.")
{
    echo $response->{"Message"} ;
    return ;
}
else
{
```

```
while (list($key, $value) = each($response))
{
    echo $value->{"CountryCode"}." : ".$value->{"Country"};
    echo "<br>";
};
}
}
?>
```

Output

