Fonetwish API Document

Version 1.0

For
Developers and Content Providers

OVERVIEW

USSD protocol allows interactive services between a mobile handset (end-users) and applications hosted by the mobile operators.

Fonetwish is a platform to enable creation and management of USSD applications. It interfaces with content providers and developers at one end; and with operators at the other end to provide USSD applications to mobile subscribers.

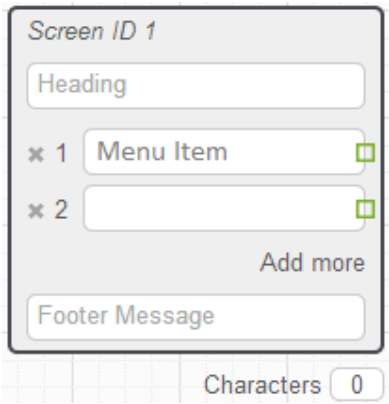
USSD applications are created using sets of menus. These sets of menus create the USSD application flow. Each menu selection or text input triggers next set of menus.

This document will give you an overview of Fonetwish platform and explain the various APIs used for communication between the platform and developers/ content providers.

Those who are familiar with RESTful APIs that use JSON data format will find the document fairly basic. Fonetwish applications use HTTPS request/response protocol.

App Screen

Following figure depicts the screen which is used to create an app:



You can play around with menu items to make your application engaging and compelling. Menu item can be static text or can be fetched from dynamic API. In case a user types a URL, the system will analyze the same and link icon in the header message will be highlighted. All the menu items of a particular screen id will be fetched from that single URL. There is no need to enter different URL's for different menu items. Maximum menu items = 9

Field details of app screen is given as under:

Field	Description	Type	Max Length	Required
Screen ID	Displays the unique screen id which is automatically generated.	NA	NA	NA
<Header Message>	Allows User to enter header message of screen	String	60	Optional
<Menu Item>	Allows User to enter menu values, menus can be static or dynamic.	String / URL	160	Required
<Footer Message>	Allows user to enter footer message of ussd application screen. This will be available on first screen only.	String	60	Optional

<Connectors>	Allows user to create and link sub menus	NA	NA	Optional
<Add More>	To add more items to menu. Maximum menu items can be 9	NA	NA	Optional

API Description

Where the application has dynamic content/menu, the platform would require to communicate at runtime. For this, Fonetwish has exposed API interfaces. These API interfaces are REST based APIs. Request and response will be in JSON format.

APIs will provide security through HTTPS protocol. Authentication will be done through SSL mechanism. Before using these APIs, content provider will need to register on platform. For content provider push, the content provider will be provided with one time access token that it will use in subsequent requests to call Fonetwish platform.

APIs support POST method with content type as application/json and data as body parameters. In case of content provider initiated sessions, the content provider would be required to send access token in headers as specified in later sections. Access token would be provided at the time of content provider registration.

Two sets of API are provided for two types of requests:

- Subscriber Initiated Requests
- Content Provider Initiated Requests

Subscriber Initiated Request:

The request (USSD Session) originates from the mobile subscriber that is forwarded to Fonetwish platform. The platform then forwards the request to the content provider. Content provider could then respond to this request that is further responded back to subscriber.

Content Provider Initiated Request:

The content provider originates a request (USSD Session or USSD Notification) from its side by calling Fonetwish platform. The platform then originates the request for subscriber from its side to the subscriber. USSD Session is an interactive session but USSD Notification is not a session.

Subscriber Initiated Requests

In subscriber initiated requests, the platform will forward the request to the content provider in case of dynamic content. Enterprise will call the URL that has been provided by the content provider in his menu flow. In this case, the content provider will provide the URL that platform will hit with following sets of request parameters.

On dynamic request, the following information will be provided to the content provider:

Fields	Description	Data Type	Length	Required
sessionId	A unique Session Id for the current user session. This is a UUID.	String	32	Required
txId	A unique Transaction Id for current transaction. This is a UUID.	String	32	Required
appId	Unique App id for the current application, given at the time of creation of App. This is a UUID.	String	32	Required
userId	Could be MSISDN for premium user or else a UUID generated userId.	String	32	Required
menuId	Current screen menu id on which the subscriber is active.	String	32	Required
contentSelection	Input from the subscriber.	String	160	Required

The response fields for the request is as follows:

Fields	Description	Data Type	Length	Required
sessionId	A unique Session Id for the user session as sent in the request.	String	32	Yes
respTxId	Unique transaction id for the request as sent in the request.	String	32	Yes
status	Response for the transaction from content provider as success code as defined in Error Codes.	Integer	3	Yes
responseType	Response type for the request. Whether a content or menu. CONTENT for content, MENU for menu	String	16	Yes
menuId	Current screen menu id on which the subscriber is right now, as sent in the request	String	32	Yes
data	The content/menu that needs to be shown back to the subscriber.	String	1MB	Yes

Data is either a content or list of content or a menu. In case, there is a list of content, the platform will not access the content provider, but cache these results with itself and will keep on communicating with the subscriber. At present it supports max of 10 data items. In case of more than 10 items, the last would be rejected.

In case of Menu, the response will be shown to the users as Menu. The max length provided for the data in case of this is 1MB, in which it has to provide the Menu data.

Content Provider Initiated Requests

In content provider initiated requests, the content provider will initiate a request for subscriber which platform will forward to subscriber. In this case, platform will provide URL that content provider will use to initiate the request.

These requests could be of two types:

- Interactive sessions
- Notifications

Interactive Sessions

If the Content Provider plans to initiate interactive session then the following request needs to be provided by the content provider to the Platform:

Fields	Description	Data Type	Length	Required
sessionId	A unique Session Id for the user session created by Content Provider	String	32	Yes
txId	Unique transaction id for the request creates by Content Provider	String	32	Yes
userId	Could be MSISDN for premium user or else a generated userId	String	32	Yes
requestType	Response type for the request. Whether a content or menu. CONTENT for content, MENU for menu	String	8	Yes
data	The content that needs to be shown back to the subscriber	String	1MB	Yes

The content provider would be responsible to manage the session and transaction for these users.

The response to this requests would be:

Fields	Description	Data Type	Length	Required
sessionId	A unique Session Id for the user session created by Content Provider.	String	32	Yes
txId	Unique transaction id for the request creates by Content	String	32	Yes

Fields	Description	Data Type	Length	Required
	Provider.			
userId	Could be MSISDN for premium user or else a generated userId.	String	32	Yes
responseCode	Whether the request was sent to the user.	Integer	3	Yes
contentSelection	Input from the subscriber.	String	160	Yes

Note: MSISDN will be only shared on demand with premium content provider on platform.

Notifications

If the content provider plans to do bulk USSD sessions, like session notifications or single session with single response, then it could provide multiple requests for the user.

Fields	Description	Data Type	Length	Required
userId	List of MSISDNs for which notification needs to be sent.	String	1 MB	Yes
notification	Notification text	String	160	Yes
requestId	UUID for the request	String	32	Yes

Since this will be bulk USSD notification, there will only be an acknowledgement of this request with a requestId. The notifications would be sent to the User and U2opia will provide a FTP site for status on this bulk messages with exactly the same requestId.

Fields	Description	Data Type	Length	Required
requestId	UUID for the request as shared in notification request.	String	32	Yes

Data Structure as Part of Above API

There are 5 data structures used in these APIs. The segregation of these data structures are made in the following way:

- 2 for different type of request, which is Subscriber initiated and Content Provider bulk request.
- 3 for different type of response, which is Menu Type, Content Type and Notification type.

Request Data Structure

Request Data Structure Initiated by Subscriber

The request data structure in JSON format is a plain data structure as shown in the following example:

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "1",
"userId": "919810301297",
"menuId": "27_1",
"contentSelection": "3"
}
```

Request Data Structure Initiated by Content Provider for Interactive Sessions

The request data structure in JSON format is a plain data structure with information having no hierarchy. For example

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"txId": "335a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "2",
"userId": "919811242341",
"requestType": "MENU",
"data": {
                    "items": [
                                {
                                            "menuContent": "11",
                                            "menuOrder": 1,
                                            "itemType": "static",
                                            "requestURL": "",
                                            "screen": {
                                                        "items": [
                                                                    {
                                                                                "menuContent": "33",
```

```
                                                "menuOrder": 1,
                                                "itemType": "static",
                                                "requestURL": "",
                                                "screen": {}
                                        },
                                        {
                                                "menuContent": "44",
                                                "menuOrder": 2,
                                                "itemType": "static",
                                                "requestURL": "",
                                                "screen": {}
                                        }
                                ],
                                "menuId": "1_1",
                                "menuHeader": ""
                        }
                },
                {
                        "menuContent": "22",
                        "menuOrder": 2,
                        "itemType": "dynamic",
                        "requestURL": "https://api.sample.com",
                        "screen": {}
                }
        ],
        "menuId": "1",
        "menuHeader": "heading1",
        "menuFooter": ""
        }
    }
}
```


Request Data Structure Initiated by Content Provider for Notifications


```
{
"userId": [
"919812345644", "919812345644","919812345644","919812345644",
"919812345644","919812345644"
],
"notification": "Welcome Message",
"requestId": "229a9695-3df8-46b5-a8ea-dfb4590bf321"
}
```


Response Data Structures

There are 3 types of response data structure as stated in above sections.

Response Data Structures for Subscriber Initiated Request

Response Where Content is Provided

The data part here is list of content:


```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "345a9695-3df8-46b5-a8ea-dfb4590bf321",
```

```
"userId": "919810301297",
"status":200,
"responseType": "CONTENT",
"menuId": "27_1",
"data": ["Content1","Content2", "Content3" …]
}
```

Response Where Menu is Provided

```
{
        "sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
        "transactionId": "345a9695-3df8-46b5-a8ea-dfb4590bf321",
         "userId": "919810301297",
         "status":200,
        "responseType": "MENU",
                "data": {
                 "items": [
                        {
                                "menuContent": "11",
                                "menuOrder": 1,
                                "itemType": "static",
                                "requestURL": "",
                                "screen": {
                                        "items": [
                                                {
                                                        "menuContent": "33",
                                                        "menuOrder": 1,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                },
                                                {
                                                        "menuContent": "44",
                                                        "menuOrder": 2,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                }
                                        ],
                                        "menuId": "1_1",
                                        "menuHeader": ""
                                }
                        },
                        {
                                "menuContent": "22",
                                "menuOrder": 2,
                                "itemType": "dynamic",
                                "requestURL": "https://api.sample.com",
                                "screen": {}
                        }
                ],
                "menuId": "1",
                "menuHeader": "heading1",
                "menuFooter": ""
                }
        }
}
```

The meaning of each of these items are listed below in tabular format.

Fields	Description	Data Type	Length	Required
menuHeader	Header information of menu	Varchar	200	Optional
items	Root tag of menu, contains the sub menu informations	Array	-	Yes
responseType	MENU/CONTENT	Varchar	4	Yes
userId	User Id where content/menu need to be sent.	Array	-	Yes
menuContent	Display text for menu item	Varchar	200	Yes
menuOrder	Order in which menu need to be dispay to user , menu Order should be start from 1 and maximum menus could be 9	Integer	1	Yes
itemType	Content type of menu, it could be either static, input or dynamic. If the item type is static then it must have menu content value and if the item type is dynamic then it should contains the tag request URL, the requestUrl must contains the valid URl	Varchar	10	Yes
menuId	unique identifier of menu	Varchar	200	Yes

Response Data Structures for Content Provider Initiated Request

There are two types of request one is for interactive session and other is for notifications.

Response Data Structure for Interactive Session

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "1",
"userId": "919810301297",
"menuId": "27_1",
"contentSelection": "3"
}
```

Response Data Structure for Bulk Content

```
{
"requestId": "229a9695-3df8-46b5-a8ea-dfb4590bf321"
}
```

Error Codes

The error codes implemented in the Enterprise Platform are shown in the following table. Most of the errors are common with the HTTP errors irrespective that they are thrown from Enterprise platform.

Error Codes with Description

Error Code	Description
200	Ok/Success
201	Created
202	Accepted
400	Bad Request The request cannot be fulfilled due to bad syntax
401	Unauthorized
404	Not Found
405	Method Not Allowed
408	Request Timeout
415	Unsupported Media Type
429	Too Many Requests
495	Cert Error
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
505	HTTP Version Not Supported

Sample Code

Subscriber Initiated Request with Content as Response

URL: As specified by Content Provider
METHOD TYPE: POST
CONTENT TYPE: application/json
HEADERS: None

REQUEST:

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "1",
"userId": "919810301297",
"menuId": "27_1",
"contentSelection": "3"
}
```

RESPONSE:

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "345a9695-3df8-46b5-a8ea-dfb4590bf321",
"status":200,
"responseType": "CONTENT",
"menuId": "27_1",
"data": ["POLITICS","SPORTS", "CULTURE"]
}
```

Subscriber Initiated Request with Menu as Response

URL: As specified by Content Provider
METHOD TYPE: POST
CONTENT TYPE: application/json
HEADERS: None

REQUEST:

```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "1",
"userId": "919810301297",
"menuId": "27_1",
"contentSelection": "3"
}
```

RESPONSE:

```
{
        "sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
        "transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
```

```
        "status":200,
        "responseType": "MENU",

        "data": {
                "items": [
                        {
                                "menuContent": "11",
                                "menuOrder": 1,
                                "itemType": "static",
                                "requestURL": "",
                                "screen": {
                                        "items": [
                                                {
                                                        "menuContent": "33",
                                                        "menuOrder": 1,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                },
                                                {
                                                        "menuContent": "44",
                                                        "menuOrder": 2,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                }
                                        ],
                                        "menuId": "1_1",
                                        "menuHeader": ""
                                }
                        },
                        {
                                "menuContent": "22",
                                "menuOrder": 2,
                                "itemType": "dynamic",
                                "requestURL": "https://api.sample.com",
                                "screen": {}
                        }
                ],
                "menuId": "1",
                "menuHeader": "heading1",
                "menuFooter": ""
                }
        }
}
```

Content Provider Initiated Request with Interactive Session

URL: http://enterprise.fonetwish.com
METHOD TYPE: POST
CONTENT TYPE: application/json
HEADERS: {ENTERPRISE_ACCESS_CODE: 100a9695-3df8-46b5-a8ea-dfb4590bf321}

REQUEST:


```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"txId": "335a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "2",
"userId": "919811242341",
"requestType": "MENU",
"data": {
                "items": [
                        {
                                "menuContent": "11",
                                "menuOrder": 1,
                                "itemType": "static",
                                "requestURL": "",
                                "screen": {
                                        "items": [
                                                {
                                                        "menuContent": "33",
                                                        "menuOrder": 1,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                },
                                                {
                                                        "menuContent": "44",
                                                        "menuOrder": 2,
                                                        "itemType": "static",
                                                        "requestURL": "",
                                                        "screen": {}
                                                }
                                        ],
                                        "menuId": "1_1",
                                        "menuHeader": ""
                                }
                        },
                        {
                                "menuContent": "22",
                                "menuOrder": 2,
                                "itemType": "dynamic",
                                "requestURL": "https://api.sample.com",
                                "screen": {}
                        }
                ],
                "menuId": "1",
                "menuHeader": "heading1",
                "menuFooter": ""
                }
        }
```

RESPONSE:


```
{
"sessionId": "229a9695-3df8-46b5-a8ea-dfb4590bf321",
"transactionId": "314a9695-3df8-46b5-a8ea-dfb4590bf321",
"appId": "2",
"userId": "919810301297",
"menuId": "27_1",
"contentSelection": "3"
}
```


Content Provider Initiated Request for Bulk USSD Requests (Notifications)

URL: http://enterprise.fonetwish.com
METHOD TYPE: POST
CONTENT TYPE: application/json
HEADERS: {ENTERPRISE_ACCESS_CODE: 100a9695-3df8-46b5-a8ea-dfb4590bf321}

REQUEST:


```
{
    "userId": [
        "919812345644", "919812345644","919812345644","919812345644",
        "919812345644","919812345644"
    ],
    "notification": "Welcome Message",
    "requestId": "229a9695-3df8-46b5-a8ea-dfb4590bf321"
}
```


RESPONSE:


```
{
    "requestId": "229a9695-3df8-46b5-a8ea-dfb4590bf321"
}
```

APPENDIX

Syntax Convention	Description
Text without brackets/braces	Indicates content you type as shown.
< *Text inside angle brackets*>	Placeholder for which you must supply a value. The variable is usually shown in italics. See Placeholders below.
[Text inside brackets]	Indicates optional items; for example, CREATE TABLE [*schema_name*.]*table_name* The brackets indicate that the *schema_name* is optional. Do not type the square brackets.
{ Text inside braces }	Indicates a set of options from which you choose one; for example: QUOTES { ON | OFF } indicates that exactly one of ON or OFF must be provided.You do not type the braces: QUOTES ON
Backslash \	Continuation character used to indicate text that is too long to fit on a single line.
Ellipses ...	Indicate a repetition of the previous parameter. For example,option[,...] means that you can enter multiple, comma-separated options. Note: Showing an ellipses in code examples might also mean that part of the text has been omitted for readability, such as in multi-row result sets.
Indentation	Is an attempt to maximize readability;
Placeholders	Items that must be replaced with appropriate identifiers or expressions are shown in italics.
Vertical bar |	Is a separator for mutually exclusive items. For example:[ASC | DESC] Choose one or neither. You do not type the square brackets.