Brekeke PAL WebSocket

Version 3

Developer's Guide

Brekeke Software, Inc.

Version

Brekeke PAL WebSocket Version 3 Developer's Guide

Copyright

Disclaimer

1. Purpose

This document describes Brekeke PAL WebSocket, which is an interface that retrieves call details and also obtains and updates settings values from Brekeke PBX. For the future development of an application that integrates with Brekeke PBX, it is recommended that you use Brekeke PAL WebSocket instead of Brekeke PAL (.NET) or WebService.

2. Software Requirements

Brekeke PAL WebSocket requires Brekeke PBX 3.2 or later with the PAL option enabled. Brekeke PBX needs to be installed under Apache Tomcat 7.0.42 or later

3. Brekeke PAL WebSocket Usage

Use a WebSocket client that conforms to RFC6455.

3.1. Configuration at Brekeke PBX

Defining the valid WebSocket client IP addresses is required to connect with Brekeke PBX.

1. Log in to Brekeke PBX Admintool with Admin privileges.
2. In the [Options] > [Settings] > [Valid WebSocket Client IP Pattern] field, enter a regular expression to define the client's IP address pattern.

For example, ^192\.168\..+$ will include any client whose IP address starts with "192.168."

3.2. URL

Access Brekeke PAL WebSocket API using the following URL:

ws://<Brekeke PBX Host>:< Brekeke PBX Port>/pbx/ws?<login information>

For details about URL format, refer to section "Connecting with Brekeke PAL WebSocket"

3.3. Protocol

JSON-RPC 2.0 is used as a remote procedure call protocol. For more information, please refer to http://www.jsonrpc.org/.

4. Connecting with Brekeke PAL WebSocket

Get connected with Brekeke PBX by logging in as system administrator “sa”, administrator or user extension and retrieve necessary information required in the following requests by defining parameters: status, voicemail, registered, park or line and gain the access to allow the use of methods. Once login succeeds with login_user other than “sa”, you will no longer need to set the tenant parameter in the request methods. If the tenant parameter is set, it will be ignored.

4.1. login (Brekeke PBX v3.3.x and later)

Description:

Send encrypted login information with encrypted login user password which are encrypted with MD5 hasher algorithm.

Parameters:

login_password – A string encrypted with MD5 hasher algorithm in the following format before encryption: `<login_user>:<nonce>:<login_password>`
where <login_user> is login user extension number, <login_password> is login user password encrypted with MD5 hasher algorithm, and <nonce> is the value of parameter “nonce” got from the connection response.

4.2. URL format

ws://<Brekeke-PBX-IP>:<Brekeke-PBX-Port>/pbx/ws?tenant=<tenant-name>&login_user=<user-name>&login_password=<password>&user=<user-extension1>&user=<user-extension2>&registered=<registered-value>&status=<status-value>&voicemail=<voicemail-value>&callrecording=<callrecording-value>&park=<park-value>&line=<line-value>

<Brekeke-PBX-IP>	IP address of Brekeke PBX server
<Brekeke-PBX-Port>	The same port number as the one used to access Brekeke PBX Admintool
<tenant-name>	Tenant name. Do not specify this value if logging in as “sa” or in the case of using the single tenant version.
<user-name>	Login user name

<password>	The password for the login_user extension; If login password is not set from URL, Brekeke PAL websocket login method can be used to send encrypted login information as explained above.
<user-extension[n]>	The user extension(s) whose information will be retrieved from Brekeke PBX. This setting can be omitted when logging in as system administrator “sa”. The values can be: *, one user extension, or a list of user extensions connected by “&”. Setting * as value means all users of the system or under the specified tenant. An admin privilege is required for login_user when user is different from login_user
<registered-value>	True or false. Setting for if to get user extension(s) register status information. This setting can be omitted when logging in as system administrator “sa”.
<status-value>	True or false. Setting for if to get user extension(s) call status information. This setting can be omitted when logging in as system administrator “sa”.
<voicemail-value>	True or false. Settign for if to get user extension(s) voicemail information. This setting can be omitted when logging in as system administrator “sa”.
<recording-value>	True or false. Settign for if to get user extension(s) call recording information. This setting can be omitted when logging in as system administrator “sa”.
<park-value>	Start * or an array of park ID. This setting can be omitted when logging in as system administrator “sa”.
<line-value>	Start * or an array of line ID. This setting can be omitted when logging in as system administrator “sa”.

4.3. Response

If login password is not specified in URL, “login_password_required” method with one parameter “nonce” will be returned.

If login succeeds, WebSocket notifications will be returned with request information.

If login fails, an error message will be returned.

5. Brekeke PAL WebSocket Notification

Once a Brekeke PAL WebSocket connection has been created, the notify methods listed below will be sent back to clients. These notify methods contain the information required by request methods.

5.1. notify_callrecording (Brekeke PBX v3.3.x and later)

Description:

Describes the user's call recording status.

Parameters:

user – user extension

status – on or off

5.2. notify_line

Description:

Shared call status.

Parameters:

line – shared line ID with index

status – on or off

line_talker_id – talker ID of the shared line

line_talker – talker of the shared line

assigned_talker_id – talker ID of the user in the shared call

5.3. notify_park

Description:

Parked call status.

Parameters:

park – park number

status – on or off

room_id – room ID of the call when park status is on, required by some request methods below as rid

talker_id – talker ID of the user when park status is on, required by some request methods below as tid

5.4. notify_registered

Description:

Describes the user’s registered status.

Parameters:

user – user extension

registered – true or false

5.5. notify_status

Description:

Information about call status.

Parameters:

user – user extension

status – call status code

other_number – the extension of the other user in the call

room_id – room ID of the call, required by some request methods below as rid

talker_id – talker ID of the user, required by some request methods below as tid

user_display_name – the display name of the user

other_user_display_name – the display name of the other user in the call

time – the time stamp of the call

logid – the ID number, which is the same as the number in the request method

rescode – response code, used only in the disconnected call response with status -1

disconnected_by – 1 or 0, used only in the disconnected call response with status -1. “1” means the call is disconnected by the callee and “0” means the call is disconnected by the caller.

q850code – q850 code, used only in the disconnected call response with status -1

Call status codes:

CALLING = 0

INCOMING = 1

CALL_SUCCESS = 2

ENDTALKING = 12

ANSWER_SUCCESS = 14

PARK_CANCEL = 21

PARK_START = 30

STARTRINGING = 65

HOLD = 35

UNHOLD = 36

DISCONNECT = -1

5.6. notify_voicemail

Description:

Information about user voicemail.

Parameters:

user – user extension

new – the number of new voicemail messages

unread – the number of unread voicemail messages

read – the number of read voicemail messages

saved – the number of saved voicemail messages

6. Brekeke PAL WebSocket Request Methods

6.1. barge

Permissions:

admin

Description:

Allows a user to barge into a conversation. The barging user is the extension specified in the user parameter. The barging user will be able to choose whether to listen or speak to the participants in the conversation.

Parameters:

tenant – tenant name

user – the extension of the user who is barging into the conversation

tid – an integer; the talker ID of one of the users whose conversation is barged into

listen – true or false

speak – true, false or tutor

Returns:

a success message or an error message

6.2. callforward

Permissions:

Admin

Description:

Forwards a conversation to a list of users.

Parameters:

tenant – tenant name

rid – room ID assigned to the conversation

user – an array of user extensions to whom the conversation will be forwarded

Returns:

a success message or an error message

6.3. cancelTransfer

Permissions:

admin or user

Description:

Cancel an attended transfer call issued by Brekeke PAL

Parameters:

tenant – tenant name

tid – talker ID that is in the process of being transferred

Returns:

a success message or an error message

Related methods:

transfer

6.4. createExtension

Permissions:

admin

Description:

Creates a Brekeke PBX extension.

Parameters:

tenant – tenant name

extension – a string representing an extension ID as defined in Brekeke PBX Admintool

password – a string representing the corresponding password for a user’s voicemail box

login_password – a string representing the corresponding password for a user account

type – a string representing the extension type; value can be user, ringgroup, ivr, conditional, conference or callback

If the “type” is set to non-user, the password and login_password do not need to be provided.

Returns:

If the method succeeds, it returns “true” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

deleteExtension, getExtensionProperties, getExtensions, setExtensionProperties

6.5. createTenant

Permissions:

sa

Description:

Creates a new tenant in Brekeke PBX.

Parameters:

tenant – tenant name

Returns:

If the method succeeds, it returns “true” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

getTenantProperties, setTenantProperties

6.6. deleteExtension

Permissions:

admin

Description:

Deletes a Brekeke PBX extension from a tenant.

Parameters:

tenant – tenant name

extension – a string representing an extension defined under the tenant

Returns:

If the method succeeds, it returns “true” in the result field.

If the method fails, it returns “false” in the result field.

Related methods:

createExtension, getExtensionProperties, getExtensions, setExtensionProperties

6.7. deleteNote

Permissions:

admin or user

Description:

Deletes a note.

Parameters:

tenant – tenant name

name – the name of the note to be deleted

Returns:

If the method succeeds, it returns “true” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

getNote, getNoteNames, setNote

6.8. deleteRouteVariables

Permissions:

sa

Description:

Deletes a route and its Route Local Variables values under the defined ARS template.

Parameters:

template – the ARS route template name

name – the route name

Returns:

If the method succeeds, it returns “OK” in the result field.

Related methods:

getRouteTemplateNames, getRouteVariables, insertRouteVariables, updateRouteVariables

6.9. disconnect

Permissions:

admin or user

Description:

Disconnects a call with either tid or rid.

Parameters:

tenant – tenant name

tid – talker ID of the user's session. Admin users can disconnect a call with either tid or rid.

rid – room ID of a call. Available only when log in with admin privilege; Admin users can disconnect a call with either tid or rid

delay – the amount of time to delay before disconnecting a call when the call is disconnected by rid

Returns:

a success message or an error message

6.10. getAllDids

Permissions:

admin

Description:

Gets the tenant's Route Local Variables within the Route Templates that are configured as DID on the Field settings page.

- ✓ *Fields that do not have the Tenant Access (List) checkbox selected in the Field settings cannot be returned.*

Parameters:

tenant – tenant name

Returns:

If the method succeeds, it returns an array of each route's Route Local Variables and value as a result.

Related methods:

setDid

6.11. getExtensionProperties

Permissions:

admin or user

Description:

Retrieves the property values of the properties specified in the property_names parameter.

Parameters:

tenant – the name of the tenant company

extension – a string representing an extension

property_names – a string array defining property names of the values to be retrieved

Returns:

If the method succeeds, it returns an array that contains the requested property_names values to the result field. If the method fails, it returns an associative array to the error field.

Related methods:

createExtension, deleteExtension, getExtensions, setExtensionProperties

6.12. getExtensions

Permissions:

admin

Description:

Returns a list of extensions that match the regular expressions filter for a particular tenant.

Parameters:

tenant – tenant name

pattern – a regular expression that the result must match

limit – restricts the number of tenants that are returned. For unlimited, set to -1.

type – a string representing the user type. Value can be user, ringgroup, ivr, conditional, conference or callback.

Returns:

If the method succeeds, it returns an array to the result field that contains the extensions for the type requested. If the method fails, it returns an associative array in the error field.

Related methods:

createExtension, deleteExtension, getExtensionProperties, setExtensionProperties

6.13. getNote

Permissions:

admin or user

Description:

Retrieves information about a note.

Parameters:

tenant – tenant name

name – the name of the note

Returns:

If the method succeeds, it returns the following list in the result field:

useraccess – access level; No access: 0; Read only: 1; Read/Write: 2

description – description

name – note name

note – note content

If the method fails, it returns an associative array in the error field.

Related methods:

deleteNote, getNoteNames, setNote

6.14. getNoteNames

Permissions:

admin or user

Description:

Returns a list of note names.

Parameters:

tenant – tenant name

Returns:

If the method succeeds, it returns an array that contains the note names in the result field.

Related methods:

deleteNote, getNote, setNote

6.15. getPlanSwitchTimer

Permissions:

admin or user

Description:

Retrieves the Timer settings.

Parameters:

extension – a string representing an extension

number – 1 or 2 represents Timer 1 or Timer 2 in the extension setting. Default setting is 1.

Returns:

If the method succeeds, it returns a string in the result field formatted as "P<start-end-date>A<days-of-week>W<weeks>D<date-pattern>T<times>." For details about the returned string format, refer to the "Schedule Extension" table in the "Extension Settings Properties" section.

Related methods:

setPlanSwitchTimer

6.16. getRouteTemplateNames

Permissions:

admin

Description:

Retrieves a list of tenant route template names.

Parameters:

tenant – tenant name

Returns:

If the method succeeds, it returns an array that contains the Route Names in the result field.

Related methods:

deleteRouteVariables, getRouteVariables, insertRouteVariables, updateRouteVariables

6.17. getRouteVariables

Permissions:

sa

Description:

Retrieves the route name and its Route Local Variables values assigned to the defined tenant and created under the defined route template name.

Parameters:

tenant – tenant name

template – name of the ARS route template

Returns:

If the method succeeds, it returns an array that contains each route's Route Local Variables in the result field. See below for more details:

val 0 – 1: Enabled, 0: Disabled

val 1 – tenant name

val 2 – route name

val 3 – password

val 4~5 – <reserved>

val 6~14 – v1~v9

val 15~23 – w1~w9

val 24~32 – x1~x9

Related methods:

deleteRouteVariables, getRouteTemplateNames, insertRouteVariables, updateRouteVariables

6.18. getTenantProperties

Permissions:

sa

Description:

Returns the list of tenant properties specified in the property_names array.

Parameters:

tenant – tenant name

property_names – a string array containing the property values to retrieve

- ✓ *Valid property names are: tenantid, desc, maxrecordingsessions, maxusers and maxsessions.*

Returns:

If the method succeeds, it returns an array that contains the requested property_names values in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

createTenant, setTenantProperties

6.19. insertRouteVariables

Permissions:

sa

Description:

Adds a new route and its Route Local Variables values to a specified route template.

Parameters:

template – the name of an ARS route template

values – a string array containing the route name and its related Route Local Variable setting

val 0 – 1: Enabled, 0: Disabled

val 1 – tenant name

val 2 – route name

val 3 – password

val 4~5 – <reserved>

val 6~14 – v1~v9

val 15~23 – w1~w9

val 24~32 – x1~x9

val 0 to val 6 are required variables for this method. Set an empty string for any variable that is not needed.

Returns:

If the method succeeds, it returns “true” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

deleteRouteVariables, getRouteTemplateNames, getRouteVariables, updateRouteVariables

6.20. makeCall

Permissions:

admin or user

Description:

Places a call between a caller and callee.

Parameters:

tenant – tenant name

user – a PBX user extension representing the call owner

from – the caller's user extension or phone number

to – an array of the callee's user extensions or phone numbers

type – Strings: "1" or "2"

Type "1" will simultaneously call the "from" number and "to" number and then connect them.

Type "2" will call the "from" number first. When the "from" number picks up, it calls the "to" number then connects the two calls.

Returns:

a success message or an error message

6.21. park

Permissions:

admin or user

Description:

Parks a call.

Parameters:

tenant – tenant name

tid – talker ID whose call will be parked

number – the number for retrieving the parked call

Returns:

a success message or an error message

Related methods:

unpark

6.22. setDid

Permissions:

admin

Description:

Sets the Route Local Variables within the tenant's Route Templates that are configured as DID on the Field settings page.

✓ *Fields that do not have the Tenant Access (Edit) checkbox selected in Field settings cannot* *be returned.*

Parameters:

template – the name of an ARS route template

name – Route Name

enabled – 1: Enabled, 0: Disabled

<Variable> - <value> – The variable can be v1~v9, w1~w9 or x1~x9.

Sample request:

```
{"jsonrpc":"2.0",    “method”:"setDid",    “params”:{"template":"temp1",
"name":"n1", "v1":"value1", "v2":"value2"}, "id":1}
```

Returns:

If the method succeeds, it returns “OK” in the result field.

Related methods:

getAllDids

6.23. setExtensionProperties

Permissions:

admin or user

Description:

Sets the values specified in the properties associative array to the extension properties.

Parameters:

tenant – the name of the tenant company

extension – a string representing a username as defined in Brekeke PBX Admintool

properties – an associative array containing the properties to be configured

Returns:

If the method succeeds, it returns “OK” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

createExtension, deleteExtension, getExtensionProperties, getExtensions

6.24. setNote

Permissions:

admin or user

Description:

Sets information for a note in the defined tenant.

Parameters:

tenant – tenant name

name – note name

description – description

useraccess – access levels are No access: 0; Read only: 1; Read/Write: 2.

note – note content

Returns:

If the method succeeds, it returns “OK” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

deleteNote, getNote, getNoteNames

6.25. setPlanSwitchTimer

Permissions:

user

Description:

Sets timer information.

Parameters:

extension – a string representing an extension. It can only be user or schedule extensions.

number – 1 or 2 represents Timer 1 or Timer 2 in the extension setting. Default setting is 1.

schedule – a string with format “P<start-end-date>A<days-of-week>W<weeks>D<date-pattern>T<times>.” For details about the returned string format, refer to the “Schedule Extension” table in the “Extension Settings Properties” section.

Returns:

If the method succeeds, it returns “OK” in the result field.

Related methods:

getPlanSwitchTimer

6.26. setTenantProperties

Permissions:

sa

Description:

Sets the values specified in the properties associative array to the Tenant properties.

Parameters:

tenant – tenant name

properties – an associative array containing the properties to be configured

- ✓ *Valid property names are: tenantid, desc, maxrecordingsessions, maxusers and maxsessions.*

Returns:

If the method succeeds, it returns “OK” in the result field.

If the method fails, it returns an associative array in the error field.

Related methods:

createTenant, getTenantProperties

6.27. startRecording

Permissions:

admin or user

Description:

Records a conversation.

Parameters:

tenant – tenant name

tid – talker ID whose conversation will be recorded

Returns:

A success message or an error message.

Related methods:

stopRecording

6.28. stopRecording

Permissions:

admin or user

Description:

Stops recording.

Parameters:

tenant – tenant name

tid – talker ID whose recording will be stopped

Returns:

a success message or an error message

Related methods:

startRecording

6.29. transfer

Permissions:

admin or user

Description:

Transfers a call.

Parameters:

tenant – tenant name

user – the extension to transfer the call to

tid – the talker ID of the caller whose conversation will be transferred

mode – when mode is set as blind, a blind transfer will be performed; otherwise, it will be an attended transfer.

Returns:

a success message or an error message

6.30. unpark

Permissions:

admin or user

Description:

Unparks a call.

Parameters:

tenant – tenant name

user – the user extension number with which the call will be unparked

number – the retrieve number

Returns:

a success message or an error message

Related methods:

Park

6.31. updateRouteVariables

Permissions:

sa

Description:

Sets a route and its Route Local Variables values to a specified route template.

Parameters:

template – the name of an ARS route template

update_password – true or false

values – a string array containing route name and related Route Local Variables settings

val 0 - 1: Enabled, 0: Disabled

val 1 – the tenant name

val 2 – the route name

val 3 – password

val 4~5 – <reserved>

val 6~14 – v1~v9

val 15~23 – w1~w9

val 24~32 – x1~x9

val 0 to val 6 are required variables for this method. Set an empty string for any variables that are not needed.

Returns:

If the method succeeds, it returns “true” in the result field.

Related methods:

deleteRouteVariables, getRouteTemplateNames, getRouteVariables, insertRouteVariables

7. Extension Settings Properties

The tables below contain the property names that may be viewed or altered using the WebSocket methods – these lists are not comprehensive:

7.1. Callback Extension

Property Name	Description	Value
type	Extension type	callback
callback.callee	Callback callee	Extension number
desc	Description	Text
noanswerforward	Forwarding destination (No answer)	Extension number
ringertime	Ringer time	Number

7.2. Conference Extension

Property Name	Description	Value
type	Extension type	conference
callback.callee	Callback callee	Extension number
conf.accept	Applied to (Caller numbers)*	Wildcard pattern
desc	Description	Text
exit.on.host.disconnected	Exit all when host leaves	true or false
phoneforward	Forwarding destinations	Comma-separated extension numbers

7.3. Groups Extension

Simultaneous Ring

Property Name	Description	Value
type	Extension type	ringgroup
stype	Specific type	sr
desc	Description	Text
noanswerforward	Forwarding destination (No answer)	Extension number
phoneforward	Forwarding destinations	Comma-separated extension numbers
ringertime	Ringer time	Number

Call Hunting

Property Name	Description	Value
type	Extension type	ringgroup
stype	Specific type	rr
desc	Description	Text
noanswerforward	Forwarding destination (No answer)	Extension number
switchmode	Mode	- cyclic - ascending
phoneforward	Hunt group extensions	Comma-separated extension numbers
queuingcallinterval	Call interval (msec)	Number
queuingmax	Max number of calls in the queue	Number
queuingtime	Waiting time in the queue	Number
ringertimes	Ringer time	Numbers separated by comma
round.tryonce	Single attempt	true or false

7.4. IVR Extension

Auto Attendant

Property Name	Description	Value
type	Extension type	ivr
stype	Specific type	aa
desc	Description	Text
ex.autotransfer	Default operator	User extension
ex.calltimesec	Ring timeout (sec)	Number
ex.digitmaxlength	Max input digits	Number
ex.dtmfwaittimesec	DTMF timeout (sec)	Number
ex.maxretry	Max retry count	Number
ex.speeddial	Speed dial	
ex.useronly	Transfer to unregistered users	true or false
language	Language	en or ja

Add/Remove forwarding destinations

Property Name	Description	Value
type	Extension type	ivr
stype	Specific type	fm
desc	Description	Text
language	Language	en or ja
fm.targetusers	Target groups*	Group extension numbers, separated by commas

Switch Plan

Property Name	Description	Value
type	Extension type	ivr
stype	Specific type	ptn
desc	Description	Text
language	Language	en or ja
ptn.index	Plan number	Plan number
ptn.toggle	On/Off	true or false

Script

Property Name	Description	Value
type	Extension type	ivr
stype	Specific type	scr
desc	Description	Text
ivr.script.autoanswer	Auto Answer	true or false
ivr.script.function	Function name	Text
ivr.script.note	Note name	Text
ivr.script.parameter	Parameter	
language	Language	en or ja

Flow

Property Name	Description	Value
type	Extension type	ivr
stype	Specific type	fl
desc	Description	Text

Property Name	Description	Value
ivr.flow.autoanswer	Auto Answer	true or false
ivr.flow.name	Flow name	Text
ivr.flow.parameter	Properties	
language	Language	en or ja

7.5. Schedule Extension

Property Name	Description	Value
type	Extension type	conditional
desc	Description	Text
pln[n]_d_noanswerforward	Plan[n] default Forwarding Schedule Call Forwarding destination	Extension number
pln[n]_d_p[n]_paging	Plan[n] phone[n] default paging	true or false
pln[n]_ptn[n]_callerroute	Route selection	- any - external - internal
pln[n]_ptn[n]_filtertext	Plan[n] Forwarding Schedule[n] Filter setting	Regular expression
pln[n]_ptn[n]_filtertype	Plan[n] Forwarding Schedule[n] Filter match radio box	match or unmatch
pln[n]_ptn[n]_timeschedule	Plan[n] Forwarding Schedule[n] time schedule setup	Format: P<start-end-date>A<days-of-week>W<weeks>D<date-pattern>T<times> Sample setup: - P<start-end-date> Pyyyymmddyyyymmdd - T<times> Thhmmhhmmhhmmhhmm - D<date-pattern> [Include] or [Exclude] days separated by comma - A<days-of-week> and W<weeks> Decimal value equals the binary number represented by bit flag of selected fields. Right side holds higher-value

Property Name	Description	Value
		digit.
pln[n]_ptn[n]_noanswerforward	Plan[n] Forwarding Schedule[n] Call Forwarding destination	Extension number
pln[n]_ptn[n]_p[n]_paging	Plan[n] Forwarding Schedule[n] phone[n] paging	true or false

7.6. User extension

Property Name	Description	Value
admin	If an admin user	true or false
allowjoin	Allow others to join my conversation	true or false
automonitor	Automatic Monitoring	Comma-separated user extensions
busyforward.voicemail	If forward, call to voicemail when user is busy	true or false
canjoin	Join other's conversation	true or false
defaultpickup	Call pickup group	A group extension number
desc	User description	Text
email	Email address	Email addresses where voicemails will be forwarded, separated by comma
emailattachment	Attach WAV file to email	true or false
emailnotification	Enable email notification or not	true or false
greetingtype	Greeting message	1 = Personal 2 = Alternative 3 = Default System
language	Language	en or ja
login.password	User login password	Text
maxsessioncount	Max inbound sessions	-1 means unlimited, or 0 thru 6
messageforward	Message forwarding	A list of user extensions, separated by comma
name	Display name of this user	Text
noanswerforward.voicemail	If forward, call to user	true or false

Property Name	Description	Value
	voicemail when user does not answer the call	
password	User voicemail box password	Text
pln[n]_d_anothercall.beep	Beep on Incoming Call	true or false
pln[n]_d_busyforward	[Call Forwarding] > [Forwarding Destination (Busy)]	Text
pln[n]_d_callnext	Call next phone if phone stops ringing	true or false
pln[n]_d_knockknock	Knock Knock period	Number
pln[n]_d_knockknock_onlyinternal	Only from internal extension	true or false
pln[n]_d_noanswerforward	[Call Forwarding] > [Forwarding Destination (No answer)]	Text
pln[n]_d_phoneforward	[Call Forwarding] > [Other Forwarding Destinations]	Comma-separated list of phone numbers
pln[n]_d_ringertime	[Call Forwarding] > [Ringer time (sec)]	Number
pln[n]_d_p[n]_delay	Plan[n] phone[n] [Delay (sec)] setting	Number
pln[n]_d_p[n]_enabled	If enabled, plan[n] phone[n]	true or false
pln[n]_d_p[n]_paging	Plan[n] phone[n] default paging	true or false
pln[n]_d_p[n]_ringertime	Plan[n] phone[n] ringer time	Number
pnumber[n]	Phone[n] ID	Text
p[n]_ptype	Phone[n] Type	Phone type set at [Options] > [Phone Type]
recording	Call Recording	True or false
recording.pattern	Call Recording Patterns	Comma-separated list of patterns
recordlength	Message recording length	Number
resourcemap	Resource map	Text
type	User type	User
userclass	User class	User class options set in [Options] > [User Access Settings]

Property Name	Description	Value
voicemail.readcallerid	Talking caller ID	true or false
voicemail.password.enter.type	Skip password from my phone|	0: no 1: yes

- ✓ *pln[n]_d_xxxx are properties' names for the user extension [Inbound] page plan [n] Default Forwarding Schedule settings.*
- ✓ *pln[n]_ptn[n]_xxxx are properties' names for the user extension [Inbound] page plan [n] Forwarding Schedule [n] settings.*
- ✓ *User Forwarding Schedule [n] properties' names are the same as those in Default Forwarding Schedule settings, but with a different prefix. For the Forwarding Schedule [n] Conditions properties' names, please check the Schedule Extension table.*

8. Sample Programs

8.1. Example 1

This is a sample program demonstrating how to retrieve property values using a string array of property names:

```
<script>
var socket = null;
var host = 'ws://192.168.200.10:18080/pbx/ws?tenant=test&login_user=1000&lo
gin_password=1000';

if('WebSocket' in window){
    socket = new WebSocket(host);
}else if('MozWebSocket' in window){
    socket = new MozWebSocket(host);
}

socket.onmessage = function(event){
    var response = JSON.parse(event.data);
    var properties = ["name", "desc", "language", "login.password", "admin
"];
    var request = '{"jsonrpc": "2.0", "method": "getExtensionProperties", "
params": {"extension": "0001", "property_names":properties}, "id": 1};
    var json_request = JSON.stringify(request);
    socket.send(json_request);

    switch(response['id']){
        case '1':
         for(var i = 0; i < properties.length; i ++)
           console.log(properties[i] + ' = ' + response['result'][i]);
         break;
    }
}
</script>
```

8.2. Example 2

This is a sample program demonstrating how to set property values using string arrays of property names and property values:

```
<script>
var socket = null;
var host =
'ws://192.168.200.10:18080/pbx/ws?tenant=test&login_user=1000&login_passwor
d=1000';
if('WebSocket' in window){
    socket = new WebSocket(host);
}else if('MozWebSocket' in window){
    socket = new MozWebSocket(host);
}
socket.onmessage = function(event){
    var response = JSON.parse(event.data);
    var properties = {"name":"0001", "desc":"admin", "language":"en",
"login.password":"0001", "admin":"true"};
    var request = '{"jsonrpc": "2.0", "method": "setExtensionProperties",
"params": {"extension": "0001", "properties": properties}, "id":1};
    var json_request = JSON.stringify(request);
    socket.send(json_request);

    switch(response['id']){
        case '1':
          console.log(response['result']);
          break;
    }
}
</script>
```