



Paycall Dynamic Dialer API

version: 2.1

Table of Content

Foreword.....	4
System Rule of Thumb.....	4
File Upload API.....	5
HTTP Path.....	5
Parameters.....	5
Curl CLI Example.....	5
Dialer API.....	6
HTTP Path.....	6
General API.....	6
Example for General API.....	9
IVR API.....	9
Example for IVR API.....	10
Steps API.....	10
Example for Steps API.....	14
Action API.....	16
Action.....	18
Action Types.....	18
Examples.....	21
Full JSON Example.....	21
Curl CLI example.....	23
Steps Report.....	23
Simple Dialer API.....	24
HTTP Path.....	24
General API.....	24
Example.....	27
CLI.....	27
simple_test.json.....	28
Upload Phone List API.....	29
HTTP Request.....	29
General API Structure.....	29
Examples.....	31
Number CSV file example.....	31
CLI.....	31
Start Campaign API.....	32
HTTP Request.....	32
Path Example.....	32
Examples.....	32
CLI.....	32
CDR By Campaign ID API.....	33

HTTP Request.....	33
HTTP URL Structure.....	33
HTTP Request Example.....	33
Return.....	33
CSV Fields.....	34
Examples.....	34
CLI.....	34
CSV Example.....	35
CDR By Number API.....	37
HTTP Request.....	37
Structure.....	37
HTTP Request Example.....	37
Return.....	37
CSV Fields.....	38
Examples.....	38
CLI.....	38
CSV Example.....	39
CDR hangup_cause statuses (for CDR API).....	40
E.164 Standard.....	41
How to read it?.....	41



Foreword

The following documentation provides an API to use with Paycall Dynamic Dialer API system.

The documentation explains how to upload recordings, create a campaign, upload phone numbers and also how to initiate call(s) to the campaign.

The documentation also includes how to retrieve CDR records on each given campaign.

The system can take either plain text (supporting Hebrew, English and Russian), or recordings to be played.

In order to use the API calls, it is mandatory to have user name and password that are provided for you by Paycall.

Lack of valid user name and password will result in authentication error.

System Rule of Thumb

- In order to use the API calls, it is mandatory to have user name and password that are provided for you by Paycall.
- Lack of valid user name and password will result in authentication error.
- All authentication are performed using HTTP Basic Authentication.
- You must escape the user name and password using URI escaping in order to use them properly.
- All messages that are returned by the system are provided as JSON unless specified otherwise at the document.
- When a key named error is returned, it is usually means that there was a problem with the content of the request.
- When a key named status is returned, it talks about the result of the action, even when there was an error.
- All API requests will be performed at the following HTTP address: `api.dialer.co.il`.

File Upload API

The following API is sub API call for Paycall Dynamic Dialer system, that allows the client to upload recordings to the server and play them at the IVR and actions.

The call is performed by POST request and allows to upload multiple files at a single request.

HTTP Path

```
http://api.dialer.co.il/v2/calls/upload
```

Parameters

In order to be able to upload a file the following parameter exists:

Parameter	Note
recs[]	The following parameter must always be used with brackets as part of it's name, and all given files to be uploaded will use the same name.

Curl CLI Example

```
$ curl -X POST \  
> -F "recs[]=@cembalo-6.wav" \  
> -F "recs[]=@hello_welcome.wav" \  
> http://user:password@api.dialer.co.il/v2/calls/upload
```

The following command does the following:

- Use POST request.
- Send the file cembalo-6.wav using the recs[] field.
- Send the file hello_welcome.wav using the recs[] field.
- Connect to the following address using the user name and password:
http://user:password@api.dialer.co.il/v2/calls/upload

Dialer API

The following API request creates a complex dialer rules that allow dynamic IVR to be defined by the client.

The request is the way to create a campaign with it's rules.

The API is using POST request, and a JSON document as the body.

The document breaks the API into sub sections to help with better understanding of each API requirement.

HTTP Path

`http://<username>:<password>@api.dialer.co.il/v2/calls/dialer`

General API

The following API features will have basic features:

Field	Description
schedule_type	<p>Determine the type of schedule to use. Optional Can be "seconds" (default), or "timestamp".</p> <p>When set to seconds, the schedule filed will be handled for the number of seconds from "now".</p> <p>When set to timestamp, it will use Unix timestamp and it must be in the future.</p> <p>Note:</p> <p>If not set, then the schedule field will be handled as seconds to wait.</p>
schedule	<p>When is the call going to start, based on schedule_type. Optional When schedule_type is set to seconds, it will wait Amount</p>

Field	Description
	<p>of seconds until it's time to initiate the call.</p> <p>When <code>schedule_type</code> is set to <code>timestamp</code>, then a time in the feature must be provided using Unix timestamp. A time in the past, or current time will trigger an exception.</p> <p>Default, if not provided, is <code>null</code>.</p> <p>Note:</p> <p>If not set, then the initiation will be immediately.</p>
<code>unique_id</code>	<p>The request unique id, that will be returned as-is. The <code>unique_id</code> should arrive from the client, for identifying it's request.</p> <p>If none provided, a custom id will be returned.</p>
<code>return_address</code>	<p>To whom the TTS system going to report back when everything else is done.</p> <p>Note:</p> <p>If no address was provided, then an error will be returned.</p>
<code>max_retries</code>	<p>How many retries to perform on each question on an Invalid/Timeout. Can be override on each action if required.</p> <p>Minimum: 0 Maximum: 10000 Default, if non provided is 3.</p>
<code>ivr</code>	An object that changes based on the IVR section .
<code>ring_timeout</code>	<p>The maximum time in seconds to wait for a call to be answered, before moving to the next destination.</p> <p>Default: 10 seconds</p>
<code>retry_wait_time</code>	<p>The number of seconds to wait until the next retry.</p> <p>Default will be 0.</p>
<code>numbers_from_queue</code>	string - "true" or "false". Optional

Field	Description								
	<p>If set to "true", then the phone numbers will be taken from the campaign queue.</p> <p>If set to "false", the number field will be looked for, with a phone number to call.</p> <p>Default: "false"</p>								
number	<p>The phone number to call to. Applicable only if the field numbers_from_queue is set to "false". When that happen an E.164 phone number is supported.</p> <p>Note:</p> <ul style="list-style-type: none"> • If the phone number is required, it cannot be empty, or invalid, An error will be sent back. • Supports E.164 format <p>E164 Format:</p> <table border="1"> <tr> <td>+97272000000</td> <td>- Israeli number</td> </tr> <tr> <td>97272000000</td> <td>- Israeli number</td> </tr> <tr> <td>+14155552671</td> <td>- USA number</td> </tr> <tr> <td>14155552671</td> <td>- USA number</td> </tr> </table>	+97272000000	- Israeli number	97272000000	- Israeli number	+14155552671	- USA number	14155552671	- USA number
+97272000000	- Israeli number								
97272000000	- Israeli number								
+14155552671	- USA number								
14155552671	- USA number								
cli	<p>The caller id to place for the call. Default, if not provided will be Hidden number.</p>								
dtmf_originator	<p>Allow to send DTMF when the dialed number (from the number field) was answered. Optional</p> <p>If set, use as a string, with integer keys 0123456789, and #* inside.</p> <p>If there is a need to wait in order to send the DTMF, use the</p>								

Field	Description
	<p>W or w chars.</p> <ul style="list-style-type: none"> • W - Wait one second • w - Wait half a second <p>Notes:</p> <ul style="list-style-type: none"> • If the field is empty or not provided, no DTMF will be sent. • If invalid chars will provide, they will be ignored.

Example for General API

```
{
  "schedule": 300,
  "unique_id": "bf0c60b2-c29a-11e5-9265-fcaa1417781c",
  "return_address": "http://service.example.com/foo",
  "max_retries": 5,
  "ring_timeout": 15,
  "retry_wait_time": 0,
  "number": "0300000000",
  "ivr": {
    ...
  }
}
```

The following example does not set a cli, and there for it is a hidden caller id, and ask the dialer to wait 300 seconds (five minutes) before initiating. There will be five tries to connect the phone number, and if failed, it will not wait and try again.

IVR API

The following section describe the IVR JSON object.

The section have the following fields:

Field	Description
count_steps	<p>Number of steps that the API is going to deal with.</p> <p>Minimum: 1</p>
have_dependency	<p>Boolean string ("true" "false") if there are steps that depends on each other.</p> <p>If this field is false, each step will go one after the other regardless what was pressed.</p> <p>If the field is true, the action field will determine what to do next, and only the first element at the 'steps' array will be used.</p> <p>Default, if non provided is "false"</p> <p>Note: If set to false, then step by step actions will happen, and goto will be ignored.</p>
steps	<p>An array of objects that holds all of the steps.</p> <p>Note: If the number of steps is different then count_steps, then the IVR will stop at the minimal steps provided.</p>

Example for IVR API

```
{
  "count_steps": 1,
  "have_dependency": "false",
  "steps": [
    {
      ...
    }
  ]
}
```

Steps API

The following section describe the steps JSON object.

The section have the following fields:

Field	Description
step_id	A unique id for the step. It will be used in order to go into it, if required, as a rule.
gender	A string that can be either male or female. Default if not provided: female.
prompt_is_file	string of "true" or "false". - Optional If set to true, we look for existed file, and if not existed reporting error. If set to false, the system will use Text To Speech instead. Default: "false"
prompt	Prompt supports three types of values: <ul style="list-style-type: none"> • Simple string - The text will be set as it was written. • Array of strings - An array of strings, will make the usage of multiple texts, with concatenation for each of the text based on the array. • An Object - A way to set place holders for changing existed text. <p>If the prompt is an object, it will support place holders like so:</p> <pre>{ "text": "Hello \$0\$, welcome to \$1\$. \$0\$, we would like to remind you that your appointment to \$2\$ will take place at \$3\$", "fields": 4, "0" : "Mr. Filnston", "1" : "stone age", "2": "driving license", "3": "01/01/-2016 BC"</pre>

Field	Description
	<p>}</p> <p>The numbers are the text to use at the place holders, while dollar number, and dollar sign again are the place holders.</p> <p>If no index for place holder (e.g. \$4\$ will be in string, but no index to change it, then the string will contain \$4\$).</p> <p>Note: Beside of "text" and "fields" fields, every other key is considered as a placeholder.</p> <p>It is possible to use a name based place holders instead of numeric.</p>
dtmf_ivr	<p>a value or an array of value for what keys can be valid: Allowed Values: 0123456789*#.</p> <p>If there is more then one char next to each other, for example: 10, it will wait for 1 and 0 chars to be together.</p> <p>If both 1, and 0 expected as a single key, it should be written as follows: "dtmf_ivr": ["0", "1"]</p> <p>If there is only a single key that is allowed to be pressed, it should be written as follows: "dtmf_ivr": "0"</p>
dtmf_ivr_timeout	<p>How much time to wait to a key press.</p> <p>Default, if none provided will be 3 seconds.</p>
dtmf_invalid_text_is_file	<p>string of true or false. - Optional</p> <p>If set to true, we look for existed file, and if not existed reporting error.</p> <p>If set to false, the system will use Text To Speech</p>

Field	Description
	instead. Default: false
dtmf_ivr_invalid_text	What to play when an invalid DTMF was pressed. If empty or non provided, it will just repeat the same IVR.
dtmf_ivr_retries	Number of retries. If not exists, the global field of max_retries will take place instead.
actions	<p>The action field is an array of objects built as follows</p> <pre>[{ "key": "1", "actions": [{ "action": "hangup" }] }, { "key": "2", "actions": [{ "action": "play", "params": { "text": "Thank you for answering" } }, { "action": "wait", "params": { "sleep": 0.5 } }, { "action": "hangup" }] }, { "key": "0", "actions": [{ "action": "play", "params": { "text": "We are going to give you the next question" } }, { "action": "goto", "params": { "step_id": "q10" } }] }]</pre>

Field	Description
	<pre>] } 1]</pre> <p>if actions is null, then the operation will end.</p> <p>Details regarding the actions field.</p> <p>Note: If the action ends without anything else, the flow will stop, and the call will hang up.</p>

Example for Steps API

```
"steps": [
  {
    "step_id": "foo",
    "gender": "female",
    "prompt": "לחץ על אד בשביל להגיע ליעד הבא, אחד עשרה בשביל לנגן, ",
    "dtmf_ivr": ["1", "11", "0", "4"],
    "dtmf_ivr_timeout": 1,
    "dtmf_ivr_invalid_text": "המקש שנלחץ אינו נתמך",
    "dtmf_ivr_retries": 10,
    "actions": [
      {
        "key": "11",
        "actions": [
          {"action": "play",
            "params": { "text": "תודה שלחצת על אחד עשרה",
              "gender": "male"}},
          {"action": "wait", "params": {"sleep": 0.5}},
          {"action": "hangup"}
        ]
      },
      {
        "key": "1",
        "actions": [
          { "action": "goto", "params": {"step_id": "bar"}}
        ]
      },
      {
        "key": "4",
```

```

        "actions": [
            {
                "action": "wait", "params": {"sleep": 100000}
            }
        ],
        {
            "key": "0",
            "actions": [
                {"action": "play",
                 "params": {"text": "...מצאת את המקש הסודי"}},
                {"action": "wait", "params": {"sleep": 3.5}},
                {"action": "hangup"}
            ]
        }
    ],
    {
        "step_id": "bar",
        "gender": "male",
        "prompt": "האם תעקוב אחר הארנב הלבן? 1 - כן, 2 - לא",
        "dtmf_ivr": ["1", "2"],
        "dtmf_ivr_timeout": 1,
        "dtmf_ivr_invalid_text": "המקש אינו חוקי כאן",
        "dtmf_ivr_retries": 10,
        "actions": [
            {
                "key": "1",
                "actions": [
                    { "action": "call",
                     "params": {
                         "number": "05000000000",
                         "cli": "0300000000"
                     }
                    }
                ]
            },
            {
                "key": "2",
                "actions": [
                    { "action": "hangup" }
                ]
            }
        ]
    }
]

```

The following example show how to use the steps section to create a simple IVR.

If the [have_dependency](#) field is set to true, the goto field will work, and go to the new section.

Action API

The actions API have the following fields:

Field	Description
key	<p>Holds the DTMF key to use. The parameter is a string.</p> <p>As of version 2, the parameter can also be an object of two elements.</p> <ul style="list-style-type: none"> min - The smallest DTMF to capture (that is part of the range) max - The biggest DTMF to capture (that is part of the range). <p>Notes:</p> <ul style="list-style-type: none"> You can extract sub DTMF as another action object, even when on range, and perform a different action for it, but the min-element and max-elements should not include it. Range <u>must be numeric</u>, it cannot use the # or * chars. <p>Using key as a range:</p> <pre>{ "key": { "min": "10", "max": "12" }, "actions": [{ "action": "play", "params": { "text": "You are at the teen ages, access denied" } }] }, { "key": "13", "actions": [{</pre>

Field	Description
	<pre> "action": "play", "params": { "text": "Happy Bar Mitzva" } }], }, { "key": { "min": "14", "max": "17" }, "actions": [{ "action": "play", "params": { "text": "You are at the teen ages, access denied" } }] }, { "key": { "min": "18", "max": "67" }, "actions": [{ "action": "play", "params": { "text": "All work, and no play, makes Jack ..." } }] }, { "key": { "min": "68", "max": "99" }, "actions": [{ "action": "play", "params": { "text": "Welcome to the 3rd age Mr."} }] } } </pre>
actions	The actions parameters takes an array of objects, where there is one mandatory field named action, and the other parameters changes based on the action type, located at the action section .

Action

The following section provides information about each action that is supported by the Paycall dynamic dialer.

There are several types of actions that can be used with the IVR. Each action can have their own sub set of parameters, and therefor, the parameters will be listed at their own section.

If an action have parameters, it will be inside a field named params.

Action Types

Action	Description
hangup	Hangup the call. No extra parameter are required.
call	Perform a new call that will be bridged.
goto	Go to a new step . Note: When have_dependency is false, this command is ignored.
mark	Mark the dtmf as part of the actions.
play	Play a text or a recording
wait	To wait for x seconds and then continue.

call action

The call action can initiate a new call to a given phone number.

The action has the following parameters that are required to be used at the params field:

Parameter	Description
number	The phone number to call
cli	The caller id to display

Parameter	Description
dtmf_keys	<p>Keys to press on answer - Optional</p> <p>If set, use as a string, with integer keys 0123456789, and #* inside.</p> <p>If there is a need to wait in order to send the DTMF, use the w or W chars.</p> <ul style="list-style-type: none"> • W - Wait one second • w - Wait half a second

goto action

The goto action has one parameter that will go into the params field.

Parameter	Description
step_id	<p>The new step to go to based on it's step_id field.</p> <p>Note: If have_dependency is false, the goto action will be ignored.</p>

mark action

The mark action takes two parameters that will go into the params field.

Parameter	Description
type	<p>The type can be one of the following:</p> <ul style="list-style-type: none"> • remove • custom <p>The type will be passed as part of the return value.</p> <p>Note: When remove was choose, the number will not be allowed to be reused on that list again.</p>
value	Always a string, and only applicable if custom was chosen as

Parameter	Description
	<p>the type.</p> <p>The value will be returned as part of the report for that step.</p>

play action

The play action takes the following parameters that will go into the params field:

Parameter	Description
text	<p>The text to play as a sound file.</p> <p>Note: If play_is_recording is set to "true", then text will have a file name to play instead.</p>
gender	<p>The gender to use (male female) - Optional</p> <p>Default: female.</p>
is_recording	<p>string of "true" or "false". - Optional</p> <p>If set to true, we look for existed file, and if not existed reporting error. If set to false, the system will use Text To Speech instead.</p> <p>Default: false</p>

wait action

The wait takes one parameter of a numeric value that go into the params field.

Parameter	Description
sleep	<p>Numeric value (floating point or integer) that represents seconds.</p> <p>Note: Floating point is rounded up (using ceil).</p>

Examples

Full JSON Example

test.json.

```
{
  "schedule": 0,
  "unique_id": "f096c975-caca-4cd4-9819-1cef62237479",
  "return_address": "http://127.0.0.1/",
  "max_retries": 3,
  "ring_timeout": 15,
  "retry_wait_time": 10,
  "number": "05460000000",
  "cli": "0300000000",
  "ivr": {
    "count_steps": 2,
    "have_dependency": "true",
    "steps": [
      {
        "step_id": "foo",
        "gender": "female",
        "prompt": "לחץ על אחד בשביל להגיע ליעד הבא, אחד עשרה בשביל לנגן",
        "dtmf_ivr": ["1", "11", "0", "4"],
        "dtmf_ivr_timeout": 1,
        "dtmf_ivr_invalid_text": "המקש שנלחץ אינו נתמך",
        "dtmf_ivr_retries": 10,
        "actions": [
          {
            "key": "11",
            "actions": [
              {
                "action": "play",
                "params": {
                  "text": "תודה שלחצת על אחד עשרה",
                  "gender": "male"
                }
              },
              {
                "action": "wait",
                "params": {
                  "sleep": 0.5
                }
              },
              {
                "action": "hangup"
              }
            ]
          },
          {
            "key": "1",
            "actions": [
              {
                "action": "goto",
                "params": {
                  "step_id": "bar"
                }
              }
            ]
          }
        ]
      },
      {
        "key": "4",
```

```

        "actions": [
            {
                "action": "wait", "params": {"sleep": 10}
            }
        ],
        {
            "key": "0",
            "actions": [
                {"action": "play",
                 "params": {"text": "... מצאת את המקש הסודי"}},
                {"action": "wait", "params": {"sleep": 3.5}},
                {"action": "hangup"}
            ]
        }
    ],
    {
        "step_id": "bar",
        "gender": "male",
        "prompt": "האם תעקוב אחר הארנב הלבן? 1 - כן, 2 - לא",
        "dtmf_ivr": ["1", "2"],
        "dtmf_ivr_timeout": 1,
        "dtmf_ivr_invalid_text": "המקש אינו חוקי כאן",
        "dtmf_ivr_retries": 10,
        "actions": [
            {
                "key": "1",
                "actions": [
                    { "action": "call",
                     "params": {
                         "number": "0530000000",
                         "cli": "0540000000",
                         "dtmf_keys": "WWW111113"
                     }
                    }
                ]
            },
            {
                "key": "2",
                "actions": [
                    { "action": "hangup" }
                ]
            }
        ]
    }
]

```

```
}  
}
```

Curl CLI example

```
$ curl -X POST -d @test.json \  
> http://user:password@api.dialer.co.il/v2/calls/dialer
```

Steps Report

The steps report is the returned JSON based on what the user has pressed.

```
{  
  "unique_id": "f096c975-caca-4cd4-9819-1cef62237479",  
  "steps": [  
    {  
      "step_name": "foo",  
      "dtmf": "0"  
    }  
  ]  
}
```

Simple Dialer API

The following dialer provide a way to create a simple campaign where a phone number can be taken from a JSON API or the [file upload](#) API, and play a text or a recording to the callee, allowing them after that call to have hangup or the system will call to another phone number that will be connected to the callee.

In order to start the campaign there is a require to use the [Start Campaign](#) API.

HTTP Path

```
http://<username>:<password>@api.dialer.co.il/v2/calls/simple_dialer
```

General API

The following API features will have basic features:

Field	Description
unique_id	A unique id for the step. It will be used in order to go into it, if required, as a rule.
dial_at	A Unix Epoch timestamp. Optional If set to 0 or not present, then when Start Campaign API is triggered, the campaign will initiate ASAP. If present with proper Epoch in the feature, the Start Campaign API will initiate the call when the proper time is due. If the timestamp is in the past, an error will be returned.
from_campaign	A string that can be "true" or "false". When the field is "true", the phone numbers will be taken from the Upload Phone List API.
leg_a	The phone number to call to, using E.164 standard.

Field	Description
	<p>If "from_campaign" is "true", the field is ignored. If "from_campaign" is "false", and the field is missing, an error will be returned.</p> <p>E.164 Format:</p> <div style="border: 1px dotted black; padding: 5px;"> +97272000000 - Israeli number 97272000000 - Israeli number +14155552671 - USA number 14155552671 - USA number </div>
cid_a	<p>Caller id for the "leg_a" field. Optional</p> <p>The phone number to display when calling to "leg_a".</p> <p>Notes:</p> <ul style="list-style-type: none"> • If empty or not sent, a hidden call will be sent. • If "from_campaign" is "true", the field is ignored.
dtmf_a	<p>Send DTMF to the call of "leg_a". Optional</p> <p>If set, use as a string, with integer keys 0123456789, and #* inside.</p> <p>If there is a need to wait in order to send the DTMF, use the W or w chars.</p> <ul style="list-style-type: none"> • W - Wait one second • w - Wait half a second <p>Notes:</p> <ul style="list-style-type: none"> • If "from_campaign" is "true", the field is ignored. • The DTMF is sent in the background and is not blocking. • If the field is empty or not provided, no DTMF will be sent. • If invalid chars will provide, they will be ignored.
tts	<p>A string field of "true" and "false".</p>

Field	Description
	<p>If “true”, then expect from the “sound” field to be a free text.</p> <p>If “false”, it must be pointed to a file that was uploaded using the File Upload API.</p> <p>Note:</p> <ul style="list-style-type: none"> If the file that was set at the field was not uploaded successfully, an error will be returned.
sound	<p>If “tts” field is set to “true”, then the system take a “free text” and will send it as a Text To Speech.</p> <p>If “tts” field is set to “false”, then the field must hold a file name uploaded by the File Upload API.</p>
gender	<p>Set the “sound” field is set to be a Text To Speech (“tts” is “true”), “gender” field make it to a be “male” or “female”. Optional</p> <p>Possible values are “male” or “female”. Default, if not set, is “female”.</p>
answer_delay	<p>Integer/String – Set the number of seconds to wait until the audio is sent to the callee. Optional</p> <p>Default if not set, is 1 second.</p>
ring_timeout	<p>Integer/String – Set the number of seconds for a ring before stopping. Optional</p> <p>Default if not set, is 10 seconds.</p> <p>Note:</p> <ul style="list-style-type: none"> GSM based Cell phones move to an Answering machine after 30 seconds if no answer.
max_retries	<p>Integer/String – Set number of retries to “leg_a” if the call failed. Optional</p> <p>Default if not set, is 1 time.</p>
retry_wait_time	<p>Integer/String – Set the number of seconds to wait between retries, to contact “leg_a”. Optional</p>

Field	Description
	Default if not set, is 3 seconds.
leg_b	An E.164 phone number to connect after the “ sound ” field finished playing. Optional If not set, then the call will hangup after the “ sound ” field finished playing.
cid_b	Caller ID for “ leg_b ” field, if set. Optional If “ leg_b ” is set, and “cid_b” is not, the a hidden caller id will be sent.
dtmf_b	DTMF to send to “ leg_b ”. Optional If set, use as a string, with integer keys 0123456789, and #* inside. If there is a need to wait in order to send the DTMF, use the W or w chars. <ul style="list-style-type: none"> • W - Wait one second • w - Wait half a second Notes: <ul style="list-style-type: none"> • The DTMF is sent in the background and is not blocking. • If the field is empty or not provided, no DTMF will be sent. • If invalid chars will provide, they will be ignored.

Example

CLI

```
$ curl -X POST -d @simple_test.json \
```

```
> http://user:password@api.dialer.co.il/v2/calls/simple_dialer
```

simple_test.json

```
{  
  "unique_id": "4aa2f0be-97c1-44a2-b68d-b3601f7374f8",  
  "from_campaign": "false",  
  "leg_a": "97231234567",  
  "cid_a": "0501234567",  
  "dtmf_a": "",  
  "tts": "true",  
  "gender": "male",  
  "sound": "This is a test",  
  "answer_delay": 1,  
  "leg_b": "972501234567",  
  "cid_b": "",  
  "dtmf_b": "97231234567",  
  "dial_at": "",  
  "max_retries": 3,  
  "retry_wait_time": 3,  
  "ring_timeout": 15  
}
```

Upload Phone List API

The following API is aimed to upload a list of phone numbers using a comma (,) file, where each line ends with either \r\n – (CR+LF – ASCII 13 and ASCII 10), or just \n (LF – ASCII 10).

The aim of the following API structure is to allow and create a campaign using Paycall's Dynamic Dialer.

The API request requires an HTTP POST using HTTP Authentication.

The POST must have a parameter named numbers.

HTTP Request

In order to create a request, the following usage will be used:

```
/v2/calls/upload_phones/{unique_id}
```

The {unique_id} field should be changed by the unique id for the campaign.

For example:

```
/v2/calls/upload_phones/bf0c60b2-c29a-11e5-9265-fcaa1417781c
```

General API Structure

The file itself must be a text file using a comma char (,) for delimitation.

Each line must end with the sequence of \r\n aka as CR+LF (ASCII 10 ASCII 13) or \n aka LF (ASCII 13).

A file can be compressed using one of the following formats, and a proper mime type must be send for them:

Format	Mime type	Description
.csv	text/csv	The default format to use
.txt	text/plain	A text file, but that is comma-delimited
.zip	application/zip	A compressed zip file, containing only a single

Format	Mime type	Description
		file to be parsed. Note: The first file that will be found at the zip archive, will be the one that parsed, the others will be ignored.
.gz	application/gzip	A compressed file using the gzip format. Note: Tarball (.tar.gz and tgz) files are <u>not supported</u> .

The number lines has the following structure:

Parameter	Description
number	<p>E.164 phone number structure:</p> <p>Valid E.164 structure example:</p> <pre>+97231000000 97231000000</pre> <p>Note: An Invalid E.164 will raise an exception.</p>
cli	<p>The caller id to place when dialing this number. Optional The cli field must be using a phone number.</p> <p>Note: If not set, then the default caller id will be used instead.</p>
dtmf_originator	<p>The DTMF to use after an answer was made. Optional</p> <p>If set, use as a string, with integer keys 0123456789, and #* inside.</p> <p>If there is a need to wait in order to send the DTMF, use the W or w chars.</p> <ul style="list-style-type: none"> W - Wait one second w - Wait half a second



Parameter	Description
	Note: <ul style="list-style-type: none">• If the field is empty or not provided. The default settings of dtmf_originator will be used instead.• If invalid chars will provide, they will be ignored.

Important Note:

It is a good advise to keep empty fields at the csv, so it will be easier to parse it.

Examples

Number CSV file example

```
+972540000000  
972723000000,0720000000  
+14155552671,4155551111,w1#113  
972212345678,,1#2
```

CLI

```
curl -X POST \  
> -F "numbers=@upload_numbers1.csv" \  
> http://<username>:<password>@api.dialer.co.il/v2/calls/upload_phones
```



Start Campaign API

The following command initiate the actual call after the phone numbers (if required) have been uploaded, and the [dialer](#) command has been created.

The command will start the call, or place it on hold until the [schedule](#) time from the dialer command has arrived.

The request must be sent using HTTP POST request.

HTTP Request

The command takes one argument as part of the URL, and that's the unique campaign id:

```
/v2/calls/start_campaign/{unique_id}
```

The {unique_id} is a place holder for the campaign unique id.

Path Example

```
/v2/calls/start_campaign/2743976e-6030-455a-821f-8d671463d3ba
```

Examples

CLI

```
curl -X POST \  
>"http://<username>:<password>@api.dialer.co.il/v2/calls/start_campaign/  
2743976e-6030-455a-821f-8d671463d3ba"
```


CDR By Campaign ID API

CDR stands for Call Details Records. The CDR reports contains information about a call.

The following command provides a list of all available CDR records for a specific campaign.

The request is sent using GET request.

HTTP Request

In order to perform the HTTP request there is a need for one dynamic parameters as part of the URL:

1. campaign unique id

HTTP URL Structure

```
/v2/calls/cdr_by_campaign/{unique_id}
```

The {unique_id} placeholder is the unique id for the campaign that the CDR should be reported.

HTTP Request Example

```
/v2/calls/cdr_by_campaign/2743976e-6030-455a-821f-8d671463d3ba
```

Return

A CSV file separated using a comma (,) and surrounding with double quotes (") char, and ending with \r\n (CR+LF – ASCII 13 and ASCII 10) end of line.

CSV Fields

Field	Description
cli	The caller id that was used in order to dial the call
phone_number	The phone number that was dialed
start_time	When was the call started (the actual dial started)
answer_time	When was the call answered (Or empty if there was no answer)
hangup_time	When was the call hangup up
duration	How much time was the call, from start time until hangup time.
billsec	How much time has passed from answer time until hangup time
hangup_cause	ITU Q.850 based hangup causes
unique_campaign_id	The campaign unique id
steps	JSON based Steps that was set by the callee.

Examples

CLI

```
curl \
>"http://<username>:<password>@api.dialer.co.il/v2/calls/cdr_by_campaign
/2743976e-6030-455a-821f-8d671463d3ba"
```



CSV Example

```
cli,phone_number,start_time,answer_time,hangup_time,duration,billsec,hangup_cause,unique_campaign_id,steps
"031234567","972501234567","2016-03-30 14:53:50+03","2016-03-30 14:53:55+03","2016-03-30
14:54:06+03","16","11","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
"031234567","972501234568","2016-03-30 15:10:41+03","2016-03-30 15:10:46+03","2016-03-30
15:10:54+03","13","8","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],,"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
"031234567","972501234569","2016-03-30 15:14:28+03","2016-03-30 15:14:33+03","2016-03-30
15:14:42+03","14","9","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],,"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
"031234567","972501234560","2016-03-30 15:17:03+03","2016-03-30 15:17:08+03","2016-03-30
15:17:12+03","9","4","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
"031234567","972501234561","2016-03-30 15:18:25+03","2016-03-30 15:18:30+03","2016-03-30
15:18:34+03","9","4","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
"031234567","972501234562","2016-03-30 16:44:16+03","2016-03-30 16:44:22+03","2016-03-30
16:44:32+03","16","10","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
```



```
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
"031234567","972501234563","2016-03-30 16:44:48+03","2016-03-30 16:44:57+03","2016-03-30
16:45:29+03","41","32","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
"031234567","972501234564","2016-03-30 16:46:30+03","2016-03-30 16:46:35+03","2016-03-30
16:46:44+03","14","9","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
"031234567","972501234565","2016-03-30 16:51:03+03","2016-03-30 16:51:09+03","2016-03-30
16:51:23+03","20","14","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
"031234567","972501234566","2016-03-30 16:52:37+03","2016-03-30 16:52:47+03","2016-03-30
16:52:55+03","18","8","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
"031234567","972501234570","2016-03-30 16:57:20+03","2016-03-30 16:57:30+03","2016-03-30
16:57:39+03","19","9","NORMAL_CLEARING","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"},"invalid":[],"unique_id":"2743976e-6030-455a-821f-8d671463d3ba"}"
```

CDR By Number API

CDR stands for Call Details Records. The CDR reports contains information about a call.

The following command provides a single CDR line for a specific phone number based on a campaign id.

The request is sent using GET request.

HTTP Request

In order to perform the HTTP request there is a need for two dynamic parameters as part of the URL:

1. campaign unique id
2. phone number

Structure

```
/v2/calls/cdr_by_number/{unique_id}/{phone_number}
```

The {unique_id} placeholder need to be replaced with the campaign unique id.

The {phone_number} placeholder need to be replaced with the phone number to look for.

HTTP Request Example

```
/v2/calls/cdr_by_number/2743976e-6030-455a-821f-8d671463d3ba/972501234567
```

Return

A CSV file separated using a comma (,) and surrounding with double quotes (") char, and ending with \r\n (CR+LF – ASCII 13 and ASCII 10) end of line.

CSV Fields

Field	Description
cli	The caller id that was used in order to dial the call
phone_number	The phone number that was dialed
start_time	When was the call started (the actual dial started)
answer_time	When was the call answered (Or empty if there was no answer)
hangup_time	When was the call hangup up
duration	How much time was the call, from start time until hangup time.
billsec	How much time has passed from answer time until hangup time
hangup_cause	ITU Q.850 based hangup causes
unique_campaign_id	The campaign unique id
steps	JSON based Steps that was set by the callee.

Examples

CLI

```
curl \
>"http://<username>:<password>@api.dialer.co.il/v2/calls/cdr_by_number/2
743976e-6030-455a-821f-8d671463d3ba/972501234567"
```



CSV Example

```
cli,phone_number,start_time,answer_time,hangup_time,duration,billsec,hangup_cause,unique_campaign_id,steps
"972501234567","2016-03-30 15:17:03+03","2016-03-30 15:17:08+03","2016-03-30
15:17:12+03","9","4","NORMAL_CLEARING","031234567","2743976e-6030-455a-821f-8d671463d3ba","{"steps":
[{"dtmf":"12","step_name":"foo"}],"invalid":[],"unique_id":"2743976e-6030-455a-821f-
8d671463d3ba"}"
```

CDR hangup_cause statuses (for CDR API)

The following section describes the common hangup_cause that can appear. The hangup_cause is using the [ITU Q.850](#) naming convention.

hangup_cause	Description
RECOVERY_ON_TIMER_EXPIRE	Timeout to connect phone carrier, or the call has arrived to timeout by the carrier.
NORMAL_TEMPORARY_FAILURE	The end carrier is not available for a long period of time
USER_BUSY	The far end was busy
ALLOTTED_TIMEOUT	The entire call had worked properly, but arrived to fixed timeout set by Paycall.
NO_USER_RESPONSE	The callee did not answer in any way to the call - No signal was provided for it.
NORMAL_CLEARING	The call was answered, and finished properly
NO_ANSWER	The callee did not answer the call
CALL_REJECTED	The callee choose to press on CANCEL when call had arrived
NO_ROUTE_TRANSIT_NET	The phone number was not found (same as 404 for web pages)
NO_ROUTE_DESTINATION	
UNALLOCATED_NUMBER	

E.164 Standard

The E.164 standard is a simple standard by the ITU, that explains how international phone numbers should be represented.

The following API supports the following number structure:

```
[+]<country code><phone number>
```

How to read it?

- [] represents optional value, in this case the plus (“+”) sign.
- <> represents required fields:
 - Country Code – The ITU’s international country prefix, such as “972” for Israel, or “1” for the U.S.
 - Phone number – The actual phone number to dial. If a country have an area prefix, then it should arrive to that number as well.

Example for such prefix (they are both the same regarding the API):

```
+97272000000 - Israeli number  
14155552671 - USA number
```

Note:

Both numbers can add or remove the “+” sign, and it will be valid by the API. The country prefix does not matter regarding the sign itself.