



USSD
API
V1.12



always active **mobile**

An AAT Group Company



We are proud to be an AAT Group company

www.alwaysactivemobile.co.za

care@aat.co.za

(+27) 031 100 0201

Contents

Version history	3
USSD Developers Guide- Introduction.....	4
USSD Functional Mechanism Overview	4
Internals of the USSD Daemon	4
XML Samples.....	5
Question 1.....	5
Question 2.....	5
Question 3.....	5
JSON Samples.....	6
The Data is deserialized into these classes	6
XML / JSON Parameters	7
<request> tag.....	7
<headertext> tag	7
<options> tag	7
<option> tag.....	7
Attributes	7
Example Session.....	8
XML Free Text Example.....	8
Json Free Text Example.....	9
USSD Timeouts and Character Limitations	10
MT (Mobile Terminating) Traffic.....	10
MO (Mobile Originating) Traffic.....	10
Network Session Timeouts	10
Handset Timeouts	10
Application Timeouts	10
Network Charge Periods	10
Parameters added in query string	11
Integration	11
External Links	11

Version history

Revision number	Revision Date	Author	Changes
1.0	2009-11-20	Kieron Thwaites	Initial document release
1.1	2010-07-02	Jon Hudson	Update Links
1.2	2010-07-09	Jon Hudson	Update Free Text
1.3	2011-09-02	Jon Hudson	Update Sessions
1.4	2011-11-18	Jon Hudson	Update Session Time-outs and costs
1.5	2012-03-08	Jon Hudson	Updated MTN MT Session Time
1.6	2012-08-23	Jon Hudson	Updated Menu Character Limits
1.7	2013-06-04	Jon Hudson	Updated CellC Charges
1.8	2013-09-13	Jon Hudson	Updated Network / Application Timeouts
1.9	2015-01-20	Jon Hudson	Updated "provider" return options
1.10	2016-09-05	Jon Hudson	Updated CellC Timeouts
1.11	2017-10-04	Jon Hudson	Added JSON support
1.12	2018-05-15	Natasha Meyer	Refined JSON support

USSD Developers Guide- Introduction

Unstructured Supplementary Service Data, abbreviated as **USSD**, is a capability of all GSM phones and all networks. Response times for interactive USSD-based services are usually quicker than those used for SMS.

As an analogy, USSD is similar to telnet, while SMS is similar to mail.

USSD operates by using a number such as ***120*123#** to trigger a menu structure which is displayed on the user's handset. Our USSD URL Caller solution allows developers to create their own menu structure with the use of HTTP and XML. A simple communication with our server allows the info to be delivered to the handset.

USSD is often used by the Networks where users can check their available balance or browse for various services and settings on their handset.

USSD Functional Mechanism Overview

This section of the doc is to explain in general how the USSD Daemon goes about handling USSD sessions. USSD implementation centres around URL based requests and responses.

Upon creation of a new USSD campaign, the client is required to create the URLs for the USSD daemon in order for the first and subsequent responses to be displayed. Later on in this doc, examples of the XML and an example USSD session is given to aid in understanding of the system.

Internals of the USSD Daemon

Upon the USSD Daemon calling the first "USSD Request" URL, the daemon will read the XML located at the URL. The Daemon will then construct a menu structure for the subscriber and will create a response URL based on the subscriber's selected menu option.

This will continue for each "USSD response" until the USSD session has been completed.

In the next section, an example of the XML, a breakdown of the XML parameters and a simple example of a USSD session is given.

XML Samples

These are the examples to demonstrate the XML required to be generated by clients for their USSD campaigns. Here we can see a simple 3-question campaign and the XML, which will represent each question.

Question 1

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>Can you answer the question?</headertext>
  <options>
    <option command="1" order="1"
callback="http://yourdomain.tld/ussdxml.ashx?file=2"
      display="true">Yes I can!</option>
  </options>
</request>
```

Question 2

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>Can you answer the second question?</headertext>
  <options>
    <option command="1" order="1"
callback="http://yourdomain.tld/ussdxml.ashx?file=3"
      display="true">Yes I can!</option>
    <option command="2" order="2"
callback="http://yourdomain.tld/ussdxml.ashx?file=1"
      display="true">Take me back to the first one!</option>
  </options>
</request>
```

Question 3

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <headertext>You passed the test. Congratulations!</headertext>
</request>
```

JSON Samples

This is an example of the JSON format expected, instead of the XML.

```
{
  "headertext": "The Header ...",
  "options": [
    {
      "command": 1,
      "order": 1,
      "callback": " http://yourdomain.tld/ussdxml.ashx?file=3",
      "display": true,
      "text": "Option 1"
    },
    {
      "command": 2,
      "order": 2,
      "callback": " http://yourdomain.tld/ussdxml.ashx?file=400&ref=1",
      "display": true,
      "text": "Option 2"
    },
    {
      "command": 3,
      "order": 3,
      "callback": " http://yourdomain.tld/ussdxml.ashx?ref=230",
      "display": true,
      "text": "Option 3"
    }
  ]
}
```

The Data is deserialized into these classes

```
public class JSONOption
{
    public short command { get; set; }
    public int order { get; set; }
    public string callback { get; set; }
    public bool display { get; set; }
    public string text { get; set; }
}

public class JSONRequest
{
    public string headertext { get; set; }
    public IList<JSONOption> options { get; set; }
}
```

XML / JSON Parameters

<request> tag

This tag encapsulates the entire XML structure.
It has no attributes.

It has the following subtags: <headertext> and <options>. <headertext> is mandatory, <options> is optional. Each subtag may only appear ONCE in the XML.

<headertext> tag

This tag contains the message that will be displayed on the mobile user's handset.
It has no attributes.

It has no subtags.

<options> tag

This tag contains any menu options that are to be displayed on the mobile user's handset.
It has no attributes.

It has the subtag <option>, which is mandatory and can be specified multiple times (for each menu option).
The <options> tag itself is optional. Should it not be present, the USSD Service will only display the text in the <headertext> tag, and then terminate the session.

<option> tag

This tag specifies a single menu option. The inner text of this tag is the menu text that will be displayed on the mobile handset.

It has no subtags.

It has the following attributes: `command`, `order`, `callback` and `Display`

Attributes

`command` - the character sequence that the user must enter on his handset to select the menu option. Must be convertible to a 16-bit signed integer.

`order` - the order amongst the menu options that this menu option will be shown. Must be convertible to a 16-bit signed integer.

`callback` - the URL that will be called when this menu option is selected by the mobile user. It is assumed that the URL will also return XML data (in this format) that will be used to determine the response to be sent to the mobile user's handset.

`display` - if set to true, the menu option will be displayed on the user's handset; if set to false, the menu item will not be displayed on the user's handset.

Example Session

Using the example XML on Page 3 above and assuming that the client has configured the USSD service to call a URL that returns the first example XML file, when the mobile user initiates a session:

```
Can you answer the question?  
1 - Yes I can!
```

The mobile user enters 1.

```
Can you answer the second question?  
1 - Yes I can!  
2 - Take me back to the first one!
```

The mobile user enters 2.

```
Can you answer the question?  
1 - Yes I can!
```

The mobile user enters 1.

```
Can you answer the second question?  
1 - Yes I can!  
2 - Take me back to the first one!
```

The mobile user enters 1.

```
You passed the test. Congratulations!
```

The USSD session ends.

For **free text** answers, add 1 option with no answer text and set the display to false. The returned input will be returned in the "request" parameter in the query string. The example below shows how to declare a free text answer type question

XML Free Text Example

```
<?xml version="1.0" encoding="utf-8" ?>  
<request>  
  <headertext>Please enter you name.</headertext>  
  <options>  
    <option command="1" order="1" callback="http://yourdomain.tld/ussdxml.ashx?file=3"  
      display="false"></option>  
  </options>  
</request>
```

If the user returns "Mark", The USSD service will append the request parameter to your callback URL. Below would be the outcome of the question:

<http://yourdomain.tld/ussdxml.ashx?file=3&msisdn=132123123&request=Mark>

Json Free Text Example

```
{
  "headertext": "The Header ...",
  "options": [
    {
      "command": 1,
      "order": 1,
      "callback": " http://yourdomain.tld/ussdxml.ashx?file=3",
      "display": false,
      "text": ""
    }
  ]
}
```

USSD Timeouts and Character Limitations

MT (Mobile Terminating) Traffic

Network	First Menu Characters	Remaining Menu's
All	140	160

MO (Mobile Originating) Traffic

Network	First Return Characters	Subsequent Return Characters
All	160	160

Network Session Timeouts

The total amount of time allowed on a single USSD Session.

Network	Minutes
Vodacom	3
MTN	4
CellC	3
Telkom Mobile (8ta)	3

Handset Timeouts

Response received from handset to network

Network	Seconds
Vodacom	60
MTN	60
CellC	30
Telkom Mobile (8ta)	40

Application Timeouts

Response received from application to network

Network	Seconds
All	10

Network Charge Periods

Network	Minutes/Seconds
Vodacom	60c per minute
MTN	20c per 20 seconds
CellC	20c per 20 seconds
Telkom Mobile (8ta)	20c per 20 seconds

**All figures inclusive of VAT*

Parameters added in query string

The USSD Service will always add the parameter `msisdn=` to the query string of any URLs to call; this parameter will equal the mobile number of the user in international format (e.g. 27820000000). Developers may use this parameter in any way that they deem fit.

The USSD service will also always add the `provider=` parameter. This parameter will indicate which Service Provider or Instant Messenger the `msisdn` is contracted to. The possible values are:

- Vodacom
- MTN
- CellC
- Telkom_8ta
- WeChat
- Mxit
- HTTP

The USSD Service will also add the parameter `request=` to the query string. This parameter holds the input that the user submitted from their device. This is used for Free Text answers.

Integration

USSD can be used for a multitude of purposes, including:

- Contests
- Polls
- Voting

Due to the fact that USSD calls a URL, it also passes the users MSISDN through to your URL allowing you to make use of any of our other mobile solutions and run business logic.

Example:

- Upon receipt of a USSD call, an SMS or MMS can be sent to the user.

External Links

USSD Wikipedia Link - http://en.wikipedia.org/wiki/Unstructured_Supplementary_Service_Data