Ruckus Wireless™ SmartCell Gateway 200, Virtual SmartZone High-Scale and SmartZone 300 for SmartZone 3.5

Hotspot WISPr Reference Guide

Part Number: 800-71299-001 Rev A
Published: 23 March 2017

www.ruckuswireless.com

Contents

Copyright Notice and Proprietary Information

Destination Control Statement

Technical data contained in this publication may be subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

THIS DOCUMENTATION AND ALL INFORMATION CONTAINED HEREIN ("MATERIAL") IS PROVIDED FOR GENERAL INFORMATION PURPOSES ONLY. RUCKUS AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THE MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR THAT THE MATERIAL IS ERROR-FREE, ACCURATE OR RELIABLE. RUCKUS RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO THE MATERIAL AT ANY TIME.

Limitation of Liability

IN NO EVENT SHALL RUCKUS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN CONTRACT OR TORT, ARISING FROM YOUR ACCESS TO, OR USE OF, THE MATERIAL.

Trademarks

About this Guide

This SmartCell Gateway™ (SCG) 200 / 300 and Virtual SmartZone High-Scale (vSZ-H) Hotspot WISPr Reference Guide describes the SCG-200/vSZ-H (collectively referred to as “the controller” throughout this guide) RESTful-like/JSON interfaces for external web portal servers.

This guide is written for service operators and system administrators who are responsible for managing, configuring, and troubleshooting Ruckus Wireless devices. Consequently, it assumes a basic working knowledge of local area networks, wireless networking, and wireless devices.

NOTE If release notes are shipped with your product and the information there differs from the information in this guide, follow the instructions in the release notes.

Most user guides and release notes are available in Adobe Acrobat Reader Portable Document Format (PDF) or HTML on the Ruckus Wireless Support Web site at https://support.ruckuswireless.com/contact-us.

Document Conventions

Table 1: Text conventions on page 5 and Table 2: Notice conventions on page 5 list the text and notice conventions that are used throughout this guide.

Table 1: Text conventions

Convention	Description	Example
`message phrase`	Represents information as it appears on screen	`[Device Name] >`

`user input`	Represents information that you enter	`[Device Name] >`
		`set ipaddr 10.0.0.12`

user interface controls	Keyboard keys, software buttons, and field names	Click **Start** > **All Programs**
screen or page names		Click **Advanced Settings**. The **Advanced Settings** page appears.

Table 2: Notice conventions

Notice type	Description
NOTE	Information that describes important features or instructions

Notice type	Description
CAUTION!	Information that alerts you to potential loss of data or potential damage to an application, system, or device
WARNING!	Information that alerts you to potential personal injury

Terminology

The table lists the terms used in this guide.

Table 3: Terms used in this guide

Terminology	Description
AP	Access Point
CP	Captive Portal
NBI	Northbound Interface
RADIUS	Remote Access Dial In User Service
SCG	Smart Cell Gateway
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
UDI	User Define Interface
UE	User Equipment
UE-IP	User Equipment - IP Address
UE-MAC	User Equipment - MAC Address

Related Documentation

For a complete list of documents that accompany this release, refer to the Release Notes.

Online Training Resources

To access a variety of online Ruckus Wireless training modules, including free introductory courses to wireless networking essentials, site surveys, and Ruckus Wireless products, visit the Ruckus Wireless Training Portal at:

https://training.ruckuswireless.com.

Documentation Feedback

Ruckus Wireless is interested in improving its documentation and welcomes your comments and suggestions.

You can email your comments to Ruckus Wireless at: docs@ruckuswireless.com

When contacting us, please include the following information:

- Document title
- Document part number (on the cover page)
- Page number (if appropriate)
- For example:
  - Administrator Guide for SmartZone 3.5
  - Part number: 800-xxxx-001
  - Page 88

Web Interface Configuration Overview 1

The controller provides Wi-Fi hotspot services in conjunction with external web portal servers. In most cases, an external web portal server provides the landing web pages with Wi-Fi hotspot usage instructions, terms and conditions, etc., while the end user submits his login ID and password directly to the AP for authentication.

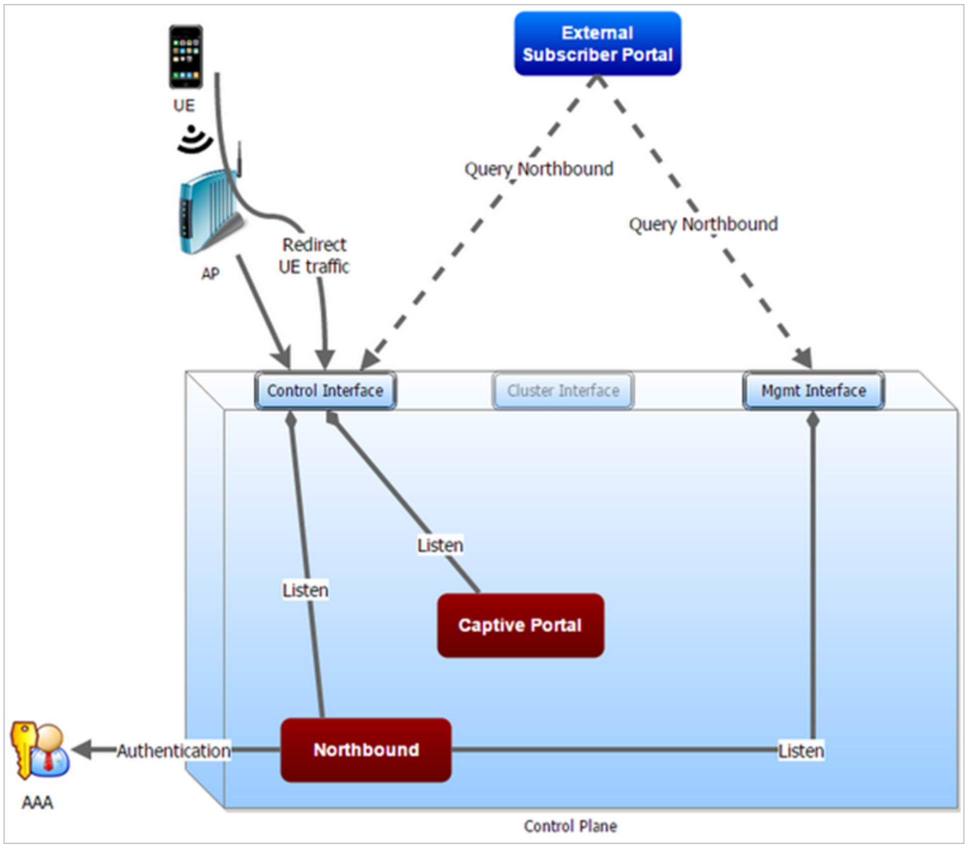
There are, however, some cases when an external web portal server requires total control of a user session by requesting authentication on the user's behalf as well as terminating user sessions. JSON interface defined in this reference guide provides a standard way for an external web portal server to communicate with the controller for this kind of usage.

The following are the hotspot components and their roles in the hotspot portal as seen in the Figure

- Northbound: Listens on the control and management interface. It is responsible for handling requests from external subscriber portal and authenticates with the AAA server.
- Captive portal: Listens on the control interface or UDI. It is responsible for providing a wall garden for web-proxy UE. It blocks UEs, which uses user agents that are listed in the configured black-list and mainly handles high scalable redirecting UEs to the external subscriber portal.
- External subscriber portal: Is a Web service. The user sends his/her login credentials (username and password) through this portal. The authentication is performed through the northbound by user input credential. The external subscriber portal can reach the northbound depending on the type of interface it can reach such as control interface, management interface or both.
- AAA server: Is responsible for authenticating the UE through the UE's login credentials (username and password).

NOTE Refer to appendix WISPr Portal Details on page 43 for IPv4 and IPv6 protocol support for GRE tunnels.

Figure 1: Hotspot portal components



This reference guide describes the controller RESTful-like/JSON interfaces for external web portal servers.

NOTE Refer to About This Guide chapter for conventions used in this guide.

Request Format

As defined in JSON commands, each request issued from an external web portal server is in JSON format.

NBI is only accessible via the management, control and user defined interfaces. The following are the request formats.

HTTP Request

```
http://scg_management_ip:9080/portalintf
```

HTTPS Request

```
https://scg_management_ip:9443/portalintf
```

NOTE You can download the log for northbound portal interface from the controller web interface by navigating to **Diagnostics** > **Application Logs** as all other applications.

The table lists the ports that must be opened on the network firewall to ensure that the controller and NBI can communicate with each other successfully.

Table 4: Portal details

Port Number	Layer 4 Protocol	Source	Destination	Configurable from Web Interface?	Purpose
9080	HTTP	Any	Controller	No	Northbound Portal Interface for Hotspot
9443	HTTPS	Any	Controller	No	Northbound Portal Interface for Hotspot

Controller Web Interface Configuration

Each JSON request must be accompanied by a request password that is preconfigured on the controller, as well as on the external web portal server.

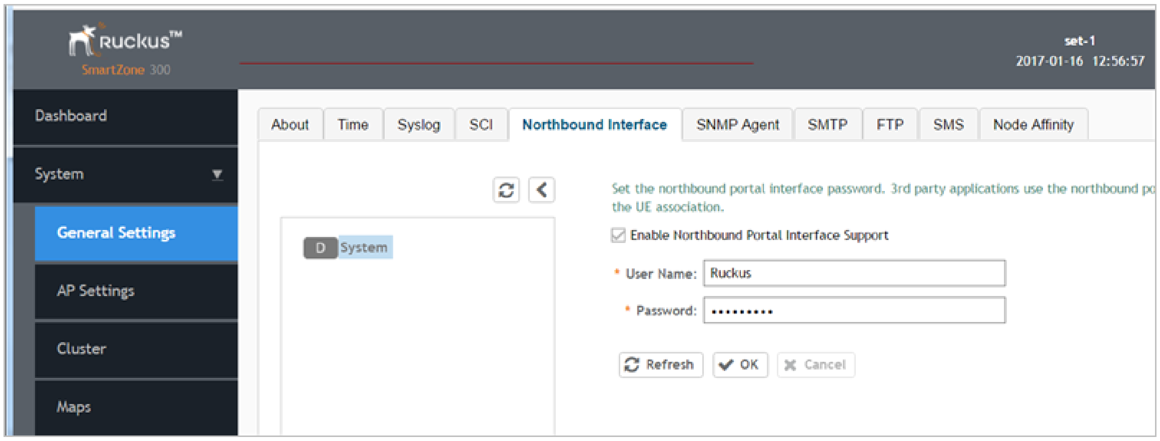
This helps ensure that only authorized web portal servers can access the northbound portal interface.

The northbound portal interface request password can be configured in the controller web interface by navigating to **System > General System Settings > Northbound Interface**. See Figure 2: Setting the password on page 11.

The password in the figure is a token to ensure that a portal has the permission to get the services from the northbound portal interface. It must be included in all JSON request as RequestPassword sent to NBI.

A web portal server must use the POST command to issue JSON requests. The controller will not accept a request with the GET request command.

Figure 2: Setting the password



Set the northbound portal interface password. 3rd party applications use the northbound po
the UE association.

☑ Enable Northbound Portal Interface Support

* User Name: Ruckus

* Password: •••••••••

NOTE Refer to Northbound Portal Interface Support on page 42 for details on MSP support.

JSON Commands - User Online Control 2

The Northbound Portal interface supports the following JSON commands:

- Login
- Login Async
- Logout
- Status
- Disconnect
- Enrichment Info

These commands are used for user authentication, user status query, terminating user sessions and verifying that the enrichment information has the same content. For each command (JSON POST), both the UE-IP and UE-MAC may be included. Where both are present, the UE-MAC will be preferred.

The NBI decrypts the strings and returns the decrypted version within the response message. This is because the Captive Portal (CP) encrypts the IP and MAC address parameters in each redirection (See the Table for the full list of these parameters) to the subscriber portal. The controller decrypts the UE-IP and UE-MAC address before returning the response, by using the Encrypt and Decrypt API described in GetConfig section.

NOTE Northbound Interface (NBI) expects to receive encrypted UE-IP and UE-MAC address (For example, ENC12bc24c4777703327f2e0aabbf6b9f9e) when the request category is UserOnlineControl. In the GetConfig request category you do not need to encrypt UE-IP and UE-MAC address (For example: 172.21.134.87)

Request Authentication - Asynchronous Login

In the Hotspot (WISPr) WLAN use case, an unauthorized user is redirected to an external web portal server by the controller. Using the asynchronous login command (RequestType=LoginAsync), the external web portal server sends a request to the controller to authenticate the user using the RADIUS server.

The external Web portal server receives the response - 202 Authentication pending, while the controller performs the authentication in the background. It is the responsibility of the Web portal to poll the controller and fetch the authentication result. This action is performed using the status command (RequestType=Status).

NOTE To use asynchronous APIs refer to Using Asynchronous API

The following is an example of the asynchronous login request:

```
{
 Vendor: “ruckus”
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “UserOnlineControl”,
 RequestType: “LoginAsync”,
 UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
 UE-MAC: “ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157”,

 UE-Proxy: “0”,
 UE-Username: “test”,
 UE-Password: “test”
}
```

The table lists the controller responses to these authentication requests.

NOTE The user account test (UE username) mentioned in the above example, has been created as an external user in the RADIUS server. The hotspot portal does not provide an interface for manipulating user account information.

Table 5: Controller responses to authentication (asynchronous login) requests

Response Type	Possible Responses
Normal response	• 101, Client authorized: Response if the user is already authorized. • 202, Authentication pending: Authentication is in progress, portal server needs to check the result later.
Service error	• 300, Not found: Response if the lookup fails with given UE-MAC or UE-IP address. • 400, Internal server error: Response when the controller internal error occurs.
General error	• 302, Bad request: Response if the JSON request is not well-formed. • 303, Version not supported: Response if there is a version mismatch. • 304, Command not supported: Response if the request type is not supported. • 305, Category not supported: Response if the request category not supported. • 306, Wrong request password: Response if the request password is mismatched.

Using Asynchronous API

When using the asynchronous API (RequestType = LoginAsync), NBI will always return a response as pending authentication.

The client must send a status request (each X seconds/milliseconds) to check for the authentication result. This is useful when using a smart device. The App in a smart device can query the login status periodically. It stores the user credentials in the background thereby reducing the user driven actions.

Request Authentication Synchronous Login

The controller also provides a synchronous login blocking command (RequestType=Login).

In synchronous login command the external Web portal must wait for the authentication process to complete, which is usually processed by the RADIUS server. This could result in a delayed response if the controller is unable to get a response from the RADIUS server. The following is an example of this command.

```
{
 Vendor: “ruckus”
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “UserOnlineControl”,
 RequestType: “Login”,
 UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
 UE-MAC: “ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157”,

 UE-Proxy: “0”,
 UE-Username: “test”,
 UE-Password: “test”
}
```

The table lists the controller responses to the synchronous login command.

Table 6: Controller responses to a synchronous login command

Response Type	Possible Responses
Normal response	• 101, Client authorized: Response if the user is already authorized. • 201, Login succeeded: Response if the login is accepted.
Service error	• 300, Not found: Response if the lookup fails with given UE-MAC or UE-IP address. • 301, Login failed: It will be replaced if the RADIUS reply message is returned. • 400, Internal server error: Response when an controller internal error occurs. • 401, Radius server error: Response when a RADIUS connection error occurs or the connection request times out.
General error	• 302, Bad request: Response if the JSON request is not well-formed. • 303, Version not supported: Response if there is a version mismatch. • 304, Command not supported: Response if the request type is not supported. • 305, Category not supported: Response if the request category not supported. • 306, Wrong request password: Response if the request password is mismatched.

NOTE If an authentication process has a result (not pending), the controller responds to it only once. For example, if the controller replies 301, Login failed to the web portal server, and the web portal server sends the same query, the response will be 100, unauthorized. If the controller replies 201, Login succeeded, and the web portal server queries again, the response will be 101, Authorized.

Querying Enrichment Information

The Northbound Portal Interface provides the JSON command EnrichmentInfo for verifying that the enrichment information has the same content as HTML header *enrichment info* sent from the AP.

This allows the captive portal to obtain the enriched parameters in an SSL (Secure Sockets Layer) scenario or in other cases wherein the AP enrichment info is not available.

NOTE The EnrichmentInfo command is only applicable for UEs connected to Ruckus APs and not for 3rd party APs.

The following is an example of the *EnrichmentInfo* request:

```
{
 Vendor: “ruckus”
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “UserOnlineControl”,
 RequestType: “EnrichmentInfo”,
 UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
}
```

The table lists the responses for enrichment information.

Table 7: Query enrichment

Response Type	Possible Responses
Normal response	• 102, Enrichment Information.
Service error	• 300, Not found: Response if the lookup fails with given UE- MAC or UE-IP address. • 400, Internal server error: Response when an controller internal error occurs.
General error	• 302, Bad request: Response if the JSON request is not well-formed. • 303, Version not supported: Response if there is a version mismatch. • 304, Command not supported: Response if the request type is not supported. • 305, Category not supported: Response if the request category not supported. • 306, Wrong request password: Response if the request password is mismatched.

NOTE If an authentication process has a result (not pending), the controller responds to it only once. For example, if the controller replies 301, Login failed to the web portal server, and the web portal server sends the same query, the response will be 100, unauthorized. If the controller replies 201, Login succeeded, and the web portal server queries again, the response will be 101, Authorized.

Terminating a User Session

After a user session is authorized, the external web portal server can terminate the user session by sending a JSON request to the controller.

In this case, the Web portal changes the status of the client from authenticated, to unauthenticated, forcing the user to login again. When un-authenticating a user, existing TCP sessions are not terminated and the UE is not disassociated from the AP. It only changes the status of the UE from authorized to unauthorized. The following is an example of the terminating a user session command:

```
{
 Vendor: “ruckus”
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “UserOnlineControl”,
 RequestType: “Logout”,
 UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
 UE-MAC: “ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157”
}
```

Disconnect Command

The controller also provides a command for terminating user TCP (Transmission Control Protocol) connections from the AP (Access Point).

In other words, the disconnect command (RequestType=Disconnect) changes the status of the UE from authorized to unauthorized and also disassociates the UE from the AP.

```
{
 Vendor: “ruckus”
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “UserOnlineControl”,
 RequestType: “Disconnect”,
 UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
 UE-MAC: “ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157”
}
```

The table lists the controller response.

Table 8: Controller responses to a disconnect command

Response Type	Possible Responses
Normal response	• 200, OK • 100, Client unauthorized: Response if the user is already unauthorized
Service Error	• 300, Not found: Response if the lookup fails with given UE- MAC or the UE-IP address. • 400, Internal server error: Response when an controller internal error occurs.
General error	• 302, Bad request: Response if the JSON request is not well-formed. • 303, Version not supported: Response if there is a version mismatch. • 304, Command not supported: Response if the request type is not supported. • 305, Category not supported: Response if the request category not supported. • 306, Wrong request password: Response if the request password is mismatched.

JSON Responses - GetConfig 3

The northbound interface supports the following JSON commands in the request category - GetConfig:

- Control Blade IP List
- Cluster Blade IP List
- Management Blade IP List
- User Interface IP List
- Encrypt
- Decrypt

NOTE It is recommended for new users to implement and use the new APIs - Encrypt and Decrypt. Existing users can continue using the legacy APIs - EncryptIP and DecryptIP provided; you have not made any changes to it during implementation on your portal server.

The first four commands are used for obtaining the different blade IP lists. The northbound portal interface simply responds with the control, cluster, management blade or user defined IP list of the controller. The following is an example of the GetConfig command:

```
{
 Vendor: “ruckus”,
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “GetConfig”,
 RequestType: “ControlBladeIPList”,
```

The following is an example of the success response:

```
{
   Vendor:“ruckus”,
   ReplyMessage:”OK",
   ResponseCode:200,
   APIVersion:"1.0"
   ControlBladeIPList:[“172.17.18.149”, “172.17.18.159”,
“172.17.18.169”]
}
```

Control Blade IP address list can be replaced by Cluster Blade IP List, Management Blade IP List or User Interface IP List, depending on context of the GetConfig command.

The following is an example of an Encrypt IP address command, which returns an encrypted IP address for direct access to the subscriber portal. By default the encryption is enabled. To disable the encryption, use the CLI command:

```
ruckus(config)# [no] encrypt-mac-ip
```

NOTE Refer to the CLI examples given below for enabling disabling the IP and MAC address encryption.

```
{
 Vendor: “ruckus”,
 RequestPassword: “myPassword”,
 APIVersion: “1.0”,
 RequestCategory: “GetConfig”,
 RequestType: “Encrypt”,
Data: “172.21.134.87”
}
```

The following is an example of the success response:

```
{
   Vendor: “ruckus”,
   ReplyMessage:”OK",
   ResponseCode:200,
   APIVersion:”1.0”
   Data: “ENC1234bfdbe5y5hbfdgh45y54ryt5y5th5”
}
```

Another example is the decrypt command, which returns a decrypted value of IP address.

```
{
Vendor: "ruckus",
RequestPassword: "myPassword", APIVersion: "1.0",
RequestCategory: "GetConfig", RequestType: "Decrypt",
Data: "ENC1234bfdbe5y5hbfdgh45y54ryt5y5th5"
}
```

The success response:

```
{
Vendor:"ruckus", ReplyMessage:"OK", ResponseCode:200,
APIVersion:"1.0"
Data: "172.21.134.87"
}
```

The following are examples of using the CLI command for enabling and disabling the IP address and MAC address encryptions.

Enabling the IP address and MAC address encryption:

```
# show running-config encrypt-mac-ip
```

Disabling the IP address and MAC address encryption:

```
# config
(config)# no encrypt-mac-ip
Do you want to continue to disable (or input 'no' to cancel)?
[yes/no] yes
Successful operation
```

Confirming that the IP address and MAC address encryption is disabled:

```
(config)# do show running-config encrypt-mac-ip
Encryption MAC and IP: Disabled
```

JSON Responses

The table lists the definitions of JSON responses from the northbound portal interface.

The following are the expansions for the abbreviations mentioned in the Used In column.

- UA: User Authenticate (includes LoginSync and LoginAsync)
- SQ: Status Query
- TU: Terminating User (Logout and Disconnect)
- EI: Enrichment Information
- GC: Get Config (Control Blade IP, Cluster Blade IP, Management Blade IP, User Interface IP, Encrypt and Decrypt)

NOTE Refer to JSON Commands for commands related to the responses mentioned above.

Table 9: JSON response definitions

Category	Code	Definition	Used In				
			UA	SQ	TU	EI	GC
Informational	100	Client unauthorized		•	•		
	101	Client authorized	•	•			
	102	Enrichment Info				•	
Success	200	OK			•		•
	201	Login succeeded		•			
	202	Authentication pending	•	•			
Client Error	300	Not found	•	•	•	•	
	301	Login failed	•	•			
	302	Bad request	•	•	•	•	•
	303	Version not supported	•	•	•	•	•
	304	Command not supported					
	305	Category not supported					
	306	Wrong request password	•	•	•	•	•
Server Error	400	Internal server error	•	•	•	•	•
	401	Radius server error	•	•			

JSON Response Examples

This section provides the following examples of JSON responses defined in the table (JSON Response Definitions)

Example: Client unauthorized

```
{
Vendor:"Ruckus",
APIVersion:"1.0",
ResponseCode:100,
ReplyMessage:"Client unauthorized",
UE-IP:"ENC323e79bf1bbd5ac4",
UE-MAC:"ENCf6b7f49da92a45f8978c35966b95eeafc6451102af391592",
AP-MAC:"00:11:22:AA:BB:CC",
SSID:" hotspot-01",
SmartClientInfo:"",
GuestUser:"0",
SmartClientMode:"none"
}
```

Example: Client authorized

```
{
Vendor: "Ruckus",
APIVersion: "1.0",
ResponseCode: "101",
ReplyMessage: "Client authorized",
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
UE-Username: "user001",
AP-MAC: "04:4f:aa:32:25:f0",
SSID: "hotspot-01"
SmartClientMode: "none",
SmartClientInfo: "",
GuestUser: "0",
}
```

Example: Enrichment information

```
{
Vendor: "Ruckus",
APIVersion: "1.0",
ResponseCode: "102",
ReplyMessage: "Enrichment Information",
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
AP-MAC: "04:4f:aa:32:25:f0",
SSID: "hotspot-01",
WLAN-ID: "1",
Location: "a location",
VLAN-ID: 1
}
```

Example: Success information

```
{
Vendor: "Ruckus",
Version: "1.0",
ResponseCode: "200",
ReplyMessage: "OK"
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
SmartClientMode: "none",
SmartClientInfo: "",
GuestUser: "0",
}
```

Example: Login succeeded

```
{
Vendor: "Ruckus",
APIVersion: "1.0",
ResponseCode: "201",
ReplyMessage: "Login succeeded",
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
UE-Username: "user001",
AP-MAC: "04:4f:aa:32:25:f0",
SSID: "hotspot-01",
SmartClientMode: "none",
SmartClientInfo: "",
GuestUser: "0",
UE-Proxy: "0"
}
```

Example: Authentication pending

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "202",
ReplyMessage: "Authentication pending",
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
UE-Username: "user001",
AP-MAC: "04:4f:aa:32:25:f0",
SSID: "hotspot-01",
SmartClientMode: "none",
SmartClientInfo: "",
GuestUser: "0",
}
```

Example: Not found

```
{
 Vendor: "Ruckus",
 APIVersion: "1.0",
 ResponseCode: "300",
 ReplyMessage: "Not found",
}
```

Example: Login failed

```
{
Vendor: "Ruckus",
APIVersion: "1.0",
ResponseCode: "301",
ReplyMessage: "Login failed",
UE-IP: "ENC12bc24c4777703327f2e0aabbf6b9f9e",
UE-MAC: "ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157",
AP-MAC: "04:4f:aa:32:25:f0",
SSID: "hotspot-01",
SmartClientMode: "none",
SmartClientInfo: "",
GuestUser: "0",
}
```

Example: Bad request

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "302",
ReplyMessage: "Bad request",
}
```

Example: Version not supported

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "303",
ReplyMessage: "Version not supported"
}
```

Example: Command not supported

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "304",
ReplyMessage: "Command not supported",
}
```

Example: Category not supported

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "305",
ReplyMessage: "Category not supported",
}
```

Example: Wrong request password

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "306",
ReplyMessage: "Wrong request password",
}
```

Example: Internal server error

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "400",
ReplyMessage: "Internal server error",
}
```

Example: RADIUS server error

```
{
Vendor: "ruckus",
APIVersion: "1.0",
ResponseCode: "401",
ReplyMessage: "Radius server error",
}
```

Example: Encrypt for MAC address

```
{
Vendor: “ruckus”,
RequestPassword: “myPassword”,
APIVersion: “1.0”,
RequestCategory: “GetConfig”,
RequestType: “Encrypt”,
Data: “04:4f:aa:32:25:f0”
}
The success response:
{
Vendor: “ruckus”,
ReplyMessage:”OK",
ResponseCode:200,
APIVersion:”1.0”,
Data: "ENC4782689566f8eac8aa30e276aa907f332d0bf93f9f60a7d8"
}
```

Example: Decrypt for MAC address

```
{
Vendor: "ruckus",
RequestPassword: “myPassword”,
APIVersion: "1.0",
RequestCategory: "GetConfig",
RequestType: "Decrypt",
Data: "ENC4782689566f8eac8aa30e276aa907f332d0bf93f9f60a7d8"
}
The success response:
{
Vendor:"ruckus", ReplyMessage:"OK",
ResponseCode:200,
APIVersion:"1.0"
Data: "04:4f:aa:32:25:f0"
}
```

A

WISPr Support for ZoneDirector Login

The WISPr hotspot portal logon API supports existing customer’s external logon page (working with Zone Director (ZD). Customers, who already have a ZD deployment and have implemented their own external logon page for hotspot WLAN, can use ZD's API (provided by Ruckus) for UE authentication. The controller provides the same API as that of ZD for customers to use their existing logon page.

NOTE This new API is provided since controller's official portal integration using JSON requests does not support ZD login API. It is our recommendation that the customer works with the JSON API as documented in this guide - Hotspot Portal Integration Interface.

Customer Login

Customers who already have ZD deployment with their own external portal will be required to make a change to their login/logout URLs to match the new supported API.

The external portal sends the login/logout request to the controller. The requests should include the parameters provided by controller's captive portal redirection

NOTE See Captive Portal Attributes on page 32 for details.

Login: The login request path in the external portal to the controller should be changed:

From:

```
https://sip:9998/login
```

To:

```
https://sip:9998/SubscriberPortal/hotspotlogin
```

NOTE The login request also supports HTTP with port number 9997.

NOTE This login request should include the customer login credentials such as the username and password parameters. It is expected that the customer's portal also sends the following parameters from Captive Portal's redirection -

- url - the original URL which the user tried to browse
- proxy - if the UE browser is set to Web proxy
- uip - UE IP address
- client-mac - UE MAC IP address

Customer Logout

The logout request path in the external portal to the controller should be changed:

From:

```
https://sip:9998/logout
```

To:

```
https://sip:9998/SubscriberPortal/hotspotlogout?uip=10.20.30.40
```

B

Captive Portal Attributes

The UE-IP and UE-MAC address parameters are decrypted at the beginning of each user online control request. This is because the Captive Portal (CP) encrypts the IP and MAC address parameters in each redirection to the subscriber portal.

The controller decrypts the UE-IP and UE-MAC address before returning the response, by using the Encrypt and Decrypt API described in the GetConfig section.

NOTE In case the external portal is in HTTPS, Apple CNA will not work. It works only for HTTP redirect.

Redirection Attributes

The table lists these parameters provided by controller's captive portal redirection.

NOTE See WISPr Support for ZoneDirector Login for login and logout details.

Table 10: Redirection attributes

Attributes	Description
client_mac	Encrypted UE Mac address. **NOTE** The format of the MAC Address is defined at the Hotspot (WISPr) Portal configuration.
dn	The domain name.
loc	AP location.
mac	AP Mac address.
proxy	The UE browser if it is set to the Web proxy.
reason	Reason for redirecting the WLAN. The value could either be: • Un-Auth-Captive – Regular unauthenticated UE redirected to Login Portal or • Un-Auth-SSL-Captive – In case of HTTPS, Captive Portal is performing a “double redirect”. Adding this value to identify this flow

Attributes	Description
sip	The value could either be the: - UDI (User Defined Interface) in case of a local breakout from AP - Or Internal D-Blade IP address in case of a proxy request to the controller. - If the controller is a SZ100 or vSZ with no D-Blade or no UDI configured, the SIP value. on the enrichment header is the controller interface IP address or 192.168.255.1. - If there is no tunnel WISPr WLAN, the SIP value is the controller interface IP address. - If there is a tunnel WISPr WLAN, the SIP value is 192.168.255.1, for the value 192.168.255.1
ssid	The broadcasted SSID name.
startUrl	The URL as per the hotspot configuration, which is to be redirected after successful login.
uip	Encrypted UE IP address.
url	Original URL which the customer tries browsing.
vlan	VLAN which the customer is set to.
wlan	WLAN ID of the UE's associated the WLAN.
wlanName	SSIDs configured WLAN Name.
zoneId	In case of 3rd party AP, this attribute will be included instead of WLAN and will include the zone ID where the SSID is configured to in the controller.
zoneName	AP zone name of the UE's associated to the WLAN. The zone name is configured using the WLANs. The zone name is used for Kumo. The value is encrypted based on a special key.

C

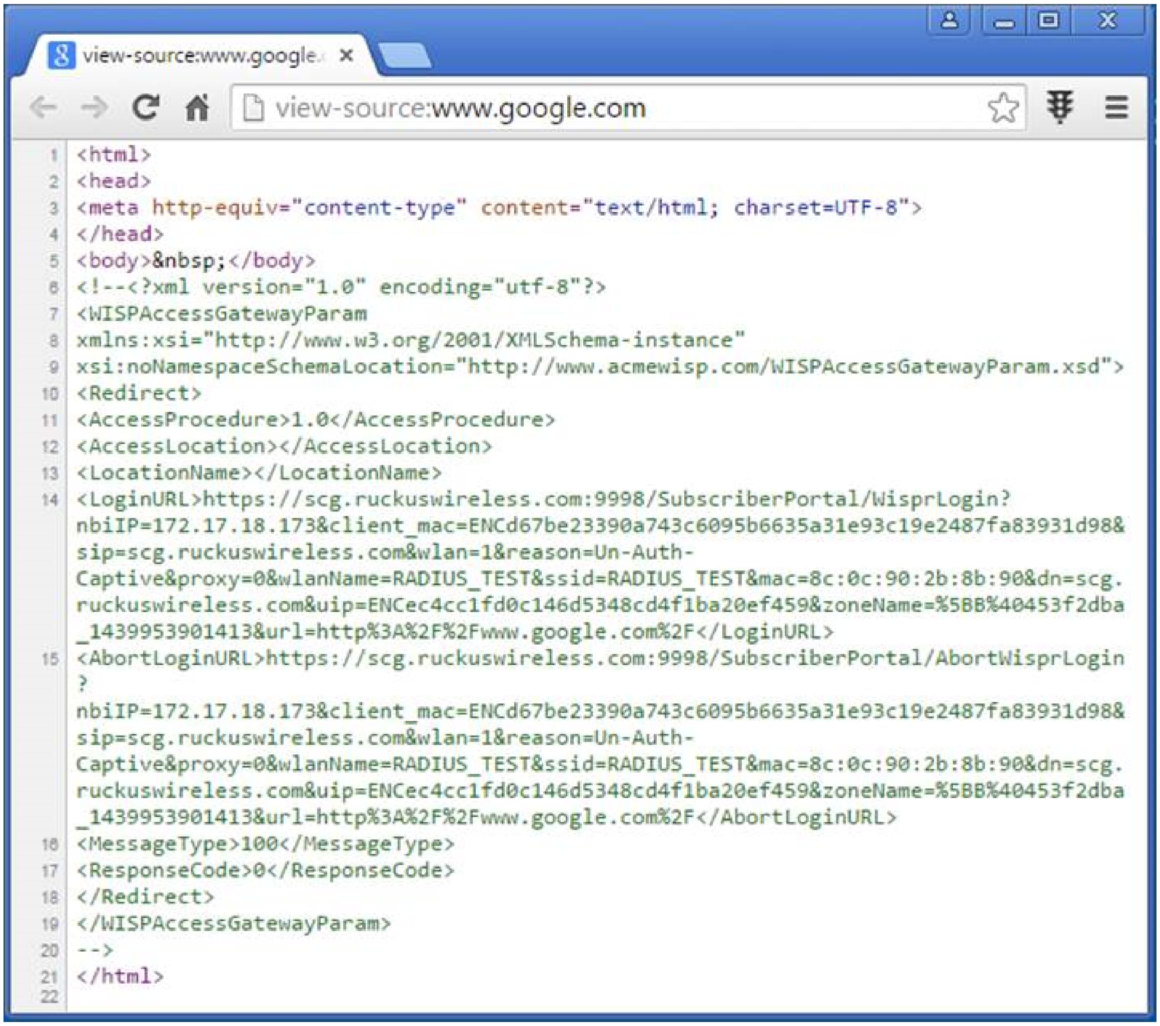
The Smart Client

The Smart Client is a software solution which resides on the user’s access device that facilitates the user’s connection to Public Access Networks, whether via a browser, signaling protocol or other proprietary method of access.

The XML is embedded in the HTML source code as a comment block as the following:

```
<html>
< head>
< meta http-equiv="content-type" content="text/html;
charset=UTF-8">
< /head>
< body></body>
<!--<?xml version="1.0" encoding="utf-8"?>
{{{ The Embedded XML }}}
-->
</html>
```

Figure 3: Smart Client Example



```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
</head>
<body>&nbsp;</body>
<!--<?xml version="1.0" encoding="utf-8"?>
<WISPAccessGatewayParam
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<Redirect>
<AccessProcedure>1.0</AccessProcedure>
<AccessLocation></AccessLocation>
<LocationName></LocationName>
<LoginURL>https://scg.ruckuswireless.com:9998/SubscriberPortal/WisprLogin?
nbiIP=172.17.18.173&client_mac=ENCd67be23390a743c6095b6635a31e93c19e2487fa83931d98&
sip=scg.ruckuswireless.com&wlan=1&reason=Un-Auth-
Captive&proxy=0&wlanName=RADIUS_TEST&ssid=RADIUS_TEST&mac=8c:0c:90:2b:8b:90&dn=scg.
ruckuswireless.com&uip=ENCec4cc1fd0c146d5348cd4f1ba20ef459&zoneName=%5BB%40453f2dba
_1439953901413&url=http%3A%2F%2Fwww.google.com%2F</LoginURL>
<AbortLoginURL>https://scg.ruckuswireless.com:9998/SubscriberPortal/AbortWisprLogin
?
nbiIP=172.17.18.173&client_mac=ENCd67be23390a743c6095b6635a31e93c19e2487fa83931d98&
sip=scg.ruckuswireless.com&wlan=1&reason=Un-Auth-
Captive&proxy=0&wlanName=RADIUS_TEST&ssid=RADIUS_TEST&mac=8c:0c:90:2b:8b:90&dn=scg.
ruckuswireless.com&uip=ENCec4cc1fd0c146d5348cd4f1ba20ef459&zoneName=%5BB%40453f2dba
_1439953901413&url=http%3A%2F%2Fwww.google.com%2F</AbortLoginURL>
<MessageType>100</MessageType>
<ResponseCode>0</ResponseCode>
</Redirect>
</WISPAccessGatewayParam>
-->
</html>
```

Extract the embedded XML as the following.

```
<?xml version="1.0" encoding="utf-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"xsi:
noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
    <Redirect>
        <AccessProcedure>1.0</AccessProcedure>
        <AccessLocation></AccessLocation>
        <LocationName></LocationName>
        <LoginURL>https://scg.ruckuswireless.com:
        9998/SubscriberPortal/
        WisprLogin?nbiIP=172.17.18.173&client_mac=
        ENCd67be23390a743c6095b6635a31e93c19e248
       7fa83931d98&sip=scg.ruckuswireless.com&wlan=1&reason=

        Un-Auth-Captive&proxy=0&
        wlanName=RADIUS_TEST&ssid=RADIUS_TEST&mac=
        8c:0c:90:2b:8b:90&dn=scg.ruckuswireless.com&
        uip=ENCec4cc1fd0c146d5348cd4f1ba20ef459&zoneName=
        %5BB%40453f2dba_1439953901413&url=
        http%3A%2F%2Fwww.google.com%2F</LoginURL>
        <AbortLoginURL>https://scg.ruckuswireless.com:
        9998/SubscriberPortal/AbortWisprLogin?
        nbiIP=172.17.18.173&client_mac=
        ENCd67be23390a743c6095b6635a31e93c19e2487fa83931d98&

        sip=scg.ruckuswireless.com&wlan=1&
        reason=Un-Auth-Captive&proxy
        =0&wlanName=RADIUS_TEST&
        ssid=RADIUS_TEST&mac=8c:0c:90:2b:8b:90&dn=
        scg.ruckuswireless.com&uip=
        ENCec4cc1fd0c146d5348cd4f1ba20ef459&zoneName=
        %5BB%40453f2dba_1439953901413&url=
        http%3A%2F%2Fwww.google.com%2F</AbortLoginURL>
        <MessageType>100</MessageType>
        <ResponseCode>0</ResponseCode>
    </Redirect>
```

Example: Information on the redirection page

```
<?xml version="1.0" encoding="utf-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<Redirect>
<AccessProcedure>1.0</AccessProcedure>
<AccessLocation></AccessLocation>
<LocationName></LocationName>
<LoginURL>https://sip:9998/SubscriberPortal/
WisprLogin?nbiIP=<nbiIP>{& ... other
Redirection attributes in Table 11}</LoginURL>
<AbortLoginURL>https://sip:9998/SubscriberPortal
/AbortWisprLogin?nbiIP=<nbiIP></AbortLoginURL>
<MessageType>100</MessageType>
<ResponseCode>0</ResponseCode>
</Redirect>
</WISPAccessGatewayParam>
```

NOTE To do authentication. An HTTP POST request must be sent to the *<LoginURL>* with the `UserName` and `Password` fields.

NOTE The content type of request must be "application/x-www-form-urlencoded".

Example: Authentication Request (HTTP)

```
POST /SubscriberPortal/WisprLogin?nbiIP=<nbiIP>
HTTP/1.1
Host: sip:9998
Content-Type: application/x-www-form-urlencoded
UserName=<UserName>&Password=<Password>
```

Example: Authentication Reply

```
<?xml version="1.0" encoding="UTF-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<AuthenticationReply>
<MessageType>120</MessageType>
<ResponseCode>201</ResponseCode>
<ReplyMessage>Authentication pending</ReplyMessage>
<LoginResultsURL>https://sip:9998/SubscriberPortal
/WisprStatus?nbiIP=<nbiIP>&UserName=
<UserName>&Password=<Password></LoginResultsURL>
```

```
</AuthenticationReply>
</WISPAccessGatewayParam>
```

Example: Authentication Result (Login succeeded)

```
<?xml version="1.0"encoding="UTF-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<AuthenticationPollReply>
<MessageType>140</MessageType>
<ResponseCode>50</ResponseCode>
<ReplyMessage>Login succeeded</ReplyMessage>
<LogoffURL>https://sip:9998/SubscriberPortal
/WisprLogout?nbiIP=<nbiIP> &UserName=
<UserName>&Password=<Password></LogoffURL>
</AuthenticationPollReply>
</WISPAccessGatewayParam>
```

Example: Authentication Result (Login failed)

```
<?xml version="1.0" encoding="UTF-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<AuthenticationPollReply>
<MessageType>140</MessageType>
<ResponseCode>100</ResponseCode>
<ReplyMessage>Login failed</ReplyMessage>
</AuthenticationPollReply>
</WISPAccessGatewayParam>
```

Example: Logoff Reply

```
<?xml version="1.0" encoding="UTF-8"?>
<WISPAccessGatewayParam xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.acmewisp.com/WISPAccessGatewayParam.xsd">
<LogoffReply>
<MessageType>130</MessageType>
<ResponseCode>150</ResponseCode>
</LogoffReply>
</WISPAccessGatewayParam>
```

D

User Defined Interface - NBI and UDI

AP uses the control interface to communicate with the controller regarding its configuration. To have a logical separation of UE traffic from the AP control traffic the administrator can create an UDI (User Define Interface).

In case the UDI (using control interface, physical interface and hotspot service as shown in the figure) is configured the AP uses it to DNAT unauthorized UE's requests to the controller's captive portal (otherwise the AP uses the control interface).

NOTE UDI option is not available for vSZ-H.

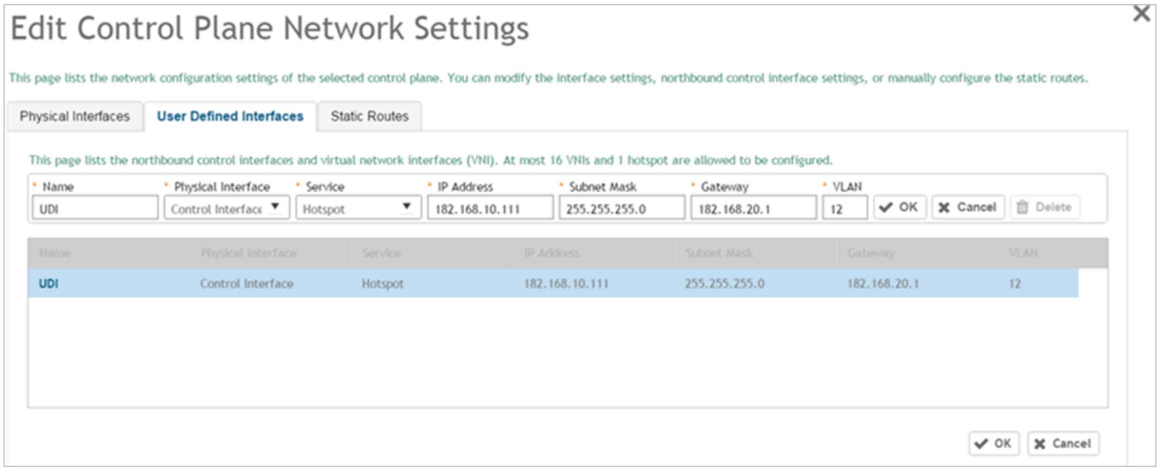
The controller's captive portal redirects the UE to the configured portal login page URL. When the UE triggers this portal URL request, the AP will let it go through (it will not DNAT to the controller's captive portal), as it is configured as ACL in AP, directly to the external portal server.

The external portal communicates with the controller's NBI for status/login/logout requests. The interfaces external portal can communicate are the interfaces NBI listens to. NBI is bound by default to the controller's control and management interfaces.

In addition, the administrator can configure UDI interface, which NBI will bind as well. This UDI for NBI can be the same UDI which AP DNAT to the controller's captive portal, or others using control or management physical interfaces and whatever service (Hotspot/not specified) as in the figure. To define UDI on the controller’s web interface navigate to **System > General Settings > Cluster Plane** > **> Select an existing Control Plane > Click on Configure > User Defined Interface.** . Click on **Add** to add and on **OK** to save the configuration details.

- Name of the UDI
- Physical Interface
- Service
- IP Address
- Subnet Mask
- Gateway
- VLAN

Figure 4: Configuring UDI



Edit Control Plane Network Settings

This page lists the network configuration settings of the selected control plane. You can modify the interface settings, northbound control interface settings, or manually configure the static routes.

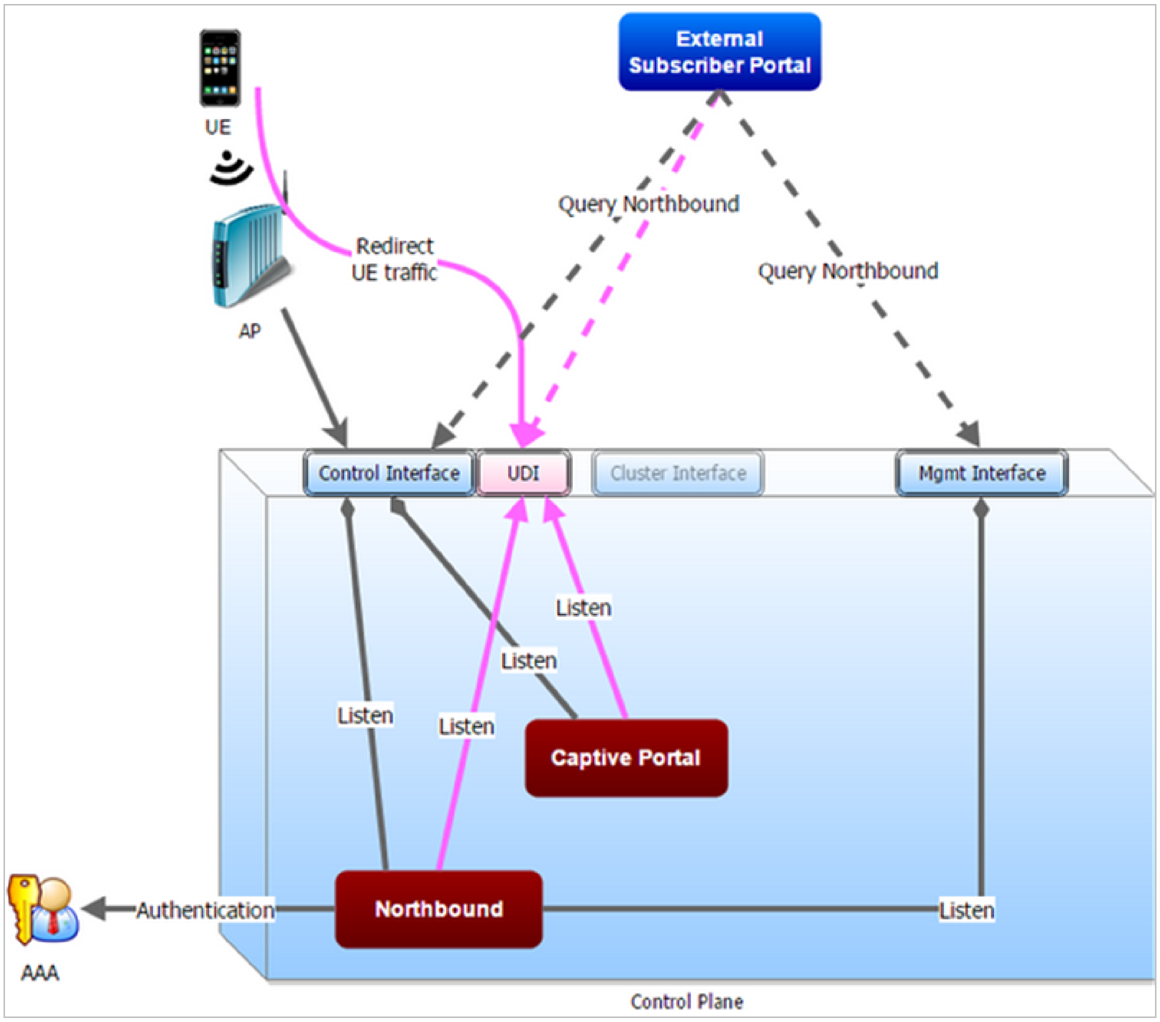
This page lists the northbound control interfaces and virtual network interfaces (VNI). At most 16 VNIs and 1 hotspot are allowed to be configured.

* Name	* Physical Interface	* Service	* IP Address	* Subnet Mask	* Gateway	* VLAN
UDI	Control Interface	Hotspot	182.168.10.111	255.255.255.0	182.168.20.1	12

Name	Physical Interface	Service	IP Address	Subnet Mask	Gateway	VLAN
UDI	Control Interface	Hotspot	182.168.10.111	255.255.255.0	182.168.20.1	12

The figure describes the request flows per interface.

Figure 5: Request flows per interface


External Subscriber Portal
UE
Redirect UE traffic
Query Northbound
Query Northbound
AP
Control Interface
UDI
Cluster Interface
Mgmt Interface
Listen
Listen
Listen
Listen
Captive Portal
Authentication
Northbound
Listen
AAA
Control Plane

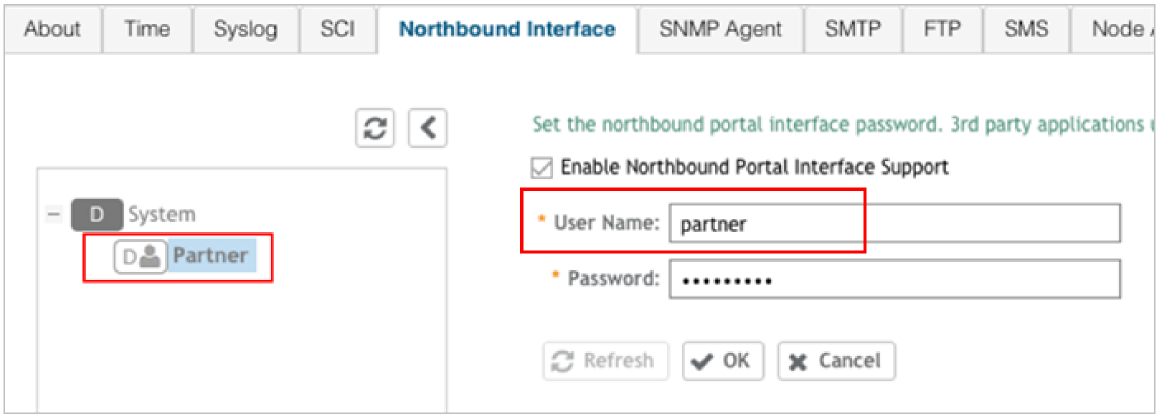
E

Northbound Portal Interface Support

This section explains Northbound Portal Interface (NBI) support for Managed Service Provider (MSP).

The **user name** is a mandatory field for MSP partner domain. It is used by partner users to query on Northbound Portal. A new *RequestUser* name field must be added to the JSON request coming from the partner user. **Question - how does one add a partner** Using this method, a partner user need not share the same NBI password with the system administrator.

Figure 6: Adding a Partner User Credentials



Set the northbound portal interface password. 3rd party applications

☑ Enable Northbound Portal Interface Support

* User Name: partner

* Password: •••••••••

For example:

```
{
Vendor: “ruckus”
RequestUserName: “partner”,
RequestPassword: “(PartnerPassword)”,
APIVersion: “1.0”,
RequestCategory: “UserOnlineControl”,
RequestType: “Login”,
UE-IP: “ENC12bc24c4777703327f2e0aabbf6b9f9e”,
UE-MAC: “ENCCDD319C6A476FA7127DF1FB80A63CD30ADC5E47C3DBE2157”,
UE-Proxy: “0”,
UE-Username: “test”,
UE-Password: “test”
}
```

F

WISPr Portal Details

The following are the WISPr portal details for GRE tunnels.

Non GRE Tunnel

The below table lists the WISPr details for non GRE tunnel.

Table 11: Non GRE tunnel

Non GRE Tunnel		IPv4	IPv6
Non WISPr Client	IPv4	Supported	Supported
	IPv6	Supported	Supported

Table 12: Non GRE tunnel and internal portal

Non GRE Tunnel		IPv4	IPv6
WISPr Client	IPv4	Supported	Supported
	IPv6	Not supported	Not supported

Table 13: Non GRE tunnel and external portal

Non GRE Tunnel		IPv4	IPv6
WISPr Client	IPv4	Supported	Supported (This portal is IPv4)
	IPv6	Not supported	Not supported

Ruckus GRE Tunnel

The below table lists the WISPr details for Ruckus GRE tunnel.

Table 14: Non GRE tunnel

Non GRE Tunnel		IPv4	IPv6
Non WISPr Client	IPv4	Supported	Not supported
	IPv6	Not supported	Not supported

Table 15: Non GRE tunnel and internal portal

Non GRE Tunnel		IPv4	IPv6
WISPr Client	IPv4	Supported	Not supported
	IPv6	Not supported	Not supported

Table 16: Non GRE tunnel and external portal

Non GRE Tunnel		IPv4	IPv6
WISPr Client	IPv4	Supported	Not supported
	IPv6	Not supported	Not supported

G

Certificate Warning

Certificate warning when end users are redirecting with HTTPS request.

When a CA-signed certificate is imported to SZ certificate store and applied to Hotspot (WISPr), SZ captive portal and internal portal page use the imported certificate. However, if an end user enters a HTTPS URL through the browser manually, one certificate warning message is still expected to be seen in the UE browser.

SZ captive portal need to complete the SSL handshake before sending 302 redirect response to UE. Since the FQDN(common name) in the certificate is impossible to match the URL that UE tries to visit, the browser will display a certificate warning.

To avoid certificate warning messages, major operating systems already have a built in some mechanisms to detect captive network and sending HTTP requests (not HTTPS), so that users can be redirected to a portal page automatically without any certificate error.

- Apple iOS CNA (captive network assistant) sends HTTP requests to some static URLs to detect captive portal.
- Android devices detected it by sending HTTP requests to http://clients3.google.com/generate_204.
- Window 7 sends HTTP requests to http://www.msftncsi.com/ncsi.txt to detect captive portal.

NOTE URL may vary based on different software releases.

In either case, user devices pop up a window and redirect users to the portal page with HTTP requests instead of HTTPs requests. No certificate warning will be shown if the UE is redirected automatically by the operating system.

Index

Ruckus®
Simply Better Wireless.

350 West Java Drive, Sunnyvale, CA

www.ruckuswireless.com