# API Documentation

**ATT**

## What is the AT&T API Platform SDK for Titanium?

The AT&T API Platform SDK for Titanium is designed to provide an easy way to develop Titanium mobile applications that can access the AT&T APIs using JavaScript. The Appcelerator Titanium experience for AT&T developers focuses on an enhanced capability to leverage the AT&T network-based services to create feature-rich apps. Access to services like SMS, Terminal Location, and MMS, in conjunction with the ability to enable direct carrier billing, provides a multitude of options not normally available to mobile application developers.

* [ATT](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT)
* [ATT.Location](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Location)
* [ATT.MMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MMS)
* [ATT.MOBO](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MOBO)
* [ATT.Notary](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Notary)
* [ATT.OAuth](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.OAuth)
* [ATT.Payment](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment)
* [ATT.SMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.SMS)
* [ATT.Speech](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Speech)
* [ATT.WAPPush](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.WAPPush)

# [ATT.Location](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Location API enables you to provide your customers with location aware applications that can be crucial to the service you offer. It provides an easy way to enhance a number of features. For instance, adding location awareness to your application allows you to engage customers by enriching social interactions, supporting location aware advertising, providing location aware search, and giving directions.

**Description:**

The Location API enables authorized applications to perform the following methods.

**Methods:**

1) Get Device Location

## Methods

### Get Device Location

[getDeviceLocation](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Location-method-getDeviceLocation)( options, success, error ) : Object

Returns the location information for a device.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + requestedAccuracy : Number (optional)

Specifies the accuracy of the device location. The value of this parameter must be within the range of 0 to 20000 meters.

This parameter value provides an indication to the API gateway as to which location technology should be used to find the location of the specified device.

* + - 0 - 800 Assisted GPS (A-GPS)
    - 801-9999 Enhanced Cell-Id (ECID)
    - 10000 - 20000 Cell-Id (CID)

The default value is: 100

* + tolerance : String (optional)

Specifies a tolerance value that defines the application's priority of response time versus accuracy. When this parameter is used, it must be present in the body as a query parameter.

Valid values are:

* + - NoDelay - No compromise on the priority of the response time over accuracy
    - LowDelay – The response time could have a minimum delay for a better accuracy
    - DelayTolerant - Response time could be compromised to have high delay for better accuracy

The default value is: 'LowDelay'

* + acceptableAccuracy : Number (optional)

Specifies an acceptable limit of accuracy that combines with the value of the requestedaccuracy parameter. If the Network is unable to return a response with an accuracy that is less than or greater than the value specified in the requestedaccuracy parameter by the amount specified in this parameter, then the service error (SVC0200) will be returned indicating “Accuracy of location is not within acceptable limit”.

The default value is: 10000

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getDeviceLocation method.

getDeviceLocation({

'requestedAccuracy' : xxxx,

'tolerance' : 'xxxxxxxxx',

'acceptableAccuracy' : xxxxx

},function(data) {

success Callback

}, function(error) {

error Callback

});

# [ATT.MMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Multimedia Messaging Service (MMS) API greatly enhances the power of your communications by moving beyond the text-only capabilities of Short Message Service (SMS) messaging. Using the MMS API, you can send Mobile Terminated or receive Mobile Originated) messages that include video, images and audio attachments as well as text.

**Description:**

The MMS API enables applications to send MMS messages and get the delivery status of the messages from the network. If an application hosts a service that complies with an AT&T specified callback, the application can also receive MMS messages. AT&T will deliver MMS messages to the application as soon as they arrive on the AT&T network.

**Methods:**

1) Get MMS Delivery Status

2) Send MMS

## Methods

### Get MMS Delivery Status

[getMMSDeliveryStatus](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MMS-method-getMMSDeliveryStatus)( options, success, error ) : Object

Returns the status of a sent MMS message.

#### Parameters

* options : Object

An object containing the following properties:

* + id : String

The MMS request id that is returned by the sendMMS method as part of the response object.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getMMSDeliveryStatus method.

getMMSDeliveryStatus({

'id' : "MMS ID that you get in response of sendMMS.",

"accept" : "application/json",

}, function(data) {

success Callback

}, function(error) {

error Callback

});

### Send MMS

[sendMMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MMS-method-sendMMS)( options, success, error ) : Object

Sends an MMS message to the specified recipient.

The Multimedia Messaging Service allows for the delivery of different file types. Please see the [Developer Documentation](https://developer.att.com/developer/tierNpage.jsp?passedItemId=2400428) for an updated list.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + body : Object/String

An Object or XML string containing all of the parameters that a user needs to send a message on an AT&T mobile device. Valid parameters are:

* + - address : String

The MSISDN of the recipient(s).

* + - subject : String (optional)

The subject of the message.

* + - priority : String (optional)

The priority of the message. Valid values are "Default", "Low", "Normal", and "High".

The defaults value is: 'Default'

* + - notifyDeliveryStatus: Boolean (optional)

This is a Boolean flag to denote the gateway whether delivery status notification is expected by the application or not.

If this flag is set to true on the request, then AT&T API Platform gateway would send a push notification to communicate the final delivery status of this message. The notification would be sent to the endpoint URI registered by the developer against the given short code in the application short code management profile. If there is more than one recipient specified in the *address* parameter, then there would be one notification per recipient.

***Note :*** *If this flag is set to true but the developer has not registered any endpoint URI against the given short code, a bad request error would be thrown back in the response.*

If this parameter value is set to false then the application would not receive any delivery status notifications from the AT&T API Platform gateway.

* + attachments : Array

An Array of JSON Objects containing three key-value pairs: {fileName:"*Name of the file*",fileType:"*Type of the file*",filePath:"*Path of the file*"}. The file can be image, audio, video, or text.

* + contentType : String

Specifies the format of the message content. Valid values are:

* + - application/json
    - application/xml
    - application/x-www-form-urlencoded
  + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request. This response object contains the MMS ID that is used in the id property of the options parameter in the getMMSDeliveryStatus method.

**Examples:**

**1.** The following example of the sendMMS method uses a contentType of 'application/json'.

sendMMS({

"body":{ "address" : "tel:xxxxxxxxxx,tel:xxxxxxxxxx", "subject" : "Test MMS", "priority" : "High" },

"contentType" : "application/json",

"accept" : "application/json",

"attachments" : [{'fileName' : "Name of the file",'fileType' : "type of the file like : image/png",'filePath' :"Path of the file"}, {'fileName' : "Name of the file",'fileType' : "type of the file like : image/jpeg",'filePath' :"Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**2.** The following example of the sendMMS method uses a contentType of 'application/xml'.

sendMMS({

"body":"<outboundMessageRequest>" + "\n" + "<address>tel:xxxxxxxxxx</address>" + "\n" + "<subject>Test MMS</subject>" + "\n" + "<priority>High</priority>" + "\n" + "</outboundMessageRequest>",

"contentType" : "application/xml",

"accept" : "application/json",

"attachments" : [{'fileName' : "Name of the file",'fileType' : "type of the file like : image/png",'filePath' :"Path of the file"}, {'fileName' : "Name of the file",'fileType' : "type of the file like : image/jpeg",'filePath' :"Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**3.** The following example of the sendMMS method uses a contentType of 'application/x-www-form-urlencoded'.

sendMMS({

"body":"address=tel%3A%2B1xxxxxxxxxx&priority=High&subject=Test%20MMS",

"contentType" : "application/x-www-form-urlencoded",

"accept" : "application/json",

"attachments" : [{'fileName' : "Name of the file",'fileType' : "type of the file like : image/png",'filePath' :"Path of the file"}, {'fileName' : "Name of the file",'fileType' : "type of the file like : image/jpeg",'filePath' :"Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

*Note: xxxxxxxxxx represents an AT&T wireless number.*

# [ATT.MOBO](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Messaging on Behalf of (MoBo) API enables your application to send and retrieve SMS and MMS messages on behalf of your customers’ regular AT&T mobile phone numbers.

**Description:**

The MoBo API allows applications to send and receive SMS and MMS on behalf of a subscriber if that subscriber has given consent. When sending a message on behalf of a subscriber, that subscriber's MSISDN is retrieved from the subscriber’s consent information. The message is then forwarded to the recipients using the subscriber's MSISDN as the sender address. A developer can make a single API request and the AT&T system will determine if the request is an SMS or an MMS message and select the correct transport mechanism accordingly.

Some major advantages of using the Messaging on Behalf of API are:

Sending messages from your application: The MoBo Send Message operation allows your customers to send text and picture messages to any U.S. mobile phone using their own AT&T phone number, from within your application. Message recipients can immediately recognize who sent the message.   
Receiving messages in your application: Responses to the sent messages can be received by the user’s mobile phone or by your application using the Get Message Headers and Get Message Content operations. Your application can display messages received by the user’s mobile phone, allowing the conversation to continue from within your application.

**Methods:**

1) Get Message Content

2) Get Message Headers

3) Send Message

## Methods

### Get Message Content

[getMessageContent](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MOBO-method-getMessageContent)( options, success, error ) : Object

Enables an application to retrieve media for one or more Subscriber Messages from the AT&T Messages environment using information returned by the getMessageHeaders method.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + messageId : String

A message identifier representing a Subscriber Message in the AT&T Messages environment.

* + partNumber : String (optional)

A content identifier representing an attachment in the referenced Subscriber Message.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getMessageContent method.

getMessageContent({

'messageId' : "MoBo message ID that you get in response of sendMessage",

'partNumber' : '1'

}, function(data) {

success Callback

}, function(error) {

error Callback

});

### Get Message Headers

[getMessageHeaders](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MOBO-method-getMessageHeaders)( options, success, error ) : Object

Enables an application to retrieve meta-data for one ore more Subscriber Messages from the AT&T Messages environment.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + headerCount : String

The number of Headers to be returned:

Valid Range: Min = 1, Max = 500

The value of headerCount is relative to the most recent Subscriber message.

If indexCursor is defined, headerCount will start with the first message defined by indexCursor.

* + indexCursor : String (optional)

Defines an index value at which the headerCount starts.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getMessageHeaders method.

getMessageHeaders({

'accept' : 'application/json',

'headerCount' : 500 //Valid Range: Min = 1, Max = 500 // Should be integer

'indexCursor' : '123' //Define an index value for which HeaderCount will start.

}, function(data) {

success Callback

}, function(error) {

error Callback

});

### Send Message

[sendMessage](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.MOBO-method-sendMessage)( options, success, error ) : Object

Enables an application to send messages on behalf of an AT&T subscriber. Each time the sendMessage method is invoked, a single message can be sent to one or more destination devices. Messages are processed synchronously and sent asynchronously to the destination.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + Addresses : String

The MSISDN of the recipient, or a comma separated list of MSISDNs for multiple recipients. For example: tel:6507038846,tel:6507038847.

* + Subject : String

The subject of the message.

* + Text : String

The text portion of the message . If the request is detected to be MMS then the following character sets will be supported:

-ASCII

-UTF-8

-UTF-16

-ISO-8859-1

If the request is detected to be SMS then the following character set will be supported:

-ISO-8859-1

Note: This data becomes mandatory if a value for the attachment parameter is NOT provided in the request.

* + attachments : Array

An Array of JSON Objects that contains three key-value pairs: {fileName:"*name of the file*",fileType:"*Type of the file*",filePath:"*Path of the file*"}The file can be image, audio, video, or text.

* + contentType : String

Specifies the format of the message content. Valid values are:

* + - application/json
    - application/xml
    - application/x-www-form-urlencoded
  + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Examples:**

**1.** The following example of the sendMessage method uses a contentType of 'application/json'.

sendMessage({

"body":{ "Addresses": ["mail@mail.mail","mail@mail.mail","tel:xxxxxxxxxx"], "Text": "Hello world", "Subject": "Hi" },

"contentType" : "application/json",

"accept" : "application/json",

"attachments" : [{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"},{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**2.** The following example of the sendMessage method uses a contentType of 'application/xml'.

sendMessage({

"body":"<MessageRequest>"+"\n"+"<Addresses>tel:xxxxxxxxxx,tel:xxxxxxxxxx,mail@mail.mail</Addresses>"+"\n"+"<Text>Hello world</Text>"+"\n"+"<Subject>Hi</Subject>"+"\n"+"</MessageRequest>",

"contentType" : "application/xml",

"accept" : "application/json",

"attachments" : [{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"},{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**3**. The following example of the sendMessage method uses a contentType of 'application/x-www-form-urlencoded'.

sendMessage({

"body":"Addresses=tel%3A%2Bxxxxxxxxxx&Addresses=tel%3A%2Bxxxxxxxxxx&Addresses=mail@mail.mail&Text=Hello&Subject=Hi",

"contentType" : "application/x-www-form-urlencoded",

"accept" : "application/json",

"attachments" : [{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"},{"fileName" : "Name of the file","fileType" : "type of the file like : image/png","filePath" : "Path of the file"}]

}, function(data) {

success Callback

}, function(error) {

error Callback

});

*Note: xxxxxxxxxx represents an AT&T wireless number*

# [ATT.Notary](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Notary API is used to digitally sign content or data The digital signing mechanism of the Notary API uses a token hash, a time stamp, and a nonce, to ensure that the sender of the data is who they say they are, and that the associated content indisputably came from the application that sent it andnot from an imposter.

**Description:**

The Notary API is used to encode and sign documents in a secure and tamper resistant manner. This operation must be used to encode and secure the payment detail information when using the New Transaction and New Subscription methods of the Payment API.

**Usage**

The signedPayload method must be used to generate the inputs for the Payment.newTransaction and Payment.newSubscription methods.

**Methods**

1) SignedPayload

## Methods

### SignedPayload

[signedPayload](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Notary-method-signedPayload)( options, success, error ) : Object

This API method treats the payload as an arbitrary set of text content. This method operation is used in conjunction with the Payment.newTransaction or Payment.newSubscription invocations of the AT&T Payment API.

When calling the signedPayload method, your application is required to identify which of the standard HTTP content types (such as XML or JSON) is used to encode the document. Upon receipt of the payload body, the AT&T API Gateway will repackage the content into a standard format before encoding and signing the payload for a response. After the AT&T API unpacks and validates the payload content, any parameters associated with the payload are determined and the content is repackaged. The content type of the result is independent of the original content type as provided by the application.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + data : Object

A JSON Object containing the following properties:

* + - Amount : String

The amount of the transaction expressed with a maximum of two decimal places. For example: 0.99 or 2.99.

* + - Category : String

The Product Category. One of the values from a predefined list of categories.

* + - Channel : String

The channel through which the payload is delivered. The only supported value is MOBILE\_WEB.

* + - Description : String

A short description of the entire Purchase. The maximum length of the String is 128 characters.

* + - MerchantTransactionId : String

The transaction ID in the merchant’s system. The ID must be unique for every purchase request, and consist of alphanumeric text.

* + - MerchantProductId : String

The product identifier, assigned by the merchant, of the product to be purchased. The maximum length of the String is 50 alphanumeric characters.

* + - MerchantPaymentRedirectUrl : String

The URL where control is returned to your application by the AT&T Gateway. The merchant application must listen at this URL for redirects from the subscriber’s browser.

* + - MerchantSubscriptionIdList : String

A list of merchant subscription identifiers. The maximum length of the string is 12 alphanumeric characters.

* + - IsPurchaseOnNoActiveSubscription : String

Indicates whether the purchase was made on a non-active subscription. For this release, the value of this property must be set to FALSE.

* + - SubscriptionRecurrences : String

Specifies the number of subscription renewals that should be made. For this release, the value of this property must be set to 99999.

* + - SubscriptionPeriod : String

The interval at which the subscription is charged. For this release, the value of this property must be set to MONTHLY.

* + - SubscriptionPeriodAmount : String

The number of periods, as specified in the SubscriptionPeriod property, that pass between each renewal. For this release, the value of this property must be set to 1.

* + contentType : String

The content type of the request. Valid values are::

* + - application/json
    - application/xml
    - application/x-www-form-urlencoded
  + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* + clientId : String

The API Key that is assigned to the application when the developer creates an application account at <https://devconnect-api.att.com/>.

* + clientSecret : String

The Secret Key that is assigned to the application when the developer creates an application account at <https://devconnect-api.att.com/>.

* + contentLength : Number

The size of the entity-body, specified as a decimal number of OCTETs (For IOS Only).

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the signedPayload method.

signedPayload({

'data' : {"Amount" : <value>,

"Category" : <value>,

"Channel" : <value>,

"Description" : <value>,

"MerchantTransactionId" : <value>,

"MerchantProductId" : <value>,

"MerchantPaymentRedirectUrl" : <value>,

"MerchantSubscriptionIdList" : <value>,

"IsPurchaseOnNoActiveSubscription" : <value>,

"SubscriptionRecurrences" : <value>,

"SubscriptionPeriod" : <value>,

"SubscriptionPeriodAmount" : <value>},

'clientId' : <accessKey value>,

'clientSecret' : <secretKey value>,

'contentType' : 'application/json'

},function(data) {

success Callback

}, function(error) {

error Callback

});

# [ATT.OAuth](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The AT&T OAuth service provides a safe and secure way for AT&T subscribers to access AT&T network resources through a third-party application without the risk of compromising security. The OAuth service ensures that secure, sensitive, and private AT&T subscription related details are not exposed to the third-party application.

OAuth authorization management is an open standard recognized as providing a strong protection to clients and applications. The OAuth service provides developers with a security model that allows them to obtain an authorization code and an access token. By acquiring an authorization code, a resource owner has authorized an application to access a protected resource on their behalf. The receipt of an access token enables an application to access a protected resource on behalf of a resource owner via the AT&T network.

The AT&T OAuth service allows third party applications access to the private resources of subscribers without requiring the end-user to provide credentials (such as username and password) to the third party application. An application that is implemented with Oauth, provides a closer and more secure integration with the AT&T API Platform.

**Methods:**

1) Authorize

2) Obtain End-User Authorization

## Methods

### authorize

[authorize](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.OAuth-method-authorize)( accessKey, secretKey, scope, grantType, oAuthCode )

**Introduction:**

Authorizes an application so that it can use specific AT&T APIs.

This method must be directly called using the ATT namespace as in the following example: **Att.authorize('apiKey','secretKey','scope','grantType','oAuthCode')**

When authorizing an application for use with the Location and MOBO APIs, the user must send their access key, secret key, scope, grantType, and OAuthCode. **(**

#### Parameters

* accessKey : String

The API Key that is assigned to the application when the developer creates an application account at <https://devconnect-api.att.com/>.

* secretKey : String

The Secret Key that is assigned to the application when the developer creates an application account at <https://devconnect-api.att.com/>.

* scope : String

A comma separated list of authScopes enumeration values that specify which APIs the app requires access to.

* grantType : String

Specifies the mechanism used to obtain the access token. The following grantType values must be used for the following cases: Web-server client applications using the authorization code to obtain the access token must set this parameter value to authorization\_code. Autonomous client applications that use just the API key and its secret to obtain the access token, must set this parameter valueto client\_credentials. When client applications of both types mentioned above) use the refresh\_token to obtain a new access\_token after the expiry of the initial access\_token, this parameter value must be set to refresh\_token.

* oAuthCode : String

The authorization code. The oAuthCode is returned in the response value of the obtainEndUserAuthorization method. Currently, this parameter is only required when using the Location and MOBO APIs. **Example:**

The following is an example of the authorize method.

Att.authorize('apiKey','secretKey','scope','grantType','oAuthCode');

### obtainEndUserAuthorization

[obtainEndUserAuthorization](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.OAuth-method-obtainEndUserAuthorization)( options, success, error ) : String

The Obtain End-User Authorization operation is the initial operation in the OAuth call flow. It is invoked by an application that requires subscriber authorization in order to obtain an OAuth access token. The application request for an OAuth access token is forwarded by redirecting the subscriber’s browser to the AT&T Gateway OAuth Server.

**Important Note:** Currently, this method call is only necessary and supported for applications attempting to consume the following APIs

1) Location API

2) Message on Behalf of (MoBo) API

#### Parameters

* options : Object

A JSON object containing the following properties :

* + clientId : String

Tthe API Key that is assigned to the application when the developer creates an application account at <https://devconnect-api.att.com/>.

* + scope : String

Specifies the services that the application is requesting the end-user to provide consent for. The value of the scope parameter is a comma delimited list containing one or more of the following values that represent the services that are in scope.

• Terminal Location – TL.

• Message On Behalf-Of – MOBO

* + redirectUrl : String (optional)

The URL where the subscriber’s browser will be redirected following completion of the authorization process. If this parameter is not present in the request, the AT&T Gateway will use the value of provided OAuth Redirect URL thatis provided by the developer in the AT&T API Platform console when the application is registered.

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* String

Specifies a query parameter that is included with the redirect\_uri parameter of the authorize method in the original OAuth request.

**Example:**

The following is an example of the obtainEndUserAuthorization method.

obtainEndUserAuthorization({

'clientId' : 'API Key',

'scope' : 'scope values',

'redirectUrl' : 'redirect url'

},function(data) {

success Callback

}, function(error) {

error Callback

});

# [ATT.Payment](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Payment API provides developers and merchants the ability to charge digital services directly to an AT&T subscriber bill for payment.

**Description:**

Using the Payment API, a merchant can create new transactions and subscriptions, get the status of the transaction or subscription, and authorize refunds.

The new transaction and subscription process includes:

* Authentication of the end-user as an AT&T Subscriber.
* Presentation of an Advice-of-Charge (AOC) page and the opportunity to cancel.

If the Subscriber chooses to accept the Charge, a Thank You message is presented.

The merchant also has the option of automatically receiving notifications of refunds and subscription changes triggered through other AT&T systems. Applications in the initial Sandbox state have lower limits on the quantity of transactions. All Sandbox payment transactions are refunded weekly. Further, in the Sandbox environment, a developer may set certain attributes for the application that would not be available to the developer in the Production environment. Once an application has been promoted to the Production state, however, a more secure environment is maintained where most of the attributes are read-only and can only be changed by the AT&T administrator. Organizations must have a Merchant Account set up for the Payment Service to work. If your organization does not have a Merchant Account please contact your Organizational Profile Administrator to arrange for one.

**Usage**

Use the Notary API to sign the payment request.

Redirect the user agent to the AT&T server to obtain consent.

Get the response, either an authorization code or an error, through the redirect back from the consent process.

Use the getTransactionStatus or getSubscriptionStatus methods with the authorization code to get the transaction id which is needed for refunds.

**Methods**

1) Acknowledge Notification

2) Get Notification

3) Get Subscription Details

4) Get Subscription Status

5) Get Transaction Status

6) New Subscription

7) New Transaction

8) Refund Transaction

## Methods

### Acknowledge Notification

[acknowledgeNotification](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-acknowledgeNotification)( options, success, error ) : Object

Acknowledges the receipt of a notification and requests that further notifications be stopped for aparticular NotificationID.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + notificationID : String

The ID of the notification to be retrieved.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the acknowleadgeNotification method.

acknowleadgeNotification({

'notificationId' : 'ID of the notification that will be retrieved.'

},function(data) {

success Callback

}, function(error) {

error Callback

});

### Get Notification

[getNotification](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-getNotification)( options, success, error ) : Object

Retrieves a notification object that has been received in a prior notification message as a result of having the server listenin for notification messages.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + notificationID : String

The ID of the notification to be retrieved.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getNotification method.

getNotification({

'notificationId' : 'ID of the notification that will be retrieved.'

},function(data) {

success Callback

}, function(error) {

error Callback

});

### Get Subscription Details

[getSubscriptionDetails](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-getSubscriptionDetails)( options, success, error ) : Object

Retrieves the details for a merchant subscription to which a consumer has subscribed. Merchants who manage consumers can use this method to determine the details of a subscriptions for a particular consumer.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + consumerId : String

Identifies the consumer that is to use the subscription. The ConsumerId is a unique user id generated in the payment system representing a unique subscriber.

* + merchantSubscriptionId : String

Identifies the the merchant’s subscription. This must be a unique value for each merchant subscription.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getSubscriptionDetails method.

getSubscriptionDetails({

'consumerId' : <returns in response of Payment.getSubscriptionStatus>,

'merchantSubscriptionId' : <returns in response of Payment.getSubscriptionStatus>,

},function(data) {

success Callback

}, function(error) {

error Callback

});

### Get Subscription Status

[getSubscriptionStatus](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-getSubscriptionStatus)( options, success, error ) : Object

Retrieves the status of a previously executed call to the newSubscription method. The TransactionId from the previously executed newSubscription method call is used to retrieve the status of the subscription. This means that the TransactionIds from any new subscriptions must be stored in order to retrieve the status of those subscriptions. This method returns the status as well as several other fields related to the purchase.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + transactionId : String

(Conditional Parameter) The value of this parameter is the transaction ID generated by AT&T Payment processing system for a purchase. This value is used for future transaction requests.

* + merchantTransactionId : String

(Conditional Parameter) The value of this parameter is the transaction ID in the merchant’s system identifying the transaction. This ID must be unique on every purchase request.

* + transactionAuthCode : String

(Conditional Parameter) The value of this parameter is returned by AT&T to the Developer’s Application Service upon successful conclusion of the Payment.newTransaction request.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml \*

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getSubscriptionStatus method.

getSubscriptionStatus({

"subscriptionAuthCode" : <The value returned by AT&T to the Developer’s Application Service at upon successful conclusion of the Payment.newSubscription request.>

},function(data) {

success Callback

}, function(error) {

error Callback

});

Note : To get the SubscriptionStatus, pass any of the following IDs:

1) "subscriptionAuthCode" - The value returned by AT&T to the Developer’s Application Service upon successful conclusion of the Payment.newSubscription request.

2) "merchantTransactionId" - The MerchantTransactionId specified in the initial call to the Notary.signedPayload method.

3) "transactionId" - The TransactionId returned by a previous call to getTransactionStatus method.

### Get Transaction Status

[getTransactionStatus](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-getTransactionStatus)( options, success, error ) : Object

Retrieves the status and all relevant information of a previously executed transaction.

Three different identifiers may be used:

* TransactionId returned by a previous call to the getTransactionStatus method.
* TransactionAuthCode returned at the end of the Merchant Payment Acknowledgement process.
* MerchantTransactionId specified in the initial call to the Notary.signedPayload method.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + transactionId : String

(Conditional Parameter) The value of this parameter is the transaction ID generated by AT&T Payment processing system for a purchase. This is used for future transaction requests.

* + merchantTransactionId : String

(Conditional Parameter)The value of this parameter is the transaction ID in the merchant’s system identifying the transaction. This ID must be unique on every purchase request.

* + transactionAuthCode : String

(Conditional Parameter) The value of this parameter is the value returned by AT&T to the Developer’s Application Service upon successful conclusion of the Payment.NewTransaction request.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getTransactionStatus method.

getTransactionStatus({

"transactionAuthCode" : <The value returned by AT&T to the Developer’s Application Service at upon successful conclusion of the Payment.newTransaction request>

},function(data) {

success Callback

}, function(error) {

error Callback

});

Note: To get the TransactionStatus, pass any of the following IDs:

1) "transactionAuthCode" - The value returned by AT&T to the Developer’s Application Service upon successful conclusion of the Payment.newTransaction request.

2) "merchantTransactionId" - The MerchantTransactionId specified in the initial call to the Notary.signedPayload method.

3) "transactionId" - The TransactionId returned by a previous call to getTransactionStatus method.

### New Subscription

[newSubscription](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-newSubscription)( options, success, error )

The newSubscription method sets up recurring charges for a specified amount, at a specified interval, for a specified number of times.

he method authorizes both the merchant and the user of the subscription, and initializes the product and price data for of the transaction.

Before this operation can be invoked, the application must call the Notary.signedPayload method and provide a payload containing the required parameters (i.e. name-value-pairs) necessary to complete the purchase process.

The Notary.signedPayload method will return the result in either JSON or XML format, as defined in the HTTP Header "Accept" included in the request to sign the document. The result will contain two name-value-pairs which will be utilized in formulating the request for this method.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + signedDocument : String

The value of this parameter should be the value of SignedDocument that is returned by the Notary.signedPayload method.

* + signature : String

The value of this parameter should be the value of Signature that is returned by Notary.signedPayload method.

* + clientId : String

The Client Id of the application making the request.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

**Sample Example**

newTransaction({

"signedDocument" : <Use value of SignedDocument returned by Notary.signedPayload>,

"signature" : <Use value of Signature returned by Notary.signedPayload>,

"clientId" : <accessKey value>

},function(data) {

success Callback

}, function(error) {

error Callback

});

### New Transaction

[newTransaction](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-newTransaction)( options, success, error )

Creates a new payment transaction for a one time product purchase (i.e. transactionType = SINGLE\_PURCHASE).

Before this method can be invoked, the application must call Notary.signedPayload and provide a payload containing the parameters (i.e. name-value-pairs) necessary to complete the purchase process.

The Notary.signedPayload method will return the result in either JSON or XML format as designed by the HTTP Header Accept included in the request to sign the document. The response will contain two name-value-pairs which will be utilized in formulating the request for this method.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + signedDocument : String

The value of this parameter should be the value of SignedDocument that is returned by the Notary.signedPayload method.

* + signature : String

The value of this parameter should be thevalue of Signature that is returned by the Notary.signedPayload method.

* + clientId : String

The Client Id of the application making the request.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

**Sample Example**

newTransaction({

"signedDocument" : <Use value of SignedDocument returned by Notary.signedPayload>,

"signature" : <Use value of Signature returned by Notary.signedPayload>,

"clientId" : <accessKey value>

},function(data) {

success Callback

}, function(error) {

error Callback

});

### Refund Transaction

[refundTransaction](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Payment-method-refundTransaction)( options, success, error ) : Object

Refunds the value of a transaction to the original consumer. The refundTransaction method can be invoked through the merchant application and will automatically pass without additional approvals.

There are specific limits configured on the number of refunds within a day, week, and month. These limits are configured for a merchant account in the AT&T payment system.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + transactionOperationStatus : String

The value of this parameter must be "Refunded".

* + refundReasonCode : String

A code that specifies a reason for the refund. Must be one of the values in a predefined list.

* + refundReasonText : String

A description of the reason for the refund.

* + transactionId : String

The original transaction ID of the transaction for the refund request.

* + action : String

Specifies an action. The value of this parameter must always be set to "refund".

* + contentType : String

Specifies the type of content to be sent, Valid values are: application/json, application/xml, application/x-www-form-urlencoded.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns success.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Examples:**

**1.** The following example of the refundTransaction method uses a contentType of 'application/json'.

refundTransaction({

"body":{"TransactionOperationStatus":<value>, "RefundReasonCode":value, "RefundReasonText":<value>},

"contentType" : "application/json",

'transactionId' : TransactionId returned by a call to getTransactionStatus,

'action' : <action value>

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**2.** The following example of the refundTransaction method uses a contentType of 'application/xml'.

refundTransaction({

"body":"<RefundTransactionRequest><TransactionOperationStatus>value</TransactionOperationStatus><RefundReasonCode>value</RefundReasonCode> <RefundReasonText>value</RefundReasonText></RefundTransactionRequest>",

"contentType" : "application/xml",

'transactionId' : TransactionId returned by a call to getTransactionStatus,

'action' : <action value>

}, function(data) {

success Callback

}, function(error) {

error Callback

});

# [ATT.SMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Short Message Service (SMS) API enables authorized applications to send SMS messages to devices on the AT&T network. The API also allows applications to query the delivery statusof SMS messages sent from an application.

**Description:**

The SMS API enables you to target specific customers with alerts and information. Additionally, the SMS API can give your customers the ability to respond to your messages in near real-time.

**Methods:**

1) Get SMS

2) Get SMS Delivery Status

3) Send SMS

## Methods

### Get SMS

[getSMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.SMS-method-getSMS)( options, success, error ) : Object

Retrieves a list of SMS messages that were sent to the application's short code.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + registrationId : String

The ID of the application which is a short code that is obtained when you register your application with AT&T.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml.

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getSMS method.

getSMS({

'accept' : 'application/json',

'registrationId' : <put registrationId here>

}, function(data) {

success Callback

}, function(error) {

error Callback

});

### Get SMS Delivery Status

[getSMSDeliveryStatus](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.SMS-method-getSMSDeliveryStatus)( options, success, error ) : Object

Checks the status of a sent SMS message.

A final state is reached when the DeliveryStatus field of a DeliveryInfo object is set to one of the following values:

* DeliveredToTerminal
* DeliveryImpossible
* DeliveredToNetwork

A single successful delivery status query can be made for a period of thirty minutes following the time a message DeliveryInfo object has been set to a final state. Thirty minutes after a query of the DeliveryInfo object has returned a final state, the DeliveryInfo object may not be available for query.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + smsId : String

The SMS request id that is returned by the sendSMS method as part of the response object.

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml.

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns success.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the getSMSDeliveryStatus method.

getSMSDeliveryStatus({

'smsId' : <put smsId here>,

'accept' : 'application/json'

}, function(data) {

success Callback

}, function(error) {

error Callback

});

### Send SMS

[sendSMS](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.SMS-method-sendSMS)( options, success, error ) : Object

Sends an SMS message to a recipient.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + body : Object

An Object containing all of the parameters that a user needs to send a message on an AT&T mobile device. Valid parameters are:

* + - address : String

The MSISDN of the recipient(s).

* + - message : String

The text of the message to send.

* + - notifyDeliveryStatus: Boolean (optional)

This is a Boolean flag to denote the gateway whether delivery status notification is expected by the application or not.

If this flag is set to true on the request, then AT&T API Platform gateway would send a push notification to communicate the final delivery status of this message. The notification would be sent to the endpoint URI registered by the developer against the given short code in the application short code management profile. If there is more than one recipient specified in the *address* parameter, then there would be one notification per recipient.

***Note :*** *If this flag is set to true but the developer has not registered any endpoint URI against the given short code, a bad request error would be thrown back in the response.*

If this parameter value is set to false then the application would not receive any delivery status notifications from the AT&T API Platform gateway.

* + contentType : String

Valid values are:

* + - application/json
    - application/xml
    - application/x-www-form-urlencoded
  + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request. This response object contains the SMS ID that is used in the smsId property of the options parameter in the getSMSDeliveryStatus method.

**Examples:**

**1.** The following example of the sendSMS method uses a contentType of 'application/json'.

sendSMS({

'body':{"message":"Test AT&T REST", "address":"tel:xxxxxxxxxx"},//For Multiple Phone Number :- "address":["tel:xxxxxxxxxx","tel:xxxxxxxxxx"]

'accept' : 'application/json',

'contentType' : 'application/json'

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**2.** The following example of the sendSMS method uses a contentType of 'application/xml'.

sendSMS({

"body":"<outboundSMSRequest><address>tel:xxxxxxxxxx</address><message>Test XML </message></outboundSMSRequest>",

'accept' : 'application/json',

'contentType' : 'application/xml'

}, function(data) {

success Callback

}, function(error) {

error Callback

});

**3.** The following example of the sendSMS method uses a contentType of 'application/x-www-form-urlencoded'.

sendSMS({

"body":"address=tel%3A%2B1xxxxxxxxxx&address=tel%3A%2B1xxxxxxxxxx&message=URL%20ENCODED",

'accept' : 'application/json',

'contentType' : 'application/x-www-form-urlencoded'

}, function(data) {

success Callback

}, function(error) {

error Callback

});

*Note: xxxxxxxxxx represents an AT&T wireless number*

# [ATT.Speech](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The Speech API enables authorized applications to transcribe speech (audio) to text. The developer can influence the results by passing a speech context in the request.

**Description:**

The Speech API is a service which accepts audio data and returns the text representation of that audio. The text output can then be processed by different speech contexts to produce text results that represent that speech context. Each speech context is tuned to produce results more suitable for a given scenario.

**Methods:**

1) Speech To Text

## Methods

### Speech To Text

[speechToText](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.Speech-method-speechToText)( options, success, error ) : Object

Returns a text translation of the specified audio file using the specified speech context. The audio file must have one of the following sets of constraints: 16 bit PCM WAV, single channel, 8 kHz sampling, or AMR (narrowband), 12.2 kbit/s, 8 kHz sampling.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + filePath : String

The Path of the audio file that will to be translated to text. (For Android, the path should be in this format: /mnt/sdcard/.../fileName.fileExtension).

* + transferEncoding : String

This header parameter is only required for a streaming request, and should be set to chunked.

* + xSpeechContext : String (optional)

Specifies the speech context to be applied to the transcribed text. Valid values are:

* + - Generic
    - TV
    - BusinessSearch
    - Websearch
    - SMS
    - Voicemail
    - QuestionAndAnswer
    - Gaming
    - SocialMedia

If **TV** context is chosen, then the X-Arg parameter **Search** must be defined.

The default value is: "Generic"

* + contentType : String

Must be set to one of the following values:

* + - audio/wav (or audio/x-wav)
    - audio/amr
    - audio/amr-wb
    - audio/x-speex
    - audio/x-speex-with-header-byte;rate=16000
    - audio/x-speex-with-header-byte;rate=8000
    - audio/raw;coding=linear;rate=16000;byteorder=LE
    - audio/raw;coding=linear;rate=16000;byteorder=BE
    - audio/raw;coding=linear;rate=8000;byteorder=LE
    - audio/raw;coding=linear;rate=8000;byteorder=BE
    - audio/raw;coding=ulaw;rate=16000
    - audio/raw;coding=ulaw;rate=8000

**Note**: the API Gateway shall not modify this header when passed onto the Speech enabler, except to remove white space in order to ensure consistent use of parameters

* + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* + contentLength : Number

**(IOS only)** This header parameter is only required for a **non-streaming request.**

* + contentLanguage: String (optional)

Only Specifies the language of the submitted audio. Only used when xSpeechContext parameter is set to “Generic”. Choose one of the following two choices:

* “en-US” (English - United States)
* “es-US” (Spanish - United States)

If Content-Language is not specified, the default shall be en-us.

* + xSpeechSubContext: String (optional)

Only used when xSpeechContext parameter is set to “Gaming”.

* + xArgs: String/Object (optional)

If a string is passed in xArgs should be a comma separated URL encoded list to define multiple name/value pairs. Otherwise pass in a flat object of name/value pairs. If xArgs is an object then the values passed in will be URL encoded

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Example:**

The following is an example of the speechToText method.

speechToText({

'filePath':'Path of audio file'//In android,Path should be like : /mnt/sdcard/.../example.wav.

'xSpeechContext' : 'Generic',

'contentType' : 'audio/wav',

'contentLength' : length,//For IOS only.

'accept':'application/json',

'contentLanguage': 'es-US'

},function(data) {

success Callback

}, function(error) {

error Callback

});

# [ATT.WAPPush](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html)

**Introduction:**

The WAP Push API is an ideal technology for targeting the users of legacy devices that have limited screen sizes or input capabilities with text or WAP formatted pages.

**Description:**

The WAP Push API enables suitably authorized applications to send WAP Push requests to supported devices on the AT&T Mobility network. The WAP Push API provides support for applications to send Service Indication (SI) type requests to the Wireless Access Protocol (WAP) Push to supported devices on the AT&T Mobile network.

**Methods:**

1) Send WAP Push

## Methods

### Send WAP Push

[sendWAPPush](file:///C:\Documents%20and%20Settings\tp1550\Desktop\2012%20ARO%20Documentation%20Updates\Titanium%20Module%20Documentation\ATT_API_Doc_Kevin\attDocs\index.html#!/api/ATT.WAPPush-method-sendWAPPush)( options, success, error ) : Object

Sends a WAP Push message.

#### Parameters

* options : Object

A JSON object containing the following properties:

* + body : Object

An Object containing all of the parameters that a user needs to send on an AT&T mobile device. Valid parameters are:

* + - address : String

The MSISDN of the recipient(s).

* + data : String

An Attached XML.

* + contentType : String

The content type of the request. Valid values are:

* + - application/json
    - application/xml
    - application/x-www-form-urlencoded
  + accept : String (optional)

Specifies the format of the body of the response. Valid values are:

* + - application/json
    - application/xml

The default value is: 'application/json'

* success : Function

The callback function that is called when the method returns success.

* error : Function

The callback function that is called when the method returns an error.

#### Returns

* Object

An object containing the response. The format of the response (JSON or XML) is specified by the value of the accept parameter in the request.

**Examples:**

1. The following example of the sendWAPPush method uses a contentType of 'application/json'.

sendWAPPush({

"body":{ "address": "tel:xxxxxxxxxx"},

"contentType" : "application/json",

"accept" : "application/json",

"data" : "xml data"

}, function(data) {

success Callback

}, function(error) {

error Callback

});

2. The following example of the sendWAPPush method uses a contentType of 'application/xml'.

sendWAPPush({

"body":"<wapPushRequest>"+"\n"+"<address>tel:xxxxxxxxxx,tel:xxxxxxxxxx</address>"+"\n"+"</wapPushRequest> ",

"contentType" : "application/xml",

"accept" : "application/json",

"data" : "xml data"

}, function(data) {

success Callback

}, function(error) {

error Callback

});

3. The following example of the sendWAPPush method uses a contentType of 'application/x-www-form-urlencoded'.

sendWAPPush({

"body":"address=tel%3Axxxxxxxxxx",

"contentType" : "application/x-www-form-urlencoded",

"accept" : "application/json",

"data" : "xml data"

}, function(data) {

success Callback

}, function(error) {

error Callback

});

*Note: xxxxxxxxxx represents an AT&T wireless number*