Updated: **28 August 2020**

axxess™
www.internet.co.za

Get your company connected to the Axxess Reseller API

This document will outline and provide you with all the necessary technical information.

Products available via the API

Mobile Data	Fibre Combos	ADSL Data
ADSL Lines	Axxess Voice (VoIP)	Fixed Wireless

The API will allow you to:

< Create new client profiles and assign available products >

< Extract currently assigned and unassigned SIMS >

< Extract a list of services for a specific client >

< Extract details which include detailed diagnostics such as data session details >

Please Note: The API may be subjected to a hourly limitation on calls. On detecting possible abuse of the API or in the event that a limitation is introduced and this limit is exceeded, a temporary suspension or a limit on accepted requests may be enforced to conserve network resources and to ensure the overall quality of the web service.

☎ 087 821 1100 ✉ businesspartner@internet.co.za https://apitest.axxess.co.za/

Table of Contents

Table of Contents continues...

Out Parameters for all requests			
intCode	String	JSON string with error break down	http code: http://www.w3.org /Protocols /rfc2616 /rfc2616-sec10.html
strMessage	String	Extra info on returned data, or error	
strStatus	String	Header status as per httpd code	e.g. OK; Internal Service Error

1. Authentication

Basic auth is required as part of every request. The username and password will be made available on request along with a bootstrap file written in PHP, which has done the client side requests through a Curl object implementation. View examples at the end of this document.

1.1 getSession (Method=GET)

To start using our Reseller API you first have to request a session id. This will allow you to interact with the API for 1 hour, before the session id expires.

Name	Type	Description	Values
In Parameters			
strUserName	String(64)	(*Required) Control Panel Username	FLE123
strPassword	String(64)	(*Required) Control Panel Username	Wer33khh
Out Parameters			
strSessionId	String(36)	This is a generated GUID that allows the reseller to interact with the API	369a1e34-4d6d-11e4-a967-002590383a96

1.2 checkSession (Method=GET)

To check if your session id is still active call this function and it will return if your session has expired.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
strSessionId	String(36)	This is a generated GUID that allows the reseller to interact with the API	369a1e34-4d6d-11e4-a967-002590383a96

2. Clients

2.1 getProvinces (Method=GET)

This function returns a list of all the provinces on our system. Use this id on creation to set the province of your client.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
arrProvinces	Array	Array break down listed below	
intProvinceId	Integer		1
strName	String(50)	Name	The display name for the province. Eg. Port Elizabeth

2.2 getAllClients (Method=GET)

This function returns an array with the details of all your clients.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
arrClients	Array	Array break down listed below	
guidClientId	String(36)		1
strName	String(50)	Name	Combined names for the client. First Name + Middle Name
strLastName	String(50)		
strCell	String(10)	Cellphone number	0812244522
intIdNumber	String(13)	Identification number	
strAddress	String(50)	Address of the client	
strSuburb	String(50)	Suburb of the client	
strCity	String(50)	City of the client	
strEmail	String(50)	Email of the client	
strHomeTel	String(10)	Home telephone number of the client	0413968000
strWorkTel	String(10)	Work telephone number of the client	0413968000
intProvinceId	Integer	The id linked to the province.	
intPostalCode	Integer	Postal Code of the client	
intIsActive	Integer	Flag that indicates whether a client is active (1) or inactive (0)	1 or 0

2.3 getClientById (Method=GET)

This function returns client details.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidClientId	String(36)	(*Required)	
Out Parameters			
arrClients	Array	Array break down listed below	
guidClientId	String(36)		1
strName	String(50)	Name	Combined names for the client. First Name + Middle Name
strLastName	String(50)		
strCell	String(10)	Cellphone number	0812244522
strIdNumber	String(13)	Identification number	
strAddress	String(50)	Address of the client	
strSuburb	String(50)	Suburb of the client	
strCity	String(50)	City of the client	
strEmail	String(50)	Email of the client	
strHomeTel	String(10)	Home telephone number of the client	0413968000
strWorkTel	String(10)	Work telephone number of the client	0413968000
intProvinceId	Integer	The id linked to the province.	
intPostalCode	Integer	Postal Code of the client	
intIsActive	Integer	Flag that indicates whether a client is active (1) or inactive (0)	1 or 0

2.4 createClient (Method=PUT)

This function creates a client profile and returns a client identifier to be used for service creation.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identifier	
strName	String(50)	(*Required) Name	First name of the client
strLastName	String(50)	(*Required) Last Name	Last name of the client
strCell	String(10)	Cellphone number	0812244522
strIdNumber	String(13)	Identification number	
strCompanyName	String(50)	Company of the client	
strAddress	String(50)	Address of the client	
strSuburb	String(50)	Suburb of the client	
strCity	String(50)	City of the client	

2.4 createClient (continues)...

This function will return a list of all the provinces we have on our system. You can use this id to set the province of your client on creation.

Name	Type	Description	Values
In Parameters			
strEmail	String(50)	Email of the client	
strHomeTel	String(10)	Home telephone number of the client	0413968000
strWorkTel	String(10)	Work telephone number of the client	0413968000
intProvinceId	Integer	The id linked to the province.	
intPostalCode	Integer	Postal Code of the client	
Out Parameters			
guidClientId	String(36)	The id that is used to retrieve a client or retrieve services of a client	
strUsername	String (50)	The client code	
strPassword	String (50)	The password	

2.5 updateClient (Method=PUT)

This function is used to update your client details. You are required to send through your strSessionId, guidClientId, strFirstName and strLastName. All other paramteres are optional. NOTE: Any optional paramter not passed through will automatically overwrite the field with an empty string.

Name	Type	Description	Values
In Parameters			
StrSessionId	String(36)	(*Required) Session Identifier	
guidClientId	String(36)	(*Required) Client Identifier	
strFirstName	String(36)	(*Required) First Name	
strLastName	String(36)	(*Required) Last Name	
strIdNumber	String(36)	ID Number	
strEmail	String(36)	Email Address	
strCell	String(36)	Cell Number	
strHomeTel	String(36)	Home Tel. Number	
strWorkTel	String(36)	Work Tel. Number	
strFax	String(36)	Fax Number	
strCompanyName	String(36)	Name of Company	
strAddress	String(36)	Street Address	
strSuburb	String(36)	Suburb	
strCity	String(36)	City	
intProvinceId	Int	Province ID	Retrieved from getProvinces function (Between 1 and 9)
intPostalCode	Int	Postal Code	
Out Parameters			
intReturnCode	String	JSON String with Error breakdown	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

3. Services

3.1 getProducts (Method=GET)

This function is used to retrieve all the products to be used when creating a service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
arrProducts	Array		
guidProductId	String(36)	The product id that will be used to create services	
strName	String(50)	This is the display name	Uncapped 4mb

3.2 getServicesByClient (Method=GET)

This function is used to retrieve all the service details for a particular client.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidClientId	String(36)	(*Required) id of the client that you want to retrieve the services for	
Out Parameters			
arrServices	Array	Each index is comprised of the following:	
guidServiceId	String(36)	GUID to identify the service	
guidLinkedServiceId	String	guidServiceId of another service the current service could be linked to	
strDescription	String(50)	The description field will be used for a range of service descripitons	If it is an ADSL service it will be used for the username
guidProductId	String(36)	The product id that the service is linked to	
intSuspendReasonId	Integer	The suspend id that the service is on	This can be null if no suspend reason is set
intCurrentQuantity	Integer	The quantity of the service	This will show the amount of gigs for ADSL products
intQuantity	Integer	The default quantity of the service without any top ups	
strDateEnd	String(50)	The date that the service ends	yyyy-mm-dd
strDateStart	String	Start date (creation date) of the service	

3.3 getAllUnassignedSimServices (Method=GET)

This function is used to retrieve all unassigned sim card services.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
arrServices	Array	Array break down listed below	
guidServiceId	String(36)	GUID to identify the client	
strDescription	String(50)	The description field will be used for a range of service descripitons	If it is an ADSL service it will be used for the username
guidProductId	String(36)	The product id that the service is linked to	
strDateEnd	String(50)	The date that the service ends	yyyy-mm-dd
DateCreated	String	Date the that service was created	

3.4 getAssignedSims (Method=GET)

This function is used to retrieve all assigned sim card services.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrAssignedSims	Array	Array breakdown below:	
guidSimServiceId	String	Sim service identifier	
guidSimProductId	String	Sim product identifier	
strSimProduct	String	Sim product name	
strSimSerialNumber	String	Sim serial number	
strSimCellNumber	String	Sim cell number	
strSimDateCreated	String	Sim creation date	
strDataProduct	String	Sim data product package name	
strUsername	String	Sim data username	
intIsDataCancelled	Int	Sim data service cancelled?	
strDataEndDate	String	Sim data service date cancelled	

3.5 getServiceById (Method=GET)

This function is used to retrieve all details of a particular service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the client that you want to retrieve the services for	
Out Parameters			
guidServiceId	String(36)	GUID to identify the service	
guidClientId	String(36)	GUID to identify the client that the service belongs to	
strDescription	String(50)	The description field will be used for a range of service descripitons	If it is an ADSL service it will be used for the username
guidProductId	String(36)	The product id that the service is linked to	
intSuspendReasonId	Integer	The suspend id that the service is on	This can be null if no suspend reason is set
intQuantity	Integer	The quantity of the service	This will show the amount of gigs for ADSL products
intCurrentQuantity	Integer	The default quantity of the service without any top ups	
strDateEnd	String(50)	The date that the service ends	yyyy-mm-dd
DateStart	String	Start date (creation date) of the service	

3.6 getServiceSessionDetailsById (Method=GET)

This function is used to retrieve session details of a particular service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the client that you want to retrieve the services for	
Out Parameters			
arrSessions			
strUsername	String(50)		
strNasPort	String		
strNASIPAddress	String(15)		
strPhoneNumber	String(10)	The current calling station for the adsl service	
strStartTime	String(50)	The session start time	
strStopTime	String(50)	The session stop time	
strDuration	String		
strTerminateCause	String(50)	The reason the service disconnected	
strFramedIP	String(15)		
strAcctUniqueId	String		
intUploadMB	String		
intDownloadMB	String		
intTotalMB	String		

3.7 getPreviousMonthUsageById

This function is used to create a client and will return a client identifier that will be used for creation of services.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
guidServiceId	String(36)	(*Required) id of the client that you want to retrieve the services for	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org /Protocols /rfc2616 /rfc2616-sec10.html
arrSessions			
strUsername	String(50)	The client code	
strNasPort	String(50)		
strNASIPAddress	String(15)		
strPhoneNumber	String(10)	The current calling station for the adsl service	
strStartTime	String(50)	The session start time	
strStopTime	String(50)	The session stop time	
strDuration	String		
strTerminateCause	String(50)	The reason the service disconnected	
strFramedIP	String(15)		
acctUniqueId	String		
intUploadMB	String		
intDownloadMB	String		
intTotalMB	String		

3.8 getRadiusServiceBandwidth (Method=GET)

This function takes in a guidServiceId and returns the current available bandwidth for that service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	GUID to identify the service	
Out Parameters			
intPurchased	Integer	Bandwidth Purchased (incl topups/servicechanges)	536870912
intRollover	Integer	If the service has rollover. Else it will be 0	0
intAvailable	Integer	Bandwidth Available. Excluding rollover.	504583630

- All bandwidth amounts are in bytes.
- Uncapped accounts will display 'N/A'

3.9 getServiceUsageDetailsById (Method=GET)

This function is used to retrieve usage of a particular service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the client that you want to retrieve the services for	
Out Parameters			
strAccountStartDate			
strDate			
intTotal			
intTotalUp			
intTotalDown			
arrUsage	Array	Array breakdown per day listed below	
intTotal			
intUp			
intDown			

3.10 getServiceSipDetailsById (Method=GET)

These are the details of your Voice Service (number,password)

Name	Type	Description	Values
In Parameters			
strSessionId	String	(*Required) Session Identifier	
guidServiceId	String(36)	Service identifier	
Out Parameters			
arrDetails	Array	Array containing the username and password of the Voice Service	
voip_username	String	SIP username	
sip_password	String	SIP password	

3.11 getVoipUsage (Method=GET)

This function allows you to get the usage data for a Voip (voice service). The dates have to be with in the format specified below. You will also need to supply the guidServiceId of the Voip service. This function is limited to only returning a maximun of 3 months worth of data at a time. Any date ranges exceeding the 3 months will result in the function returning data for the current month only.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identifier	
guidServiceId	String(36)	(*Required) Service identifier of Voip Service	
strStartDate	String(36)	(*Required) Start Date Range	yyyy-mm-dd
strEndDate	String(36)	(*Required) End Date Range	yyyy-mm-dd
Out Parameters			
arrCalltimeUsage	String(36)	Array breakdown listed below	
strDialed	String(36)	Number dialed for call	
strDate	String(36)	Date and time of call	
strStart	String(36)	Start date and time of call	
strEnd	String(36)	End date and time of call	
strDuration	String(36)	Duration of the call	
decTalkTime	Decimal(18,3)	Amount of Talk time used for call	
decCost	Decimal(18,3)	Cost of the call	
decBalance	Decimal(18,3)	Balance remaining after the call	

3.12 getVoipBalances (Method=GET)

This function returns expects the guidServiceId of a Voip service and returns an array consisting of the necessary details related to usage, allocation and rollover. To calculate the calltime for the current month you will need to subtract the sum of all rollover available (strAvailable1 + strAvailable2 + strAvailable3) from the total allocation (strAvailable).

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identifier	
guidServiceId	String(36)	(*Required) Service identifier	
Out Parameters			
arrVoipBalances	Array	Array break down listed below	
decAvailable	Decimal(18,3)	Total available calltime	
decCreditPurchased	Decimal(18,3)	Total allocation (Current month+Rollover+Topups)	
decUsed	Decimal(18,3)	Total usage for current month	
strMonth1	String(36)	Rollover month 1	
strMonth2	String(36)	Rollover month 2	
strMonth3	String(36)	Rollover month 3	
decStart1	Decimal(18,3)	Rollover month 1 allocated calltime	
decStart2	Decimal(18,3)	Rollover month 2 allocated calltime	
decStart3	Decimal(18,3)	Rollover month 3 allocated calltime	
decAvailable1	Decimal(18,3)	Rollover month 1 available calltime	
decAvailable2	Decimal(18,3)	Rollover month 2 available calltime	
decAvailable3	Decimal(18,3)	Rollover month 3 available calltime	

4. Create Service

4.1 createService (Method=PUT)

This function allows you to create a new service for a client.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidClientId	String(36)	(*Required) Id of the client that you want to retrieve the services for	
guidProductId	String(36)	(*Required) The productid that the service is linked to	
Out Parameters			
decBalance	Decimal(18,3)	Balance that you have left to purchase services	
arrServices	Array	Array break down listed below	

4.2 ADSL Service (Method=PUT)

These details are required to create an ADSL service.

Name	Type	Description	Values
In Parameters			
intQuantity	Integer	(*Required) The per gig amount for the ADSL service	5 for a 5 gig service
strUsername	String(20)	(*Required) Username for the router	test@axxess
strDateStart	String	(*Feature not yet available) Date for when the service will begin and the invoice created	
Out Parameters			
guidServiceId	String(36)	The identifier for the service	These will only be returned on successful service creation.
strUsername	String (50)		
strPassword	String (50)	Password for your router	

4.3 Mobile Service (Method=PUT)

These details are required to create a mobile service.

Name	Type	Description	Values
In Parameters			
guidMobileServiceId	String(36)	(*Required) The GUID of the sim card for data allocation	
strDateStart	String	(*Feature not yet available) Date for when the service will begin and the invoice created (Feature unavailable presently)	
Out Parameters			
guidServiceId	String(36)	The identifier for the service	This will only be returned on successful service creation.

4.4 DSL Line Service (Method=PUT)

These details are required to create a DSL line service.

Name	Type	Description	Values
In Parameters			
strLineNumber	String(50)	(*Required) The line number of the DSL line that needs to be created	
strOwnerName	String(50)	(*Required) The owner name of the line being created	
strAddress	String(250)	(*Required) Full address of the place where the line is	
strSuburb	String(100)	(*Required) The suburb of the place where the line is	
strCity	String(100)	(*Required) The town or city of the place the line is	
intPostalCode	Integer	(*Required) The postal code of the place the line is	
intIsTransfer	Integer	(*Required) Is it a transfer 0|1	
Out Parameters			
guidServiceId	String(36)	The identifier for the service	This will only be returned on successful service creation.

4.5 getServiceChangeHistory (Method=GET)

This function takes in a guidServiceId and a date (YYYY-MM) and returns all the servicechange/topup history for the given month.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	GUID to identify the service	
strDate	String(36)	Year, Month of history requested	Eg: ‘2016-04’
Out Parameters			
arrDetails	String(36)	This is a generated GUID that allows the reseller to interact with the api	369a1e34-4d6d-11e4-a967-002590383a96
OriginalProductGuidId	String(36)	Guid to identify start product	
OriginalQuantity	String(36)	Quantity of start service	
ChangeProductGuidId	String(36)	Guid to identify the change service	
ChangeQuantity	String(36)	Quantity of change service	
DateCreated	String(36)		
DateStart	String(36)		
Type	String(36)	To identify what type of change it was	“Topup” “ServiceChange”

4.6 VoIP Service (New)

These are the details that are required for creation of a NEW Voip Service

Name	Type	Description	Values
In Parameters			
guidProductId	String(36)	Product identifier	
strVoiceNumberType	String(36)	Identifier for either a new or ported voip number	‘new’
strOwnerName	String(36)	The owner name of the Voip Service	
strAddress	String(36)	The owner’s address	
strSuburb	String(36)	The owner’s suburb	
strCity	String(36)	The owner’s City	
intPostalCode	Int(4)	The owner’s postal code	
strEmail	String(36)	The owner’s email address	
strCell	String(36)	The owner’s cellphone number	
Out Parameters			
guidServiceId	String(36)	The identifier for the service	These will only be returned on successful service creation.

VoIP Service (Existing)

*When porting a voip number, certain documents are needed.

Before Calling the createService api call, documents first need to be uploaded.

The Documents needed are:

- Copy Of Your ID
- Latest paid invoice of current Service Provider
- Proof of Payment of existing network operators account
- Signed REQUEST FORM

Getting the Request Form.

The Request Form can be obtained by logging into your control panel (https://rcp.axxess.co.za) and going to My Account -> Documents . They will be under the Voice Porting Documents tab.

Once the form is filled in, re-upload it and the other documents using the following:

4.7 createVoiceFileAttachments (Method=POST)

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Product identifier	
strNumber	String(36)	The phone number you are trying to port	
fileIdCopy	file	Copy Of Your ID	
filePayment	file	Proof of Payment of existing network operators account	
fileStatement	file	Latest paid invoice of current Service Provider	
fileSignedRequest	file	Signed REQUEST FORM	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

4.8 VoIP Service (Existing)

These are the details that are required for creation of a **NEW** Voip Service

Name	Type	Description	Values
In Parameters			
guidProductId	String(36)	Product identifier	
strVoiceNumberType	String(36)	Identifier for either a new or ported voip number	'existing'
strPortedNumber	String(36)	The number you want to port. (this must match the number submitted in the createVoiceFileAttachments Call)	
strOwnerName	String(36)	The owner name of the Voip Service	
strAddress	String(36)	The owner's address	
strSuburb	String(36)	The owner's suburb	
strCity	String(36)	The owner's City	
intPostalCode	Int(4)	The owner's postal code	
strEmail	String(36)	The owner's email address	
strCell	String(36)	The owner's cellphone number	
Out Parameters			
guidServiceId	String(36)	The identifier for the service	These will only be returned on successful service creation

4.9 Fixed LTE Service

4.9.1 Telkom Fixed LTE

4.9.1.1 checkTelkomLteAvailability (Method=POST)

An address will need to be searched using the the Telkom LTE map render (8.2). Once the check is run there will be some details popolated in the html within hidden fields. These details in the hidden fields is what is needed for running this function.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
strLongitude	String	Address longitude retrieved from map render	
strLatitude	String	Address latitude retrieved from map render	
strAddress	String	Street address retrieved from map render	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.1.2 getTelkomLteHardwareProductOptions (Method=GET)

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
Out Parameters			
arrTelkomLteHardwareProductOptions	Array	Array breakdown listed below	
guidProductId	String(36)	Product Identifier	
strName	String(36)	Product Name	

4.9.1.3 getTelkomLteProductsForPurchase (Method=GET)

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidProductId	String(36)	Hardware Product Identifier	
Out Parameters			
arrTelkomLteSmartComboProducts	Array	Array breakdown listed below	
guidProductId	String(36)	Product Identifier	
strName	String(36)	Product Name	
strHardware	String(36)	Hardware Option	

4.9.1.4 getTelkomLteAvailableSims (Method=GET)

This is used to retrieve a list of available/unassigned Telkom LTE sims on your account. Having an available sim is only needed when a purchase is made using the sim-only hardware option.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLteAvailableSims	Array	Array breakdown below:	
guidServiceId	String	Sim service identifier	
strSerialNumber	String	Sim serial number	

4.9.1.5 purchaseTelkomLteService (Method=POST)

This function is used when you want process your purchase. The combo product identifier is retrieved using the getTelkomLteProductsForPurchase function. The sim service identifier, retrieved from the getTelkomLteAvailableSims function, is only required when making a sim-only purchase which requires that the sim-only boolean indicator to be set to 1. The reseller client identifier indicates the client to whom you would want to assign the new Telkom LTE service to.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
strClientId	String	Reseller client identifier	
guidProductId	String	Combo product identifier	
guidServiceId	String	Sim service identifier	
intSimOnly	Int	Sim only boolean indicator	1 or 0 (One or Zero)
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
guidServiceId	String	Created combo service identifier	

4.9.1.6 getTelkomLteTopupProducts (Method=GET)

In order to topup your Telkom LTE service you need to retrieve the topup options. This function retrieves those topup options.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrTelkomLteTopupProducts	Array	Array breakdown below:	
guidProductId	String	Topup product identifier	
strName	String	Topup product description	

4.9.1.7 purchaseTelkomLteTopup (Method=POST)

Once you retrieve the topup options from the getTelkomLteTopupProducts function you need to pass one of those options (Topup product identifier) through along with the combo service identifier, which is the service you would like to topup.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
guidProductId	String	Topup product indentifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.1.8 getTelkomLteServiceChangeProducts (Method=GET)

In order to change the package size of your Telkom LTE service you need to retrieve the service change options. This function retrieves those service change options.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo Service identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrTelkomLteServiceChangeProducts	Array	Array breakdown below:	
guidProductId	String	Service change product identifier	
strName	String	Service change product description	

4.9.1.9 purchaseTelkomLteServiceChange (Method=POST)

Once you retrieve the service change options from the getTelkomLteServiceChangeProducts function you need to pass one of those options (Service change product identifier) through along with the combo service identifier, which is the service you would like to change the package size of.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
guidProductId	String	Service change product identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.1.10 getTelkomLteBandwidth (Method=GET)

Use this function to retrieve the usage/bandwidth for the specified combo service

In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrTelkomLteBandwidth	Array	Array breakdown below:	
updated	String	The time that the usage data was last updated	
peak	Array	Array	
total	Float	Peak total bandwidth	
remaining	Float	Peak remaining bandwidth	
used	Float	Peak used bandwidth	
offpeak	Array	Array	
total	Float	Offpeak total bandwidth	
remaining	Float	Offpeak remaining bandwidth	
used	Float	Offpeak used bandwidth	
topup	Array	Array	
total	Float	Topup total bandwidth	
remaining	Float	Topup remaining bandwidth	
used	Float	Topup used bandwidth	
topup_offpeak	Array	Array	
total	Float	Offpeak topup total bandwidth	
remaining	Float	Offpeak topup remaining bandwidth	
used	Float	Offpeak topup used bandwidth	
rollover	Array	Array	
total	Float	Rollover total bandwidth	
remaining	Float	Rollover remaining bandwidth	
used	Float	Rollover used bandwidth	
summation	Array	Array	
total	Float	Summation total bandwidth	
remaining	Float	Summation remaining bandwidth	
used	Float	Summation used bandwidth	
current_month	Array	Array	
total	Float	Current month total bandwidth	
remaining	Float	Current month remaining bandwidth	
used	Float	Current month used bandwidth	

4.9.2 MTN Fixed LTE

4.9.2.1 checkMtnFixedLteAvailability (Method=POST)

An address will need to be searched using the the MTN Fixed LTE map render (8.3). Once the check is run there will be some details popolated in the html within hidden fields. These details in the hidden fields is what is needed for running this function.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
strLongitude	String	Address longitude retrieved from map render	
strLatitude	String	Address latitude retrieved from map render	
strAddress	String	Street address retrieved from map render	
strBBox	String	Retrieved from map render	
strWidth	String	Retrieved from map render	
strHeight	String	Retrieved from map render	
strICoOrdinate	String	Retrieved from map render	
strJCoOrdinate	String	Retrieved from map render	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.2.2 getMtnFixedLtePurchaseOptions (Method=GET)

Use this function to retrieve the available hardware options for purchasing MTN Fixed LTE. The hardware options returned should normally consist of Sim-Only and/or a Router.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLtePurchaseOptions	Array	Array break below:	
guidProductId	String	Hardware product identifier	
strName	String	Hardware product description	

4.9.2.3 getMtnFixedLtePurchaseProducts (Method=GET)

A hardware product from the previous function (getMtnFixedLtePurchaseOptions) is required for running this function. The hardware product is passed into this function and the result should be a list of MTN Fixed LTE combo products representing the package plus the hardware.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidProductId	String	Hardware product identifier	
Out Parameters			
IntCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLtePurchaseProducts	Array	Array breakdown below:	
guidProductId	String	Combo product identifier	
strName	String	Combo product description	
strHardware	String	Combo hardware product description	

4.9.2.4 getMtnFixedLteAvailableSims (Method=GET)

This is used to retrieve a list of available/unassigned MTN Fixed LTE sims on your account. Having an available sim is only needed when a purchase is made using the sim-only hardware option.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
Out Parameters			
IntCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLteAvailableSims	Array	Array breakdown below:	
guidServiceId	String	Sim service identifier	
strSerialNumber	String	Sim serial number	

4.9.2.5 purchaseMtnFixedLteService (Method=POST)

This function is used when you want process your purchase. The combo product identifier is retrieved using the getMtnFixedLtePurchaseProducts function. The sim service identifier, retrieved from the getMtnFixedLteAvailableSims function, is only required when making a sim-only purchase which requires that the sim-only boolean indicator to be set to 1. The reseller client identifier indicates the client to whom you would want to assign the new MTN Fixed LTE service to.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidClientId	String	Reseller client identifier	
guidProductId	String	Combo product identifier	
guidServiceId	String	Sim service identifier	
intSimOnly	Int	Sim only boolean indicator	1 or 0 (One or Zero)
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
guidServiceId	String	Created combo service identifier	

4.9.2.6 getMtnFixedLteTopupProducts (Method=GET)

In order to topup your MTN Fixed LTE service you need to retrieve the topup options. This function retrieves those topup options.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLteTopupProducts	Array	Array breakdown below:	
guidProductId	String	Topup product identifier	
strName	String	Topup product description	

4.9.2.7 purchaseMtnFixedLteTopup (Method=POST)

Once you retrieve the topup options from the getMtnFixedLteTopupProducts function you need to pass one of those options (Topup product identifier) through along with the combo service identifier, which is the service you would like to topup.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
guidProductId	String	Topup product indentifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.2.8 getMtnFixedLteServiceChangeProducts (Method=GET)

In order to change the package size of your MTN Fixed LTE service you need to retrieve the service change options. This function retrieves those service change options.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo Service identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLteServiceChangeProducts	Array	Array breakdown below:	
guidProductId	String	Service change product identifier	
strName	String	Service change product description	

4.9.2.9 purchaseMtnFixedLteServiceChange (Method=POST)

Once you retrieve the service change options from the *getMtnFixedLteServiceChangeProducts* function you need to pass one of those options (Service change product identifier) through along with the combo service identifier, which is the service you would like to change the package size of.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
guidProductId	String	Service change product identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.2.10 getMtnFixedLteBandwidth (Method=GET)

Use this function to retrieve the usage/bandwidth for the specified combo service

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Combo service identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrMtnFixedLteBandwidth	Array	Array breakdown below:	
peak	Array	Array	
total	Float	Peak total bandwidth	
remaining	Float	Peak remaining bandwidth	
used	Float	Peak used bandwidth	
offpeak	Array	Array	
total	Float	Offpeak total bandwidth	
remaining	Float	Offpeak remaining bandwidth	
used	Float	Offpeak used bandwidth	
topup	Array	Array	
total	Float	Topup total bandwidth	
remaining	Float	Topup remaining bandwidth	
used	Float	Topup used bandwidth	
rollover	Array	Array	
total	Float	Rollover total bandwidth	
remaining	Float	Rollover remaining bandwidth	
used	Float	Rollover used bandwidth	
summation	Array	Array	
total	Float	Summation total bandwidth	
remaining	Float	Summation remaining bandwidth	
used	Float	Summation used bandwidth	
current_month	Array	Array	
total	Float	Current month total bandwidth	
remaining	Float	Current month remaining bandwidth	
used	Float	Current month used bandwidth	

4.9.3 checkFixedLteAvailabilty (Method=POST)

Use this to check for the availability of Telkom LTE and/or MTN LTE based on the parameters passed in.
The parameters can be retrieved using 8.4 LTE Map Render.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
strLongitude	String	Longitude	
strLatitude	String	Latitude	
strAddress	String	Address	
strBBox	String	B-Box values	
strWidth	Interger	Width	
strHeight	Interger	Height	
strICoOrdinate	Interger	I-Coordinate	
strJCoOrdinate	Interger	J-Coordinate	
Out Parameters			
intCode	Interger	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

4.9.4 checkFixedWirelessAvailability (Method=GET)

This is used to check if there is coverage at a specific address for either Rain or Cell C. The strLongitude, strLatitude and strAddress can be aquired with the Fixed Wireless Map Render found at 8.2 and 8.3

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identifier	
guidNetworkProviderId	String(36)	(*Required) Network Provider identifier	
strLongitude	String(36)	(*Required) Longitude of address	
strLatitude	String(36)	(*Required) Latitude of address	
strAddress	String(36)	(*Required) Address	
Out Parameters			
intReturnCode	String(36)	JSON string with error breakdown	
strMessage	String(36)	User message indicating availability	

4.9.5 cancelFixedWirelessService (Method=PUT)

Cancel Fixed Wireless service or cancel Fixed Wireless order. These are the details that are required for canceling a fixed wireless service or canceling a Fixed Wireless order.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidNetworkProviderId	String(36)	Network Provider identifier	
guidServiceId	String(36)	Service identifier	
strDate	String(36)	The date that the service needs to end	yyyy-mm-dd
Out Parameters			
intReturnCode	String	JSON string with error breakdown	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

4.9.6 getFixedWirelessTopupProducts (Method=GET)

Get the Top-up products that you can use to top-up your Fixed Wireless service with. These are the details that are required for getting fixed wireless top-up products.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidNetworkProviderId	String(36)	Network Provider identifier	
Out Parameters			
arrProducts	Array	Array breakdown listed below	
guidProductId	String(36)	Product identifier	
strName	String(36)	Name of top-up product	

4.9.7 funcFixedWirelessTopup (Method=PUT)

Top-up your Fixed Wireless data service. These are the details that are required for topping up a Fixed Wireless service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidServiceId	String(36)	Service identifier	
guidProductId	String(36)	Product identifier of top-up product	
Out Parameters			
intReturnCode	String	JSON string with error breakdown	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

4.9.8 getFixedWirelessDataUsage (Method=GET)

Get usage for the last 7 days (including today), current month and the last 2 months. These are the details that are required for retrieving the data usage.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidNetworkProviderId	String(36)	Network Provider identifier	
guidServiceId	String(36)	Session identifier	
Out Parameters			
arrUsage	Array	Array breakdown listed below	
decMonth1	Decimal(18,3)	Current month usage (GB)	55.32
decMonth2	Decimal(18,3)	Previous month usage (GB)	55.32
decMonth3	Decimal(18,3)	Month before last usage (GB)	55.32
decLast7Days	Decimal(18,3)	Usage for the last 7 days (GB)	55.32

4.9.9 getFixedWirelessServiceChangeProducts› (Method=GET)

Get the Service Change products that you can use to Service Change your Fixed Wireless service with. These are the details that are required for getting fixed wireless Service Change products.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidNetworkProviderId	String(36)	Network Provider identifier	
Out Parameters			
arrProducts	Array	Array breakdown listed below	
guidProductId	String(36)	Product identifier	
strName	String(36)	Name of Service Change product	

4.9.10 funcServiceChanges (Method=PUT)

Service change your existing Fixed Wireless smart combo to a different smart combo package. These are the details that are required for actioning the service change.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identifier	
guidServiceId	String(36)	Service identifier	
guidProductId	String(36)	Product identifier. The product you want to change to	
intQuantity	Int(1)		1
strDateStart	String(36)	The date that the change needs to take place	yyyy-mm-dd
Out Parameters			
intReturnCode	String	JSON string with error breakdown	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

4.10 Just DSL

4.10.1 checkJustDslAvailability (Method=POST)

This function requires a latitude and longitude in order to be used to check for Just DSL availability. This function will indicate whether a location/address is able to sign up for Just DSL.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
strLatitude	String	Address latitude	
strLongitude	String	Address longitude	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
intMaxSpeed	Int	Maximum line speed	

4.10.2 getJustDslComboProducts (Method=GET)

Use this function to retrieve available Just DSL products for purchase. The guidProductId data returned is needed when running the purchaseJustDslCombo() function.

Name	Type	Description	Values
In Parameters			
strSessionId	String	(*Required) Session identifier	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrJustDslProducts	Array	Array breakdown below:	
guidProductId	String	Product identifier	
strName	String	Product name	

4.10.3 purchaseJustDslCombo (Method=POST)

Use this function to purchase/create a Just DSL combo service. Use the getAddressTypes() function to retrieve the valid address types accepted by this function. If the address type is either Complex or Flat, then strUnitNumber and strBlockName becomes required.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidClientId	String	Client identifier	
guidProductId	String	Product identifier	
strOwnerName	String	The client name that will have this service installed	
strCell	String	The client cell number that will have this service installed	
strAddressType	String	The address type where the service will be installed	getAddressTypes
strAddress	String	The address where the service will be installed	
strSuburb	String	The suburb where the service will be installed	
strCity	String	The city where the service will be installed	
strCode	String	The postal code where the service will be installed	
strLatLon	String	The co-ordinates where the service will be installed (corresponds to the address details)	
strUnitNumber	String	(*Conditionally required) Unit number within the complex or flat	
strBlockName	String	(*Conditionally required) Name of the complex or flat	
strUsername	String	(*Optional) Set a custom username for the data service	
Out Parameters			
intCode	Int	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
decBalance	Float	Remaining balance after purchase	
arrServices	Array	Array breakdown below:	
guidServiceId	String	Service identifier of created services	
strUsername	String	Username for data service	
strPassword	String	Password for data service	

5. Tickets

5.1 getDepartments (Method=GET)

This function is used for retrieving all the departments that will be used when creating a ticket.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
Out Parameters			
arrDepartments	Array		
guidDepartmentId	String(36)	The department id that will be used to create tickets	
Name	String	This is the display name	Shipping

5.2 createTicket (Method=GET)

This function is used for retrieving all the departments that will be used when creating a ticket.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidDepartmentId	String(36)	Department related to ticket query	
strSubject	String(255)	Your ticket subject	
strMessage	String(255)	The body message of your ticket	
Out Parameters			
strMessage	String(36)	The id of the ticket that was just created	

5.3 updateTicket (Method=GET)

This function is used for retrieving all the departments that will be used when creating a ticket.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
strTicketId	String(36)	The TicketId Related to the ticket you updating.	
strMessage	String(255)	The body message of your ticket update	
Out Parameters			
intReturnCode	String	String JSON with error breakdown	http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

5.4 viewTicket (Method=GET)

This function is used for retrieving all the departments that will be used when creating a ticket.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
strTicketId	String(36)	The TicketId Related to the ticket you viewing.	
Out Parameters			
arrDetails	Array	Array of details pertaining to the ticket.	http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
Subject	String(36)	Ticket Subject	
DateCreated	String(36)	Date Created of the ticket	
guidDepartmentId	String(36)	Guid to identify the department	
Details(array)	Array	Array of ticket messages	
Message	String(255)	Ticket message	
Operator	String(36)	Author of ticket message	
DateCreated	String(36)	Date created of the ticket messages	

6. Functions

6.1 funcServiceChanges (Method=PUT)

This function allows you to change existing services.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) The service id for the service that the change is for	
guidProductId	String(36)	(*Required) The product id that the service is changing to	
intQuantity	Integer	This is optional field that will be used only if the product is a unit product	Capped 3 gig to a 6 gig the quantity will be 6. Uncapped services and lines will not need this
strDateStart	String(50)	(*Required) The start date of the service change. If you set this to any date later then the current day it will start on the first of the next month	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.2 funcTopups (Method=PUT)

This allow for creation topups.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	Integer	(*Required) id of the client that you want to retrieve the services for	
guidProductId	String(36)	(*Required) The product id that the service is changing to	
intQuantity	Integer	(*Required) This function only applies to unit services so this field must be populated	Capped 3 gig to a 6 gig the quantity will be 3.
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.2.1 Mobile

- guidProductId
    - Topup via a guidProductId:

guidProductId:	Name
009bfff3-cc80-11e4-b901-000c2994d8d2	~~Mobile 500MB~~
009c0062-cc80-11e4-b901-000c2994d8d2	Mobile 1GB
009c00d1-cc80-11e4-b901-000c2994d8d2	Mobile 2GB
288f5951-16b0-11e6-83e7-000c2994d8d2	Mobile 3GB
009c0142-cc80-11e4-b901-000c2994d8d2	Mobile 4GB
288f59df-16b0-11e6-83e7-000c2994d8d2	Mobile 5GB
009c01b1-cc80-11e4-b901-000c2994d8d2	Mobile 6GB
009c021f-cc80-11e4-b901-000c2994d8d2	Mobile 8GB
009c028e-cc80-11e4-b901-000c2994d8d2	Mobile 10GB
009c02fe-cc80-11e4-b901-000c2994d8d2	Mobile 12GB
009c03db-cc80-11e4-b901-000c2994d8d2	Mobile 16GB
009c0447-cc80-11e4-b901-000c2994d8d2	Mobile 20GB
009c04b5-cc80-11e4-b901-000c2994d8d2	Mobile 24GB

- intQuantity = 1;

6.2.2 ADSL

- guidProductId
    - guidProductId of the current service.

- intQuantity
    - The Amount of gigs you want to topup by.

6.3 funcPod (Method=GET)

This function sends a packet of disconnect on an ADSL service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to POD	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.4 funcPasswordReset (Method=GET)

This function will update your radius password for the specified service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to POD	
strPassword	String(10)	(* Optional) Password you want to use.	
Out Parameters			
strPassword	String(50)	New reset password	

6.5 funcSuspend (Method=GET)

This function suspends a service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to suspend	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.6 funcLiftSuspend

This function will lift the suspended status whatever service identifier is passed in. (only for services that have used the ‘funcSuspend’)

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to suspend	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.7 funcSendTotalBandwidth (Method=POST)

This function sends total usage via email.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
intDayStart	Integer	Starting day of the month. If no start day has been set then it will default to the 1st	
intDayEnd	Integer	End day of the month. If no end day has been set then it will default to the current day	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.8 funcExpireService

This function will set a service to either expire now, or for the end of the month.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to suspend	
strDate	String(36)	(*Required) The date you want the expiry to run on. (only valid for todays date or the last day of the months date)	Format(Y-m-d) eg. ‘2015-09-30’
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

6.9 funcFixLine (Method=GET)

This function runs a Uniweb port recreate on a DSL line service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to run	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

Functions (continues)...

6.10 funcCheckLine (Method=GET)

This function runs an interrogation on a DSL line service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidServiceId	String(36)	(*Required) id of the service that you want to check the line for	
Out Parameters			
strDownloadCurrentSpeed	String(50)	Shows the current speed in kbps	8189kbps
strDownloadInterference	String(50)	Shows current interference in dB	10dB
strDownloadNoiseRatio	String(50)	Shows current noise ratio in dB	12.2dB
strUploadCurrentSpeed	String(50)	Shows the current speed in kbps	8189kbps
strUploadInterference	String(50)	Shows current interference in dB	10dB
strUploadNoiseRatio	String(50)	Shows current noise ratio in dB	12.2dB

Example code

Basic auth is required as part of every request. The username and password will be made available on request along with a bootstrap file written in PHP which has done the client side requests through a Curl object implementation. The code is documented with both required inputs as well as expected output for example:

```
/**
*
*
* @param array $d array("strSessionId", "guidServiceId")
*
* @return object json encoded with:
* public 'intCode' => int 200
public 'strStatus' => string 'OK' (length=2)
public 'strMessage' => null
public 'strDownloadCurrentSpeed' => string '10239 kbps'
public 'strDownloadInterference' => string '12.2 dB'
public 'strDownloadNoiseRatio' => string '12.7 dB'
public 'strUploadCurrentSpeed' => string '1021 kbps'
public 'strUploadInterference' => string '16.9 dB'
public 'strUploadNoiseRatio' => string '12.3 dB'
*
*/
case "funcCheckLine":
{
$Url = "https://apitest.axxess.co.za/" . "calls/rsapi/funcCheckLine.json";
$curl->setBasicAuthentication($Username, $Password);
$curl->setOpt(CURLOPT_SSL_VERIFYPEER, false);
$curl->setOpt(CURLOPT_SSL_VERIFYHOST,2);
$curl->get($Url, $d);
break;
}
```

6.11 transferData (Method=POST)

Use this function to transfer data between services based on certain criteria. This function caters for Capped DSL, Capped Fibre, MTN 3G and MTN Fixed LTE transfers. Transfers can only be processed between services of the same type and between services on the same reseller account. This function expects the guidServiceId of the Data Service to be passed in and the strUsername of the Data Service you wanting to transfer the data to.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session Identifier	
guidServiceId	String	Service Identifier	
strUsenameTo	String	Username to transfer to	
intQuantityToTransfer	Integer	Amount of GB to transfer (increments of 1 GB only and nothing less than 1 GB)	Value >= 1
boolTransferAll	Integer	To transer all data set to 1 (when set to 1 it overwrites intQuantityToTransfer)	1 or 0
Out Parameters			
intCode	Integer	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

6.12 getRainTransferableData (Method=GET)

This funtion allows you to retrieve transferable data for specified Rain LTE services. You can either transfer anytime data or topup data depending on the availability. This function expects the guidServiceId of the Combo Service to be passed in and NOT the Data Service. The results of this function is used in the transferRainData funtion.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session Identifier	
guidServiceId	String	Service Identifier	
Out Parameters			
intCode	Integer	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrRainTransferableData	Array	Array breakdown below	
strType	String	Data type	‘topup’ or ‘anytime’
decAvailable	Float	Available transferable data	1.653 GB

6.13 transferRainData (Method=POST)

Use this function to transfer data between services based on certain criteria. This function caters for Rain LTE transfers. Transfers can only be processed between Rain LTE services and between Rain LTE services on the same reseller account. This function expects the guidServiceId of the Combo Service to be passed in and NOT the Data Service. It aslo expects the strUsername of the Data Service you wanting to transfer the data to. Use the getRainTransferableData function to retrieve information required to be passed into this funtion.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session identifier	
guidServiceId	String	Service identifier	
strUsername	String	Username to transfer to	
strFromType	String	Transfer from type	'topup' or 'anytime'
strToType	String	Transfer to type	'topup' or 'anytime'
intQuantityToTransfer	Integer	Amount of GB to transfer (increments of 1 GB only and nothing less than 1 GB)	
Out Parameters			
intCode	Integer	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	

6.14 viewDataTransferHistory (Method=GET)

Use this function to view the history of transfers. This function is used for Capped DSL, Capped Fibre, MTN 3G, MTN Fixed LTE and Rain LTE. The function expects the guidServiceId of the data service to passed in except when using it for Rain LTE, where you need to pass in the Rain LTE combo service.

Name	Type	Description	Values
In Parameters			
strSessionId	String	Session Identifier	
guidServiceId	String	Service Identifier	
Out Parameters			
intCode	Integer	HTTP status code	
strStatus	String	HTTP status message	
strMessage	String	Response message	
arrResults	Array	Array breakdown below	
arrTransfersIn	Array	Array containing details of transfers coming in	
arrTransfersOut	Array	Array containing details of transfers go out	

7. Fibre

7.1 getAddressTypes

This function will return a list of all the address types used to place a fibre order.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
Out Parameters			
arrAddressTypes	Array		1

7.2 getNetworkProviders

This function will return a list of all the network providers.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
Out Parameters			
arrNetworkProviders	Array	Array break down listed below	1
guidNetworkProviderId	String(36)	Network provider identifier	
strName	String(36)	Network provider identifier	

7.3 getNetworkProviderProducts

This function will return the product identifiers for a specific network provider.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	(*Required) Session identified	
guidNetworkProviderId	String(36)	Network provider identifier	
Out Parameters			
arrNetworkProviderProducts	Array	Array break down listed below	1
guidProductId	String(36)	Product Identifier	
strName	String(36)	Product Name	

7.4 checkFibreAvailability

This checks to see whether given coordinates are capable of receiving fibre.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
strLongitude	String(36)	Longitude	25.5868448999999
strLatitude	String(36)	Latitude	-33.9549916
strAddress	Text	Physical Address	185 Cape Road, Port Elizabeth
Out Parameters			
arrAvailableProvidersGuids	Array	Array break down listed below	1
guidNetworkProviderId	String(36)	Network provider identifier	
intPreOrder	Integer	If the location is pre-order	

7.5 createFibreComboService

Create a fibre service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
guidClientId	String(36)	Id of the client you want to retrieve the services for	
guidProductId	String(36)	The product id that the service is linked to	
guidNetworkProviderId	String(36)	Network provider identifier	
strOwner	String(36)	Fibre Line Owner Name	
strCell	String(36)	Fibre Line Owner Cell	
strAddress	String(36)	Fibre Line Owner Address	
strSuburb	String(36)	Fibre Line Owner Suburb	
strCity	String(36)	Fibre Line Owner City	
strCode	String(36)	Fibre Line Owner Postal Code	
strLatLong	String(36)	Fibre Line Installation coordinates	
strAddressType	String(36)	Retreived from the ’getAddressType’ function	
strBuildingId	String(36)	Fibre Line Installation Building Number	
strFloorId	String(36)	Fibre Line Installation Floor	
strUnitNumber	String(36)	Fibre Line Installation Unit Number	
strBlockName	String(36)	Fibre Line Installation Block Name	
Out Parameters			
arrServices	Array	Array break down listed below	1
guidServiceId	String(36)	Identifies for the combo you just purchased	
decBalance	String(36)	Credit Balance.	

7.6 createFibreComboPreOrder

Create a pre-order fibre service.

Name	Type	Description	Values
In Parameters			
strSessionId	String(36)	Session identified	
guidClientId	String(36)	Id of the client you want to retrieve the services for	
guidProductId	String(36)	The product id that the service is linked to	
guidNetworkProviderId	String(36)	Network provider identifier	
strAddress	String(36)	Fibre Line Owner Address	
strSuburb	String(36)	Fibre Line Owner Suburb	
strCity	String(36)	Fibre Line Owner City	
strCode	String(36)	Fibre Line Owner Postal Code	
strLatLong	String(36)	Fibre Line Installation coordinates	
strAddressType	String(36)	Retreived from the ’getAddressType’ function	
strBuildingId	String(36)	Fibre Line Installation Building Number	
strFloorId	String(36)	Fibre Line Installation Floor	
strUnitNumber	String(36)	Fibre Line Installation Unit Number	
strBlockName	String(36)	Fibre Line Installation Block Name	
Out Parameters			
intReturnCode	String	JSON string with error break down	http code: http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

8. Map Render Requirements

To use our customised Google Maps, you must first register your app project on the Google Cloud Platform Console and get a Google API key.

Start here:

https://developers.google.com/maps/documentation/javascript/get-api-key

Please ensure that the following API's are enabled in the Google Cloud Platform Console:

- Maps JavaScript API
- Geocoding API
- Places API

To view your list of enabled APIs:

- Go to the Google Cloud Platform Console.
- Click the Select a project button, then select the same project you set up for the Maps JavaScript API and click Open.
- From the list of APIs on the Dashboard, look for the API.
- If you see the API in the list, you're all set. If the API is not listed, enable it:
    - At the top of the page, select ENABLE API to display the Library tab. Alternatively, from the left side menu, select Library.
    - Search for the API, then select it from the results list.
    - Select ENABLE. When the process finishes, the API will appear in the list of APIs on the Dashboard.

8.1 Fibre Map Render

```
<!DOCTYPE html>
<html lang=”en”>
<head></head>
<body>
<form method=”post”>

        <script type=”text/javascript” id=”fibrescript”>
                (function() {
                        var ax = document.createElement(‘script’);
                        ax.id = ‘mainscript’;
                        ax.type = ‘text/javascript’;
                        ax.async = true;
                        ax.src = ‘https://rcp.axxess.co.za/public/js/fibremapJs.php?key=google-api-key&width=width-in-
                                  percentage&height=height-in-pixels’;
                        var s = document.getElementsByTagName(‘script’)[0];
                        s.parentNode.insertBefore(ax, s);
                })();
        </script>

        <input type=”submit” value=”submit” />
</form>
</body>
</html>
```

Instructions:

On $_POST, you will get the following:

```
    ‘address-input’ => string  ‘ADDRESS_OF_MAP_MARKER’
    ‘latlong-input’ => string  ‘LAT_LONG_OF_MAP_MARKER’
```

8.2 Telkom LTE Map Render

```
<!DOCTYPE html>

<html lang=”en”>

<head></head>

<body>

        <script type=”text/javascript” id=”telkomscript”>

                (function() {

                        var ax = document.createElement(‘script’);

                        ax.id = ‘mainscript’;

                        ax.type = ‘text/javascript’;

                        ax.async = true;

                        ax.src = ‘https://rcp.axxess.co.za/public/js/telkomLteJs.php?key=google-api-key&width=width-in-
                                  percentage&height=height-in-pixels’;

                        var s = document.getElementsByTagName(‘script’)[0];

                        s.parentNode.insertBefore(ax, s);

                })();

        </script>

</body>

</html>
```

8.3 MTN Fixed LTE Map Render

Use this snippet of HTML code to render the the MTN Fixed LTE coverage map. The map shows coverage only. You will need to click on the map to search or you can type the address into the address bar to search. When the check is performed you will get the required data which you need to use in the checkMtnFixedLteCoverage function.

```
<!DOCTYPE html>
<html lang=”en”>
<head>
     <title>MTN Fixed LTE Coverage</title>
     <meta content=”utf-8” http-equiv=”encoding”>
     <meta name=”viewport” content=”width=device-width, initial-scale=1, maximum-scale=1, user-scalable=0”>
</head>

<body>
     <script type=”text/javascript” id=”mtn-fixed-lte-script”>

     (function() {
         var ax = document.createElement(‘script’);
         ax.id = ‘main-script’;
         ax.type = ‘text/javascript’;
         ax.async = true;
         ax.src = ‘https://rcp.axxess.co.za/public/js/mtnFixedLteCoverageJs.php?key=google-api-key&width=width-in-per
         centage&height=height-in-pixels’;
         var s = document.getElementsByTagName(‘script’)[0];
         s.parentNode.insertBefore(ax, s);
     })();

     </script>
</body>
</html>
```

8.4 Fixed LTE Map Render

```
<!DOCTYPE html>
<html lang=”en”>
<head></head>
<body>
     <script type=”text/javascript” id=”telkommtnscript”>
         (function() {
                var ax = document.createElement(‘script’);
                        ax.id = ‘mainscript’;
                        ax.type = ‘text/javascript’;
                        ax.async = true;
                        ax.src = ‘https://rcp.axxess.co.za/public/js/telkomMtnLteJs.php?key=
google-api-key&width=width-in-percentage&height=height-in-pixels’;
                        var s = document.getElementsByTagName(‘script’)[0];
                        s.parentNode.insertBefore(ax, s);
         })();

     </script>
</body>
</html>
```