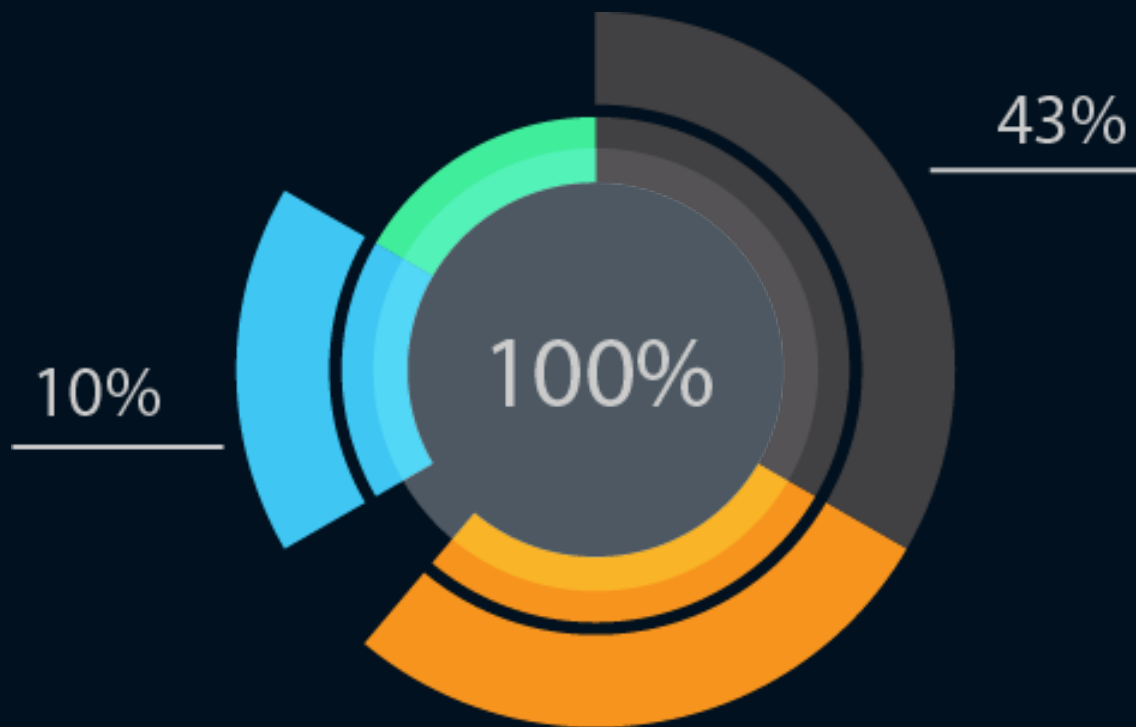


ZARBAT TESTER

SETUP & RUN

2021

Version 0.7.0



This document outlines the writing, testing, and result analysis processes using it

AVAYA CPaaS API services

Zarbat Tester

Dear Teammate,

Thank you for considering "Zarbat Tester" to help you in the process of writing and coding tests for the InboundXML elements and the REST API CPaaS services. This framework aims to develop end-to-end tests over CPaaS features in the various environments where AVAYA OneCloud CPaaS is installed. Through a continuous system's diagnosis, its mission is to monitor health and guarantee the quality of services with an easy to write, code and report testing platform.

The biggest and nicest motivations

Knowledge sharing: Due to the union among stakeholders, developers and testers to test and validate APIs and API-driven applications.

Synergy: one group transmits knowledge to the other and to make the team technically cohesive leading to coincident interests.

Communication between teams: Generally, in software engineering companies, it is difficult to see developers and testers working together, however, BDD encourages communication between teams.

Sincerely

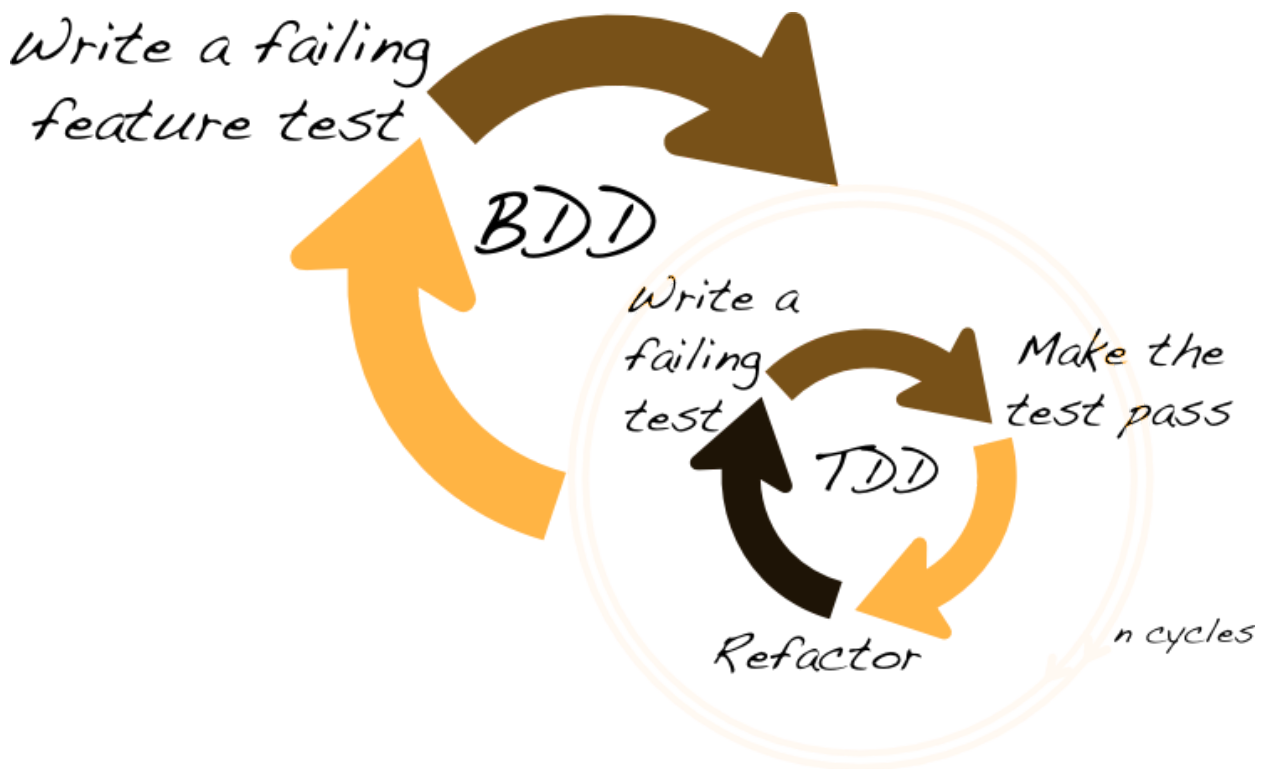


About “BDD”

In the first decade of 2000's, BDD emerged from Test Driven Development (TDD), being an agile development technique that aimed to integrate business rules with a programming language, focusing on the software **behavior**.

The paradigm “Code First and Test Later” took the place of “Test First and Code Later”.

BDD uses domain specific languages (DSL) to transform sentences into executable tests.



About “Gherkin Language”

Basically, in this syntax, scenarios are described using *current state markers* (Given), *state change triggers* (When) and *new state* (Then).

The Figure below exemplifies the use of the words Given – When – Then

```
features > purchasenumber > ☰ purchasenumber.feature
1  Feature: Purchase a Number
2      In order to list available numbers from one area code
3      As an end user
4      I want to list at least one number from this area
5
6      Scenario: List available numbers and buy one
7
8          Given my test setup runs
9          When I list all available numbers
10         Then I should get to buy 1 from list
11
```

Gherkin was created in 2008 and is a Business Readable, Domain Specific Language. This allows non-developer stakeholders to use the language too. So the Gherkin language allows the behavior of the software to be described without detailing how it will be implemented, the language is not important, what matters is the value delivered to customers.

Gherkin's keywords are: Feature, Scenario, Background, Scenario Outline and Examples. The word **Feature** is used to document the name of the feature that will be described. The word **Scenario** is the keyword used to create the description of a scenario. After this keyword comes the scenario name, which identifies what will be described. **Background** is used to describe prerequisite/context situations for all scenarios in the description file. **Scenario Outline** allows the use of variables in scenarios so that examples can be used to test the described resource. These examples are described in table format, after defining the scenario under the **Examples** keyword. The scenarios are described using the **Given – When – Then** User Story pattern

About “CTLang”

CTLang stands for “CPaaS Test Language” and is the set of sentences used to define the steps necessary to perform an end-to-end test of the AVAYA OneCloud CPaaS service API.

```
ctx.Step(`^Append To "([^\"]*)" config hang up$`, AppendToConfigHangUp)
ctx.Step(`^"([^\"]*)" configured as conference "([^\"]*)" with size (\d+)$`, ConfiguredAsConferenceWithSize)
ctx.Step(`^"([^\"]*)" configured to dial "([^\"]*)"`$, ConfiguredToDial)
ctx.Step(`^"([^\"]*)" configured to dial and send digits "([^\"]*)" to "([^\"]*)"`$,
ConfiguredToDialAndSendDigitsTo)
ctx.Step(`^"([^\"]*)" configured to gather speech$`, ConfiguredToGatherSpeech)
ctx.Step(`^"([^\"]*)" configured to gather digits until "([^\"]*)"`$, ConfiguredToGatherDigitsUntil)
ctx.Step(`^"([^\"]*)" configured to pause (\d+) seconds$`, ConfiguredToPauseSeconds)
ctx.Step(`^"([^\"]*)" configured to ping URL$`, ConfiguredToPingURL)
ctx.Step(`^"([^\"]*)" configured to play last recording$`, ConfiguredToPlayLastRecording)
ctx.Step(`^"([^\"]*)" configured to play tone "([^\"]*)"`$, ConfiguredToPlayTone)
ctx.Step(`^"([^\"]*)" configured to record calls$`, ConfiguredToRecordCalls)
ctx.Step(`^"([^\"]*)" configured to record calls for download$`, ConfiguredToRecordCallsForDownload)
ctx.Step(`^"([^\"]*)" configured to say "([^\"]*)"`$, ConfiguredToSay)
ctx.Step(`^"([^\"]*)" configured to send SMS "([^\"]*)" to "([^\"]*)"`$, ConfiguredToSendSMSTo)
ctx.Step(`^"([^\"]*)" should get last call duration more than or equals to (\d+)$`,
ShouldGetLastCallDurationMoreThanOrEqualsTo)
ctx.Step(`^"([^\"]*)" configured to redirect to ping URL$`, ConfiguredToRedirectToPingURL)
ctx.Step(`^"([^\"]*)" configured to reject call$`, ConfiguredToRejectCall)
ctx.Step(`^"([^\"]*)" configured to hang up after (\d+) seconds$`, ConfiguredToHangUpAfterSeconds)
ctx.Step(`^I list all available numbers$`, IListAllAvailableNumbers)
ctx.Step(`^I should get to buy (\d+) from list$`, IShouldGetToBuyFromList)
ctx.Step(`^I make a call from "([^\"]*)" to "([^\"]*)"`$, IMakeACallFromTo)
ctx.Step(`^my test setup runs$`, MyTestSetupRuns)
ctx.Step(`^"([^\"]*)" should be enter conference "([^\"]*)"`$, ShouldBeEnterConference)
ctx.Step(`^"([^\"]*)" should be able to listen to frequencies "([^\"]*)"`$,
ShouldBeAblaybleToListenToFrequencies)
ctx.Step(`^"([^\"]*)" should be able to view the SMS "([^\"]*)"`$, ShouldBeAbleToViewTheSMS)
ctx.Step(`^"([^\"]*)" should be reset$`, ShouldBeReset)
ctx.Step(`^"([^\"]*)" should get a ping request on the URL$`, ShouldGetAPingRequestOnTheURL)
ctx.Step(`^"([^\"]*)" should get call cancel status$`, ShouldGetCallCancelStatus)
ctx.Step(`^"([^\"]*)" should get digits "([^\"]*)" from "([^\"]*)"`$, ShouldGetDigitsFrom)
ctx.Step(`^"([^\"]*)" should get speech "([^\"]*)"`$, ShouldGetSpeech)
ctx.Step(`^"([^\"]*)" should get the incoming call from "([^\"]*)"`$, ShouldGetTheIncomingCallFrom)
ctx.Step(`^"([^\"]*)" should get transcription "([^\"]*)"`$, ShouldGetTranscription)
ctx.Step(`^"([^\"]*)" configured to send MMS "([^\"]*)" and media "([^\"]*)" to
"([^\"]*)"`$,ConfiguredToSendMMSAndMediaTo)
ctx.Step(`^"([^\"]*)" should be able to view the MMS "([^\"]*)" with media "([^\"]*)"`$,
ShouldBeAbleToViewTheMMSWithMedia)
```

Prerequisites

ngrok

To rapidly construct End-to-End Functional Testing that continuously validates all critical aspects of complex transactions—which may extend through gRPC, CORBA, WSDL, WebAssembly, Telnet, Socket, databases, and everything in between, you should have ngrok installed on your machine to provide callback services and keep a conversation with OneCloud during the test until the needed and final assertion is reached.

Download the following latest version: <https://ngrok.com/download>.

Add to "Path" System Variable

On Windows:

Click on the Advanced system settings link and then click Environment Variables. Under the section System Variables, select the environment variable "Path", and click Edit. Add a new entry and put: "C:\ngrok".

Run in your terminal: `ngrok http 5000`

And get the dns address provided to set in your config.ini file.

Run & Install

There are 3 ways to run & install ZARBAT.

1) Docker

To run your test you should pull from Docker Hub and have the folders "config" and "test" created. Below the command needed to pull an image or a repository from the registry.

```
docker pull masaruohashi/zarbat_tester:zarbat_tester
```

Then, set your config.ini and test.ctlang as shown below, and execute:

```
docker run -v test:/app/test -v config:/app/config -p 5000:5000 -it zarbat_tester -config  
config/config.ini -l log/zarbat.log -level info -n 3 -test test/[your_test].feature -url  
[your_public_ip] -port 5000
```

Folders

Please check these the existence of the following folders:

/temp

To allow the "Zarbat Tester" use it to create temporary files needed to adapt godog framework to run more than one Feature per time and parse a single file with ctlang extension.

```
▼ temp\~ctlang973888891
  ≡ ~test.Gather.028764830.feature
  ≡ ~test.Hangup.331644773.feature
  ≡ ~test.Number.552128384.feature
  ≡ ~test.Pause.913785567.feature
  ≡ ~test.Ping.216609458.feature
  ≡ ~test.Play.252384361.feature
  ≡ ~test.PlayLastRecording.959775924.feature
  ≡ ~test.Record.487512195.feature
  ≡ ~test.Redirect.683283462.feature
  ≡ ~test.Reject.092804525.feature
  ≡ ~test.Say.211645224.feature
  ≡ ~test.SMS.217396327.feature
```

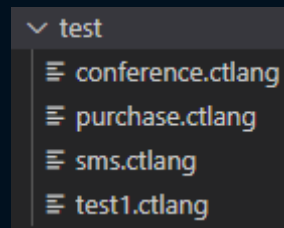
/config

This folder is needed to place your context files containing the conditions and parameters to execute your tests properly. With parametrization it is possible to better analyse errors triggered in certain conditions reported by customers.

```
1  ApiUrl="https://api.zang.io"
2  BaseUrl="http://mohashi.ngrok.io"
3  ApiVersion="v2"
4  # Have to change to 34 bytes beggining with AC to be considered valid.
5  AccountSid="AC777c3e3262352901b0e24b7092f6eef0"
6  # Have to change to 32 bytes to be considered valid.
7  AuthToken="c59b774e18b44769bdc8b4253a7ee0d7"
8  NumberD="+12048003047"
9  NumberDSid="DI777c3e32697ba80d5a4f4082bd4dcff5"
10 NumberC="+12048003029"
11 NumberCSid="DI777c3e3270908f97680344b5a3fea522"
12 NumberA="+12048003030"
13 NumberASid="DI777c3e32d3e6623dbbe740ea93fdebb4"
14 NumberB="+12048003021"
15 NumberBSid="DI777c3e32f4fd09f0fb5b4894ae41ca6d"
16 NumberE="+12048003028"
17 NumberESid="DI777c3e3265711172eae445bc926c72c7"
18 NumberBR1="+5561984385415"
19 NumberBR2="+5561992574040"
20 From=""
21 To=""
22 ToSid=""
23 ActionUrl="http://zang.io/ivr/welcome/call"
24 StatusCallback = ""
25
```


/test

In this folder you should put your written ctlang file with Gherkin syntax.



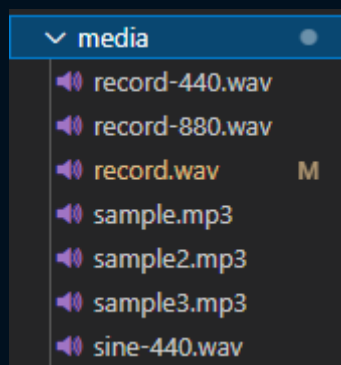
/log

In this folder, “Zarbat Tester” puts your logging files: one for INFO level and the other for DEBUG level.

```
670 2021/08/03 14:23:11 INFO: *****
671 2021/08/03 14:23:11 INFO: START OF TEST SUITE
672 2021/08/03 14:23:11 INFO: Config: config/config.ini
673 2021/08/03 14:23:11 INFO: Number of Tries: 3
674 2021/08/03 14:23:11 INFO: Log: log/zarbat.log
675 2021/08/03 14:23:11 INFO: Tests: [conference][gather][hangup][number][pause][ping][play][playlastrecording][
676 2021/08/03 14:23:11 INFO: .....
677 2021/08/03 14:23:27 INFO: * Feature/Scenario: CONFERENCE - Status: OK
678 2021/08/03 14:23:43 INFO: * Feature/Scenario: GATHER - Status: OK
679 2021/08/03 14:23:57 INFO: * Feature/Scenario: HANGUP - Status: OK
680 2021/08/03 14:24:14 INFO: * Feature/Scenario: NUMBER - Status: OK
681 2021/08/03 14:24:27 INFO: * Feature/Scenario: PAUSE - Status: OK
682 2021/08/03 14:24:55 INFO: * Feature/Scenario: PING - Status: OK
683 2021/08/03 14:25:28 INFO: * Feature/Scenario: PLAY - Status: OK
684 2021/08/03 14:26:54 INFO: * Feature/Scenario: PLAYLASTRECORDING - Status: Not OK
685 2021/08/03 14:27:47 INFO: * Feature/Scenario: PLAYLASTRECORDING - Status: OK
686 2021/08/03 14:28:06 INFO: * Feature/Scenario: RECORD - Status: OK
687 2021/08/03 14:28:16 INFO: * Feature/Scenario: REDIRECT - Status: OK
688 2021/08/03 14:28:28 INFO: * Feature/Scenario: REJECT - Status: OK
689 2021/08/03 14:29:35 INFO: * Feature/Scenario: SAY - Status: Not OK
690 2021/08/03 14:29:51 INFO: * Feature/Scenario: SAY - Status: OK
691 2021/08/03 14:30:05 INFO: * Feature/Scenario: SMS - Status: OK
692 2021/08/03 14:30:10 INFO: Passed 13
693 2021/08/03 14:30:10 INFO: Failed 0
694 2021/08/03 14:30:10 INFO: ...END OF TEST SUITE
695
```

/media

In this folder, “Zarbat Tester” places the media file downloaded from CPaaS.



/xml

In this folder, "Zarbat Tester" places the inbound.xml files to be read by CPaaS.

Files

To run your test through this appliance you must have the folders:

CTLang

Feature: PlayLastRecording

In order to listen the last recording of my calls

As an end user

I want to call a number and set up to play the last recording.

Scenario: Record Call

Given my test setup runs

And "NumberA" configured to say "This is the last recording"

And "NumberB" configured to record calls

When I make a call from "NumberA" to "NumberB"

Then "NumberB" should get speech "This is the last recording"

Scenario: Play Last Recording

Given my test setup runs

And "NumberA" configured to gather speech

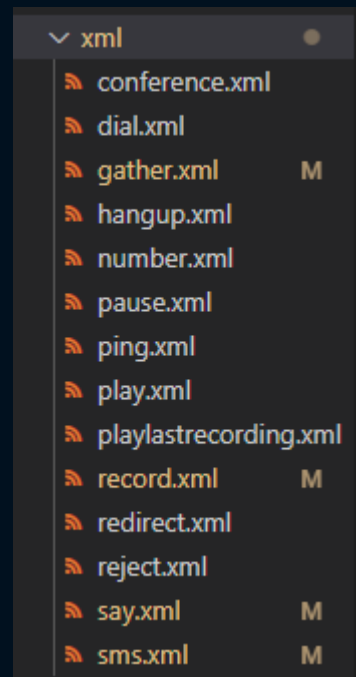
And "NumberB" configured to play last recording

And I make a call from "NumberA" to "NumberB"

Then "NumberA" should get speech "This is the last recording"

And "NumberA" should be reset

And "NumberB" should be reset



Configuration

ApiUrl is the URL used for interacting with Avaya OneCloud CPaaS. The following is the main base URL: <https://api.zang.io/v2/>

BaseUrl is the URL used for interacting with Zarbat Tester callbacks to keep the conversation between OneCloud and Zarbat during test executions. The following is the main base URL: <https://yourname.ngrok.io/>

AccountSid is the System Identification of your account on CPaaS registered at the environment being tested.

AuthToken is the token that is required for the CPaaS authorization registered at the environment being tested.

```
config > config.ini
1  ApiUrl="https://api.zang.io"
2  BaseUrl="http://mohashi.ngrok.io"
3  ApiVersion="v2"
4  # Have to change to 34 bytes beggining with AC to be considered valid.
5  AccountSid="AC777c3e3262352901b0e24b7092f6eef0"
6  # Have to change to 32 bytes to be considered valid.
7  AuthToken="c59b774e18b44769bdc8b4253a7ee0d7"
8  NumberD="+12048003047"
9  NumberDSid="DI777c3e32697ba80d5a4f4082bd4dcff5"
10 NumberC="+12048003029"
11 NumberCSid="DI777c3e3270908f97680344b5a3fea522"
12 NumberA="+12048003030"
13 NumberASid="DI777c3e32d3e6623dbbe740ea93fdebb4"
14 NumberB="+12048003021"
15 NumberBSid="DI777c3e32f4fd09f0fb5b4894ae41ca6d"
16 NumberE="+12048003028"
17 NumberESid="DI777c3e3265711172eae445bc926c72c7"
18 NumberBR1="+5561984385415"
19 NumberBR2="+5561992574040"
20 From=""
21 To=""
22 ToSid=""
23 ActionUrl="http://zang.io/ivr/welcome/call"
24 StatusCallback = ""
25 |
```

Log File

There are two log files today: "zarbat_INFO.log" and "zarbat_DEBUG.log".

```
2021/08/04 11:55:44 INFO: *****
2021/08/04 11:55:44 INFO: START OF TEST SUITE
2021/08/04 11:55:44 INFO: Config: config/config.ini
2021/08/04 11:55:44 INFO: Number of Tries: 2
2021/08/04 11:55:44 INFO: Log: log/zarbat.log
2021/08/04 11:55:44 INFO: Tests:
[conference][gather][hangup][number][pause][ping][play][playlastrecording][record][redirect][reject][say][sms][dial]
2021/08/04 11:55:44 INFO: .....
2021/08/04 11:56:01 INFO: * Feature/Scenario: CONFERENCE - Status: OK
2021/08/04 11:56:15 INFO: * Feature/Scenario: GATHER - Status: OK
2021/08/04 11:56:28 INFO: * Feature/Scenario: HANGUP - Status: OK
2021/08/04 11:56:44 INFO: * Feature/Scenario: NUMBER - Status: OK
2021/08/04 11:56:57 INFO: * Feature/Scenario: PAUSE - Status: OK
2021/08/04 11:57:06 INFO: * Feature/Scenario: PING - Status: OK
2021/08/04 11:57:37 INFO: * Feature/Scenario: PLAY - Status: OK
2021/08/04 11:58:02 INFO: * Feature/Scenario: PLAYLASTRECORDING - Status: OK
2021/08/04 11:58:21 INFO: * Feature/Scenario: RECORD - Status: OK
2021/08/04 11:58:31 INFO: * Feature/Scenario: REDIRECT - Status: OK
2021/08/04 11:58:43 INFO: * Feature/Scenario: REJECT - Status: OK
2021/08/04 11:58:59 INFO: * Feature/Scenario: SAY - Status: OK
2021/08/04 11:59:17 INFO: * Feature/Scenario: SMS - Status: OK
2021/08/04 11:59:23 INFO: * Feature/Scenario: DIAL - Status: OK
2021/08/04 11:59:52 INFO: Passed 14
2021/08/04 11:59:52 INFO: Failed 0
2021/08/04 11:59:52 INFO: ...END OF TEST SUITE
```

How to write your test

To write your tests you have to create a file with "ctlang" extension, and inside of it use the Gherkin language to describe the Feature, the Scenarios, and define your Steps. There are already made sentences available for you. Attention: "Given - When - Then" can be complemented using the "And" particle.

Example:

Write file "example.ctlang" inside "/test" folder.

```
Feature: example
  In order to make gather speech from call
  As an end user
  I want to make a call
  And say "something" and get speech "something".

  Scenario: Make an Example

Given my test setup runs
  And "NumberA" configured to say "what we do in life echoes in eternity"
  And "NumberB" configured to gather speech
  When I make a call from "NumberA" to "NumberB"
  Then "NumberB" should get speech "what we do in life echoes in eternity"
```

How to run your test

First run ngrok with the same subdomain configured in [BaseUrl](#) your configuration file, then execute in the "Zarbat Tester" installation folder through CLI:

```
.\zarbat_test.exe -config config/config.ini -l log/zarbat.log -level info -n 3 -test test/test1.ctlang
```

Where -config is the argument used pass the path of your config.file, -l is the argument used to pass the path of you log file, -level is the argument used to inform the level needed to be written by "Zarbat Tester", and -n is the argument used as the number of tries before a test is considered failed, and - test is the argument used to pass the path of your single tests file containing all the features, scenarios and step defined in the tests.

How to code your test

First: Write your steps inside a go lang file and put into the folder "internal/steps".

Second: After that you will be able to add in the "internal/steps/global_step.go" file your methods and reuse them inside your ctlang files.

How to extend your test

Next Steps

This framework is extensible in the measure that you can create your reserved word as markups and use them for:

1. Tests Grouping
2. Test Constraints
3. Test Chaining
5. Test Scheduling
6. Test Metrics
7. Test Notes
7. Test Notifications (Application Management)

Variables Scope

Today the scope is restricted to each test, but in the future we will propagate variable scope between feature tests enabling variables overriding possibilities.

CTLang Sentences

1. "Number" configured as conference "Zarbat" with size 3

This sentence is used to configure "Number" as a conference room, with the name "Zarbat", and 3 as the maximum number of participants. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

2. "NumberA" configured to dial "NumberB"

This sentence is used to configure "NumberA" to dial to "NumberB". Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

3. "NumberA" configured to dial and send digits "www1234#" to "NumberB"

This sentence is used to configure "Number" to send digits where w represents a waiting time of 500 ms and # represents hash. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

4. "Number" configured to gather digits until "#"

This sentence is used to configure "Number" to gather digits until a hash button is sent. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

5. "Number" configured to gather speech

This sentence is used to configure "Number" to gather speech. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

6. "Number" configured to pause N seconds

This sentence is used to configure "Number" to pause "N" seconds. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

7. "Number" configured to ping URL

This sentence is used to configure "Number" to PING a given url. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

8. "Number" configured to play last recording

This sentence is used to configure "Number" to play the last recording available. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

9. "Number" configured to play tone "5000,10,850"

This sentence is used to configure "Number" to play a tone for 5 seconds of duration, 10 milliseconds of pause, and 850 Hz. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+1999999999" in the Configuration file.

10. "Number" configured to record calls

This sentence is used to configure "Number" to record the inbound calls. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

11. "Number" configured to record calls for download

This sentence is used to configure "Number" to record calls and send download urls back. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

12. "Number" configured to say "I see skies of blue"

This sentence is used to configure "Number" to say "I see skies of blue" to a called number. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

13. "NumberA" configured to send SMS "Actions speak louder than words" to "NumberB"

This sentence is used to configure "NumberA" to send an SMS message to "NumberB". Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

14. "NumberA" configured to dial "NumberB"

This sentence is used to configure "NumberA" to dial to "NumberB". Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

15. "Number" should get last call duration more than or equals to N

This sentence is used to check if the "Number" gets the last call duration more than or equals to N seconds. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

16. "Number" configured to redirect to ping URL

This sentence is used to configure "Number" to redirect incoming calls to ping a given URL. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

17. "Number" configured to reject call

This sentence is used to configure "Number" to reject the incoming calls . Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

18. "Number" configured to hang up after 3 seconds

This sentence is used to configure "Number" to hang up after N seconds. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

19. I list all available numbers

This sentence is used to list all available numbers from CPaaS for buying.

20. I should get to buy 1 from list

This sentence is used to check if the configured account gets to buy 1 from the list. Numbers must be declared in the Configuration File. Example: "AccountSid=A9A9A9A9A9" has to be present in the Configuration file.

21. I make a call from "NumberA" to "NumberB"

This sentence is used to configure "NumberA" to make a call to "NumberB". Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

22. my test setup runs

This sentence is used to run the setup of the test suite. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

23. "Number" should be enter conference "Zarbat"

This sentence is used to check if the "Number" entered in the conference room named "Zarbat". Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

24. "Number" should be able to listen to frequencies "850"

This sentence is used to check if the "Number" got to listen to the frequency of 850 Hz. Numbers must be declared in the Configuration File. Example: "NumberA" in CTLang file has to be present as "NumberA=+19999999999" in the Configuration file.

25. "Number" should be able to view the SMS "test"

This sentence is used to check if the "Number" received a sms message "test".

26. "Number" should be reset

This sentence is used to reset the VoiceUrl parameter of the "Number" on CPaaS.

27. "Number" should get a ping request on the URL

This sentence is used to check if the "Number" gets a ping request on a given URL.

28. "Number" should get call cancel status

This sentence is used to check if the "Number" gets a canceled call status.

29. "NumberB" should get digits "1234" from "NumberA"

This sentence is used to check if the "NumberB" gets digits sent by "NumberA".

30. "Number" should get speech "I see skies of blue"

This sentence is used to check if the "Number" gets the speech "I see skies of blue".

31. "NumberCA1" should get the incoming call from "Number"

This sentence is used to check if the "NumberCA1" gets an incoming call from "Number".

32. "Number" should get transcription "what we do in life echoes in eternity"

This sentence is used to check if the "Number" gets the transcription "what we do in life echoes in eternity".

33. "NumberB" configured to send MMS "Actions speak louder than words" and media "Avaya.jpg" to "NumberC"

This sentence is used to configure "NumberB" to send an MMS to "NumberC" with a message and a media file.

34. Then "NumberB" should be able to view the MMS "I think to myself" with media "Avaya.jpg"

This sentence is used to check if the "Number" gets the message and the media.

Document History

Revised on	Author	Version	Description
03/08/2021	MASARU	0.5.0	Delivery of the Alpha Version.
04/08/2021	MASARU	0.6.0	Run & Install section. Configuration notes for some of the available parameters. Picture with INFO logs view.
05/08/2021	MASARU	0.6.1	Correction of 20th sentence definition. 20. I should get to buy 1 from the list. This sentence is used to check if the configured account gets to buy 1 from the list. Numbers must be declared in the Configuration File. Example: "AccountSid=A9A9A9A9A9" has to be present in the Configuration file.
10/08/2021	MASARU	0.6.2	Added MMS sentences and methods.
19/08/2021	MASARU	0.61.0	Docker Image published.
24/08/2021	ARON TALAI	0.7.0	docker pull masaruohashi/zarbat_tester:zarbat_tester
25/08/2021	ARON TALAI	0.71.0	Enriching the debugging logs of Zarbat.