

radblue

RLT REST Interface Programming Guide

03 DEC 2013

Copyright © 2013 Radical Blue Gaming, Inc. All rights reserved.

All trademarks used within this document are the property of their respective owners. No part of this work may be reproduced in whole or in part, in any manner, without the prior written permission of Radical Blue Gaming, Inc.

Radical Blue Gaming, Inc.

85 Keystone Avenue Suite F
Reno, Nevada 89503

call us: +1.775.329.0990

visit us: www.radblue.com

drop us an email: sales@radblue.com

Need help?

At the RadBlue forum you can find the latest release information, report issues, get your questions answered, and submit suggestions for improving our products. Simply log on to:

<http://radblue.mywowbb.com>

Find out more about the GSA protocols

If you want to find out more about the Gaming Standards Association and the work being done in the area of protocol standardization for the gaming industry, we encourage you to visit their website at

www.gamingstandards.com.

RLT REST Interface Programming Guide	1
About RadBlue	3
Contents	5
Chapter 1: Introducing the RLT REST Interface	7
About the RLT REST Interface	7
Method Overview	7
Additional Resources	8
Chapter 2: Getting Started	9
Get Going with RLT REST	9
Use the RLT Scratch Pad for Reference	10
Set HTTP Field Headers	11
Test the Remote Interface URL	12
Chapter 3: XML Methods	13
Available XML Methods	13
Method: Start Engine	14
Request: EngineState	14
Response: EngineState	14
Examples	14
Method: Stop Engine	15
Request: EngineState	15
Response: EngineState	15
Examples	15
Method: Engine Status	16
Request: Not Applicable	16

Response: EngineState	16
Examples	16
Method: Start Tiger Scripts	17
Request: TigerScriptState	17
Response: TigerScriptState	17
Examples	17
Method: Stop Tiger Scripts	18
Request: TigerScriptState	18
Response: TigerScriptState	18
Examples	18
Method: Tiger Script Status	19
Request: Not Applicable	19
Response: TigerScriptState	19
Examples	19
Method: Reset Metrics	20
Request: MetricsState	20
Response: MetricsState	20
Examples	20
Chapter 4: Enumerations	21
EngineStatus	21
XML Data Types	21
Index	23

About the RLT REST Interface

The RLT REST interface allows you to control specific functions of RLT from a remote interface as well as request status information. The following functions can be executed through the remote interface:

- start and stop the RLT engine.
- start and stop all Tiger Scripts.
- request the status of the RLT engine.
- determine whether Tiger Scripts are running.
- reset RLT metrics back to zero (0).

You can use either Extensible Markup Language (XML) or JavaScript Object Notation (JSON) to remotely interact with RLT. Message formats for both protocols are detailed in this document.

The [Scratch Pad](#) webpage lets you see how a REST interface might work and assists you in developing your own interface. All commands available through the REST interface are available through Scratch Pad.

The REST Interface is included with the standard version of RLT version 34 or higher, and does not require a special license.

Method Overview

The following table summarizes the function of each REST method and its usage.

Method	Description
Engine Status	Requests the status of the RLT engine.
Reset Metrics	Resets the RLT metrics back to zero (0). Use this method when the host system is upgraded and you want to reset the metrics without restarting RLT.
Start Engine	Starts RLT engine.
Start Tiger Scripts	Starts all Tiger Scripts.
Stop Engine	Stops RLT engine.
Stop Tiger Scripts	Stops all Tiger Scripts.
Tiger Script Status	Determines, through a status request, whether Tiger Scripts are running or not.

Additional Resources

- [G2S Message Protocol](#)
- [RLT User Guide](#)
- [W3C XML Schema Definition Language \(XSD\) 1.1 Part 2: Datatypes](#)
- [JSON.org](#)

Get Going with RLT REST

Before you can begin to interface with RLT, you must write a program to remotely control RLT.

1. Use the [Scratch Pad](#), an RLT REST interface demonstration, to see how a REST interface might work and assist you in developing your own interface.

Troubleshooting Tip: To verify that the interface URL is working, try [testing the remote control URL](#).

2. Use the [method](#) information provided in this document to create your program. For a quick reference, the [Method Overview](#) provides a description of each available method and its use.

Note: Be sure to specify the [HTTP field headers](#) as needed.

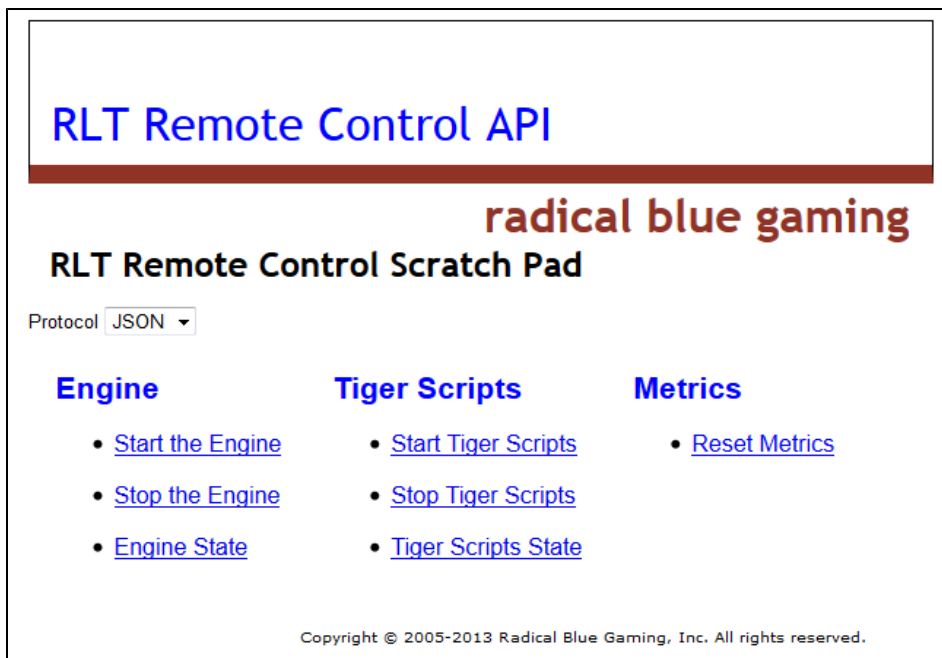
3. Connect to RLT through the following URL: **http://<IP address>:33501/<path>**
4. Use the interface using the **remote-control.txt** file ([**RLT installation directory**] > **logs**) for testing and troubleshooting. This file contains a log of all requests made by the client so you can verify that RLT processed the command properly.

Use the RLT Scratch Pad for Reference

The RLT Remote Control API Scratch Pad is a demonstration of what an RLT XML or RLT JSON interface might look like. It contains all of the supported methods. Once you've sent a method, the XML or JSON for both the request and response displays as well as a summary of the response.

To access the RLT Scratch Pad:

1. Open either a **Firefox** or **Safari** browser.
2. Paste the following HTTP location into the address window, and press **Enter**:
http://<ip-address>:33501/RLT/pages/ScratchPad-RLT.html/



3. Select the **Protocol** (XML or JSON) you are using. For this example, we chose **XML**.
4. Click any method. In this example, we chose **Start the Engine**. This method starts the RLT engine.

5. Click **Start the Engine**.



Once the message is sent, the **Remote Control Result** section displays a summary of the results along with the XML or JSON for the request and response messages.

Note: If you send multiple commands without exiting the screen, the summary response information (in pink and yellow) is added below the current summary response information, and the request/response information is updated for the last command sent.

6. Click **Return To Menu** to go back to the list of methods.

Set HTTP Field Headers

If you are implementing JSON for your REST interface, you must set the specified HTTP header fields to the following values. **If you do not specify the protocol in the HTTP field headers, RLT uses XML.**

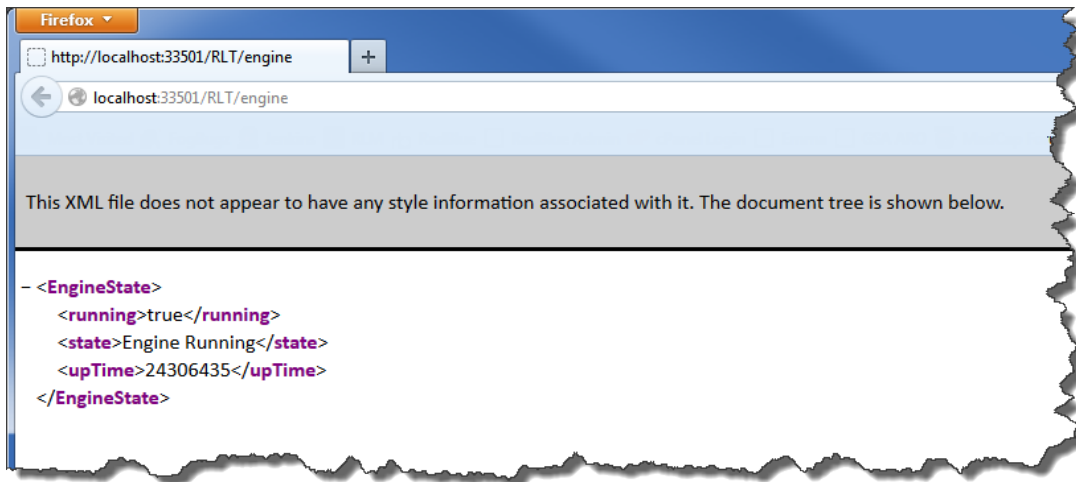
HTTP Field Name	Value	Description	Example
Content-Type	application/json	Content type of the client's request.	Content-Type:application/json
Accept	application/json	Content type that the client is willing to accept.	Accept:application/json

The **Accept** field must be present in both the HTTP GET and POST methods. The **Content-Type** field is only need for the POST method.

Test the Remote Interface URL

You can verify that the Remote Interface URL is working in RLT. Use this procedure if your program is not getting a response from RLT *or* if your program is getting an error from RLT.

1. Open **RLT**.
2. Configure **RLT** to communicate with either the RadBlue G2S Simulator (RGS) or the host system of your choice.
3. Start **RLT**.
4. Open a browser.
5. In the address bar, type **http://127.0.0.1:33501/RLT/engine**, and press **Enter**.



6. Verify that the URL is valid by checking that RLT returns content similar to the following:

```
<EngineState>
  <running>true</running>
  <state>Engine Running</state>
  <upTime>24306435</upTime>
</EngineState>
```

Available XML Methods

The following methods are available for the RLT REST interface:

- [Start Engine](#)
- [Stop Engine](#)
- [Engine Status](#)
- [Start Tiger Script](#)
- [Stop Tiger Script](#)
- [Tiger Script Status](#)
- [Reset Metrics](#)

Method: Start Engine

Using the Start Engine method, you can start the RLT engine.

To start the RLT engine through the user interface, go to RLT Controls on the main screen, and click **Start Engine**.

HTTP Type: POST

Call Type: Synchronous

Path: /RLT/engine

Request: EngineState

Element	Restrictions	Description
running	type: boolean	True starts the RLT engine.

Response: EngineState

Element	Restrictions	Description
running	type: boolean	True if RLT engine is running; False if RLT engine is not running. Depending on the number of EGMs being started, this value may be false because the RLT is still loading when the response is sent.
state	type: EngineStatus	State of RLT engine.
upTime	type: int	Number of milliseconds that the RLT engine (not the RLT <i>application</i>) has been running.

Examples

XML	JSON
This request starts the RLT engine.	
<pre><EngineState> <running>true</running> </EngineState></pre>	<pre>{"EngineState":{"running":true}}</pre>
This response provides the status of the RLT to the client.	
<pre><EngineState> <running>false</running> <state>Engine Loading</state> <upTime>0</upTime> </EngineState></pre>	<pre>{"EngineState":{"running":true,"state":"Engine Running","upTime":82761}}</pre>

Method: Stop Engine

Using the Stop Engine method, you can stop the RLT engine.

To stop the RLT engine through the user interface, go to RLT Controls on the main screen, and click **Stop Engine**.

HTTP Type: POST

Call Type: Synchronous

Path: /RLT/engine

Request: EngineState

Element	Restrictions	Description
running	type: boolean	False stops the RLT engine.

Response: EngineState

Element	Restrictions	Description
running	type: boolean	True if RLT engine is running: False if RLT engine is not running.
state	type: EngineStatus	State of RLT engine. When stopping the RLT engine, the state must <i>not</i> be Engine Loading .
upTime	type: int	Number of milliseconds that the RLT engine (not the RLT <i>application</i>) has been running.

Examples

XML	JSON
This request stops the RLT engine.	
<pre><EngineState> <running>false</running> </EngineState></pre>	<pre>{"EngineState":{"running":false}}</pre>
This response provides the status of a stopping RLT engine.	
<pre><EngineState> <running>false</running> <state>Engine Stopping</state> <upTime>500000</upTime> </EngineState></pre>	<pre>{"EngineState":{"running":false,"state":"Engine Stopping","upTime":0}}</pre>

Method: Engine Status

Using the Engine Status method, you can get the engine status of the RLT from the client.

HTTP Type: GET

Call Type: Synchronous

Path: /RLT/engine

Request: *Not Applicable*

There is no request information for this method.

Response: EngineState

Element	Restrictions	Description
running	type: boolean	True indicates the RLT engine is running; False indicates the RLT engine is not running.
state	type: EngineStatus	State of RLT engine.
upTime	type: int	Number of milliseconds that the RLT engine (not the RLT <i>application</i>) has been running.

Examples

XML	JSON
This response provides the status of the RLT engine.	
<pre><EngineState> <running>true</running> <state>Engine Running</state> <upTime>64132</upTime> </EngineState></pre>	<pre>{"EngineState":{"running":true,"state":"Engine Running","upTime":17465}}</pre>

Method: Start Tiger Scripts

Using the Start Tiger Scripts method, you can start all Tiger Scripts associated with the simulated EGMs.

To start Tiger Scripts through the user interface, go to RLT Controls on the main screen, and click **Start All Tiger Scripts**.

HTTP Type: POST

Call Type: Synchronous

Path: /RLT/engine/tigerscripts

Request: TigerScriptState

Element	Restrictions	Description
running	type: boolean	True starts all Tiger Scripts.

Response: TigerScriptState

Element	Restrictions	Description
running	type: boolean	True indicates Tiger Scripts are running; False indicates Tiger Scripts are not running.
upTime	type: int	Number of milliseconds that the Tiger Scripts have been running.

Examples

XML	JSON
This request starts the Tiger Scripts.	
<pre><TigerScriptState> <running>true</running> </TigerScriptState></pre>	<pre>{"TigerScriptState":{"running":true}}</pre>
This response provides the status of the request to start all Tiger Scripts.	
<pre><TigerScriptState> <running>true</running> <upTime>0</upTime> </TigerScriptState></pre>	<pre>{"TigerScriptState":{"running":true,"upTime":0}}</pre>

Method: Stop Tiger Scripts

Using the Stop Tiger Scripts method, you can stop all Tiger Scripts.

To stop all Tiger Scripts through the user interface, go to RLT Controls on the main screen, and click **Stop All Tiger Scripts**.

HTTP Type: POST

Call Type: Synchronous

Path: /RLT/engine/tigerscripts

Request: TigerScriptState

Element	Restrictions	Description
running	type: boolean	False stops all Tiger Scripts.

Response: TigerScriptState

Element	Restrictions	Description
running	type: boolean	True indicates Tiger Scripts are running; False indicates Tiger Scripts are not running.
upTime	type: int	Number of milliseconds that the Tiger Scripts have been running.

Examples

XML	JSON
This request stops all Tiger Scripts.	
<pre><TigerScriptState> <running>false</running> </TigerScriptState></pre>	<pre>{"TigerScriptState":{"running":false}}</pre>
This response provides the status of the request to stop all Tiger Scripts.	
<pre><TigerScriptState> <running>false</running> <upTime>500000</upTime> </TigerScriptState></pre>	<pre>{"TigerScriptState":{"running":false,"upTime":0}}</pre>

Method: Tiger Script Status

Using the Tiger Script Status method, you can get the status of the Tiger Scripts.

There is no corresponding feature for this method in the RLT user interface.

HTTP Type: GET

Call Type: Synchronous

Path: /RLT/engine/tigerscripts

Request: *Not Applicable*

There is no request information for this method.

Response: TigerScriptState

Element	Restrictions	Description
running	type: boolean	True indicates Tiger Scripts are running; False indicates Tiger Scripts are not running.
upTime	type: int	Number of milliseconds that the RLT engine (not the RLT <i>application</i>) has been running.

Examples

XML	JSON
This response provides the status of all Tiger Scripts.	
<pre><TigerScriptState> <running>true</running> <upTime>500000</upTime> </TigerScriptState></pre>	<pre>{"TigerScriptState":{"running":false,"upTime":0}}</pre>

Method: Reset Metrics

Using the Reset Metrics method, you can reset RLT metrics. This method useful when the host system is upgraded and you want to reset the metrics without restarting RLT.

To reset RLT metrics through the user interface, click Metrics under the Load Tester header on the left side of the screen, and click **Reset Metrics**.

HTTP Type: POST

Call Type: Synchronous

Path: /RLT/engine/metrics

Request: MetricsState

Element	Restrictions	Description
reset	type: boolean	True resets RLT metrics..

Response: MetricsState

Element	Restrictions	Description
reset	type: boolean	True if metrics were reset.

Examples

XML	JSON
This request resets RLT metrics.	
<pre><MetricsState> <reset>true</reset> </MetricsState></pre>	<pre>{"MetricsState":{"reset":true}}</pre>

EngineStatus

Enumeration	Description
Engine Running	Indicates that RLT engine is running.
Engine Loading	Indicates that RLT engine is loading.
Engine Starting	Indicates that RLT engine is starting.
Engine Stopped	Indicates that RLT engine is stopped.
Engine Stopping	Indicates that a Stop Engine request was sent, but the request has not been completed.
Scripts Running	Indicates that the Tiger Scripts are running.
Scripts Starting	Indicates that the Tiger Scripts are starting.
Scripts Stopping	Indicates that the Tiger Scripts are stopping.
Status Unknown	Indicates that the status of the RLT engine is not known.

XML Data Types

RadBlue uses standard XML data types as defined by the [W3C XML Schema Definition Language \(XSD\) 1.1 Part 2: Datatypes](#) section of version 1.1 of the XML schema.

- boolean
- int
- string

A

about rlt rest 7
additional resources 8

B

boolean 21

C

connect to url 12

D

data types 21

 boolean 21

 string 21

date types

 int 21

E

engine loading 21

engine running 21

engine starting 21

engine status 16

engine stopped 21

engine stopping 21

engineStatus 21

enumeration

 engineStatus 21

G

g2s message protocol 8

get started 9

I

int 21

interface example 10

J

json.org 8

L

list of methods 13

M

method overview 7

methods

 available methods 13

 engine status 16

 reset metrics 20

 start engine 14

 start tiger scripts 17

 stop engine 15

 stop tiger scripts 18

 tiger script status 19

metrics

 reset 20

R

remote control api 10
remote interface url 12
reset 20
reset metrics 20
rlt rest
 about 7
 get started 9
 method overview 7
 sample interface 10
rlt user guide 8
running 14-19

S

sample interface 10
scratch pad 10
scripts running 21
scripts starting 21
scripts stopping 21
start engine 14
start tiger scripts 17
state 14-16
status unknown 21
stop engine 15
stop tiger scripts 18
string 21

T

test browser connection 12
tiger script status 19

U

uptime 14-19

W

w2c xml schema definition language 8
w3c xml schema definition language 21

X

xml data types 21
xml schema definition language 21
xsd 21