

IVR Interface

Basic Request Guide	1
Basic Response Guide	2
Response Status Codes.....	2
Using XML instead of JSON	2
Code Example.....	3
Send Text to Speech message	8
Send Voice Message	10
Receive Notifications.....	11

Basic Request Guide

All functions in this document works in the same way.

The URL contains version and function names.

The authentication can be transmitted in the header in Base64, it can also be transmitted in the Body.

The Body contains the JOSN data.

Example – Authentication by Header:

URL

<https://capi.inforu.co.il/api/V2/FunctionName>

Header

content-type: application/json
Authorization: Basic USER:TOKEN (Encode Base64)

Body

```
{  
  "Data": {  
    Some Fields.....  
  }  
}
```

Example – Authentication by Body:

URL

<https://capi.inforu.co.il/api/V2/FunctionName?json=>

Body

```
{  
  "Auth": {  
    "Username": "USER",  
    "Token": "TOKEN"  
  }  
  "Data": {  
    Some Fields.....  
  }  
}
```

Basic Response Guide

```
{  
  "StatusId": 1,  
  "StatusDescription": "Success",  
  "DetailedDescription": "",  
  "FunctionName": "RelativeFunctionPath",  
  "RequestId": "lhdfsdkjfhdsdlhfsdk",  
  "Data": {...}  
}
```

Response Status Codes

In response to your request you will receive an Http code, as well as a StatusId & Status Description.

Note: the Status Description may change from time to time, hence we do not recommend basing logic on it, only on StatusId or Http code.

Http Code	StatusId	Status Description
200	1	OK
401	-1	General failure
401	-2	Authentication failed or illegal IP address
403	-221	User Not Active
403	-222	Locked Out
403	-2010	Max records for this method is 100
403	-2011	Max records for this method is 5000
403	-2014	RequestToken not found
403	-2015	Job in progress

Using XML instead of JSON

For any function you can use XML instead of JSON.

For documentation of xml use same link like:

<https://capi.inforu.co.il/api/v2/Voice/Message/Send?view=json>

with ext. view=xml, for exam

<https://capi.inforu.co.il/api/v2/Voice/Message/Send?view=xml>

Code Example

curl

```
--request POST \  
--url 'https://capi.inforu.co.il/api/v2/FunctionName' \  
--user 'USER:TOKEN' \  
--header 'content-type: application/json' \  
--data '{" "}'
```

PHP

<?php

```
$url = "https://capi.inforu.co.il/api/v2/Voice/Message/Send";
$username = "XXXXXXXXXX";
$password = "XXXXXXXXXX";
$request = array();
$request["Data"] = array();
$request["Data"]["List"] = array();
$request["Data"]["List"][] = array(
    "TextToSpeech"=>"Hello World",
    "Phone"=>"0526666666",
    "CustomerMessageID"=>"112354",
);
$json = json_encode($request);
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json; charset=utf-8',
));
curl_setopt($ch, CURLOPT_USERPWD, $username.":".$password);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $json );
$response = curl_exec($ch);
curl_close ($ch);

$result = json_decode($response,true);
print_r($result);

?>
```

C#

Add a reference to System.Web.Extensions (4.0)

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Web.Script.Serialization;

namespace SendVoice
{
    class Program
    {
        static void Main(string[] args)
        {
            string userName = "XXXXX";
            string token = "XXXXX";

            List<Message> MessagetList = new List<Message>();
            MessagetList.Add(new Message
            {
                TextToSpeech = "Hello World",
                Phone = "0589995551",
                CustomerMessageID = "9865998888"
            });

            string Response = SendTextToSpeech(userName, token, MessagetList);
        }

        public class Message
        {
            public string TextToSpeech { get; set; }
            public string Phone { get; set; }
            public string CustomerMessageID { get; set; }
        }

        public class requestBody
        {
            public Data Data { get; set; }
        }

        public class Data
        {
            public Message[] List { get; set; }
        }

        private static string SendTextToSpeech(string UserName, string Token,
            List<Message> MessagetList)
        {
            string responseBody = string.Empty;
            string Url = "https://capi.inforu.co.il/api/v2/Voice/Message/Send";

            requestBody re = new requestBody();
            re.Data = new Data();
            re.Data.List = MessagetList.ToArray();
        }
    }
}
```

```

var json = string.Empty;
try
{
    json = new JavaScriptSerializer().Serialize(re);
}
catch (Exception ex)
{
    responseBody = "Failed";
    return responseBody;
}

string requestBody = json.ToString();

//Setup the web request
HttpRequest request = (HttpRequest)WebRequest.Create(Url);
request.Timeout = 30000;
request.Method = "POST";
request.ContentType = "application/json";
// Set Authorization
string basicAuthorizationString = UserName + ":" + Token;
request.Headers.Add("Authorization", "Basic " +
System.Convert.ToBase64String(Encoding.UTF8.GetBytes(basicAuthorizationString)));
//Set the POST data in a buffer
byte[] postBuffer = Encoding.UTF8.GetBytes(requestBody);
//Specify the length of the buffer
request.ContentLength = postBuffer.Length;
//Open up a request stream
Stream requestStream = request.GetRequestStream();
//Write the POST data
requestStream.Write(postBuffer, 0, postBuffer.Length);
//Close the stream
requestStream.Close();
request.PreAuthenticate = true;
try
{
    //Create the Response object
    HttpResponse response;
    try
    {
        response = (HttpResponse)request.GetResponse();
    }
    catch (WebException e)
    {
        response = (HttpResponse)e.Response;
    }
    HttpStatusCode status = response.StatusCode;
    if (status != HttpStatusCode.OK)
    {
        // Raise error
    }

    //Create the reader for the response
    StreamReader sr = new StreamReader(response.GetResponseStream(),
Encoding.UTF8);
    //Read the response
    responseBody = sr.ReadToEnd();
    //Close the reader, and response
    sr.Close();
    response.Close();
}

```

```
    }  
    catch (Exception ex)  
    {  
        //Raise  
    }  
    return responseBody;  
}  
}
```

Send Text to Speech message

This function sends Text to Speech message.
You can send up to 100 items per request.

URL

<https://capi.inforu.co.il/api/v2/Voice/Message/Send>

Header

content-type: application/json
Authorization: Basic USER:TOKEN (Encode Base64)

Body (simple example)

```
{
  "Data": {
    "List": [
      {
        "TextToSpeech": "Hello World",
        "Phone": "0589995551",
        "CustomerMessageID": "9865998888"
      }
    ]
  }
}
```

Simple copy and advanced options

<https://capi.inforu.co.il/api/v2/Voice/Message/Send?view=json>

Parameters:

CampaignName – (string) Optional - The name of a new campaign (New Campaign created if you send a CampaignRefId that was not used before).

CampaignRefId – (string) Optional - Your Campaign ID for which the reports will be stored on the system. If this parameter is not sent, the reports will be staked on a single monthly campaign. (As if sent with CampaignRefId DefaultYYYYMM).

IgnorePossibleSendingTime – (bool) Ignore the sending limit that you set on the system.

TextToSpeech – (string) the message that you want to send.

Phone – The phone number to which you are sending the message.

CustomerMessageID – (string) Optional - Message ID on the client application. When confirmation on delivery is sent back to the client.

Response

```
{
  "StatusId": 1,
  "StatusDescription": "Success",
  "DetailedDescription": "",
  "FunctionName": "api/v2/Voice/Message/Send",
  "RequestId": "lhdfsdkjfhdsdlhfsdk",
  "Data": {
    "CampaignID": 194,
    "RequestToken": "sjhjsbbjujjnjbjb"
  }
}
```

Send Voice Message

This function sends Existing Campaign.
You can send up to 100 items per request.

URL

<https://capi.inforu.co.il/api/v2/Voice/Campaign/SendExisting>

Header

content-type: application/json
Authorization: Basic USER:TOKEN (Encode Base64)

Body (simple example)

```
{
  "Data": {
    "CampaignId": "183",
    "List": [
      {
        "Phone": "0589995551",
        "CustomerMessageID": "9865998888"
      }
    ]
  }
}
```

Simple copy and advanced options

<https://capi.inforu.co.il/api/v2/Voice/Campaign/SendExisting?view=json>

Parameters:

CampaignId – (string) Campaign ID in the system you want to send

Duplicate – (bool) Optional – if you want to duplicate the Campaign

DuplicateName – (string) if "Duplicate" = 1, the Name of the duplicated Campaign

IgnorePossibleSendingTime – (bool) Ignore the sending limit that you set on the system.

Phone – The phone number to which you are sending the message.

CustomerMessageID – (string) Optional - Message ID on the client application. When confirmation on delivery is sent back to the client.

Response

```
{
  "StatusId": 1,
  "StatusDescription": "Success",
  "DetailedDescription": "",
  "FunctionName": "api/v2/Voice/Campaign/SendExisting",
  "RequestId": "lhdfsdkjfhdsdlhfsdk",
  "Data": {
    "CampaignID": 194,
    "RequestToken": "sjhjsbbjujnjbjb"
  }
}
```

Receive Notifications

For receive notification on your system you need to configuration your URL in the settings.
The system do **POST** to your URL.

Example

```
{
  "CustomerId": 2,
  "ProjectId": 30253,
  "UserId": 30253,
  "Data": [ {
    "Channel": "IVR",
    "Type": "PhoneNumber",
    "Value": "0585007010",
    "EventName": "EndOfCall",
    "CampaignID": 217,
    "CustomerMessageID": "",
    "Status": "Answered",
    "ListeningDuration": "00:08",
    "Digit": "",
    "DateTime": "2019-04-30 15:27:07"
  }, {
    "Channel": "IVR",
    "Type": "PhoneNumber",
    "Value": "0732224528",
    "EventName": "EndOfCall",
    "CampaignID": 218,
    "CustomerMessageID": "",
    "Status": "Answered",
    "ListeningDuration": "00:06",
    "Digit": "",
    "DateTime": "2019-04-30 15:27:07"
  }
]
```