



Arad ITC

Arad SMS Restful API

Developer's guide



Preferences

Starting up	2
Basic concepts.....	2
Parameters and Methods	3
Send Method	3
Sending via URL	5
Send method for VAS services	6
GetCredit method	8
Error Code List	10



Starting up

This document is intended for developers who wish to send and receive SMS in their software via Arad's messaging systems. All the required methods are described here.

After signing a contract with our sales department, your account will be activated. Afterwards, you will be provided with a username and a password. These items are private and personal and are used to login to your page at <http://yourdomain.com> .

Basic concepts

When assigning values to parameters, please note:

- Each SMS consists of at maximum 70 characters for Persian and 160 characters for English. If the SMS is bigger, it will be divided to two or more messages.
- All parameters are case sensitive.
- Mobile phone numbers should be in the format of 989*****.
- The sender's phone number does not require any prefixes.
- Text messages are sent in packets of 200.i.e. in each request you can send 200 messages. The requests will change based on the number of parts per each SMS. If SMS has one part, the number of recipient will be 200. If the message has 2 parts, the recipient's count will be 100 and so on.
- Your IP should be set in your user panel before you could call the methods in the API.



Parameters and Methods

Send Method

Method's URL: <http://your-server-ip:8080/Messages/Send>

This method should be called using POST mechanism and parameters are set using JASON or XML.

Below you can find a sample input along with a brief description.

XML input:

```
<Sms>

  <CheckId>C63DEE7C-7CD9-453F-8B58-357526D8A824 </CheckId>

  <SmsText>hi</SmsText>

  <Receivers>98912xxxxxxx,98913xxxxxxx,... </Receivers>

  <SenderId>6500</SenderId>

  <IsFlash>true</IsFlash>

  <UDH></UDH>

</Sms>
```

JSON input:

```
{ "SmsText": "hi", "Receivers": "98913xxxxxxx,98912xxxxxxx,...",

  "SenderId": "6500", "IsUnicode": true, "IsFlash": false, "UDH": "", "CheckId": "", }
```



Method			Send
Input			
Description	Required/ Optional	Parameter Name	Type
Sender number	Optional	SenderId	string
text Message	Required	SmsText	string
Mobile phone numbers of the recipients separated by “ , ”	Required	Receivers	string
ID which is assigned in user's system	Optional	CheckId	Guid
Header which is defined by user	Optional	udh	string
Display as flash	Optional	isflash	bool
Please note username and password are read from the header of the request.			
Output			
Description	Type	Parameter Name	
Request result	bool	IsSuccessful	
State code	int	StatusCode	
Registered Id in the SMS system	Guid	BatchId	

If you do not specify a SenderId, your message will be send using a default number which is set on your user account.

Sending via URL

Send your parameters to the address below using GET mechanism:

<http://your-server-ip:8080/Messages/SendViaURL>

Method			SendViaURL
Input			
Description	Required/ Optional	Parameter Name	Type
User name	Required	Username	string
Password	Required	Password	string
Sender number	Optional	SenderId	string
text Message	Required	SmsText	string
Mobile phone numbers of the recipients separated by “ , ”	Required	Receivers	String
Header which is defined by user	Optional	CheckId	Guid
Header which is defined by user	Optional	udh	string
Display as flash	Optional	isflash	bool
Output			
Description	نوع پارامتر		نام پارامتر
Request result	bool		IsSuccessful
State code	int		StatusCode
Registered Id in the SMS system	Guid		BatchId



For Example:

<http://your-server-ip:8080/Messages/SendViaURL?Username=username&password=password&SenderId=6500&SmsText=salam&Receivers=989111111111,989111111111>

Send method for VAS services

Methods URL:

<http://your-server-ip:8080/VASMessages/Send>

This method should be called using POST mechanism and parameters are set using JASON or XML.

XML input:

```
<VAS>
<CheckId>C63DEE7C-7CD9-453F-8B58-357526D8A824 </CheckId>
<ServiceId>1234</ServiceId>
<SmsText>salam</SmsText>
<Receiver> [9891*****] </Receiver>
<GroupId>1100</GroupId>
</VAS>
```



Method			Send
Input			
Description	Required/ Optional	Parameter Name	Type
Text message	Required	SmsText	string
ID which is assigned in user's system	Optional	CheckId	Guid
VAS service ID	Required	ServiceId	string
Added value group code	required	GroupId	int
The recipient number (if it is empty , the whole group will be receiving the text message)	optional	Receiver	String
Please note username and password will be read from the header.			
Output			
Description	Parameter Type		Parameter Name
Request result	bool		IsSuccessful
State code	int		StatusCode
Registered Id in the SMS system	Guid		BatchId

If the operation is **successful**, the output will be in the format of:

```
<SendSms>
  <IsSuccessful>true</IsSuccessful>
  <StatusCode>0</StatusCode>
  <BatchId>4cbac7e3-5b24-489c-96cf-057639e87c01</BatchId>
</SendSms>
```




IsSuccessful: states the result of the operation, if the operation is successful, it will be assigned the value 'true' otherwise 'false'.

StatusCode : this parameter will contain the error code in cases which the operation has not successful. A list of error codes can be viewed at the end of this document.

BatchId: If your operation is successful an Id will be returned.

If the operation **failes** the output will be in the format of:

```
<ResponseMessage>
  <IsSuccessful>false</IsSuccessful>
  <StatusCode>7</StatusCode>
  <Message>Minimum length for receiver number is 12 characters. </Message>
  <StackTrace> Minimum length for receiver number is 12 characters. </StackTrace>
</ResponseMessage>
```

It is worth mentioning that the correct format for receivers' number is 98913xxxxxxx.

GetCredit method

Methods URL: <http://your-server-ip:8080/Messages/GetCredit>

This method should be called using POST mechanism and parameters are set using JASON or XML.

The input for this method can be seen below:

Method			GetCredit
Input			
Description	Required/ Optional	Parameter Name	Type
User name	Required	Username	string
Password	Required	Password	string



A sample output has been shown below:

```
<UserCredit>  
  <IsSuccessful>false</IsSuccessful>  
  <StatusCode>0</StatusCode>  
  <Credit>9954975744.84</Credit>  
</UserCredit>
```

Error Code List

StatusCode			
Error code	Title	Description	Points to notice
0	None	No error	If the operation is successful, this value is set in status code.
1	AccountIsInactive	Account is inactive	
2	AccountIsExpired	Account has expired	
3	AccountIsInvalid	Account is invalid	
4	BadRequest	The request is invalid	
5	InternalServerError	Server generated an error	
6	ReceiversIsEmpty	Recipient list is empty	The correct format for the recipient number is : 98913xxxxxxx
7	ValidationNotValid	Input data is not valid	Parameter validation errors
8	IPInvalid	IP is invalid	
9	ReceiverCountNotValid	The total number of recipients is invalid	Total number of recipients should be proportional to the number of SMS parts. Maximum 200 SMS