

REST / HTTP

Establishing connection

When a connection is being established the following REST requests are used:

connect - main connection establishing method

ConnectionStatusEvent - connection status

RegistrationStatusEvent - SIP registration status if SIP registration is required

The image shows a Wireshark packet capture of a REST/HTTP connection between 127.0.0.1 and 127.0.0.1. The filter is 'ip.src==127.0.0.1 && tcp.dstport==9091'. The capture shows a sequence of packets: a POST request to /EchoApp/connect, several TCP ACKs, a POST request to /EchoApp/ConnectionStatusEvent, more TCP ACKs, and a final POST request to /EchoApp/RegistrationStatusEvent. The packet details pane shows the structure of the HTTP frames, including the Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol layers.

No.	Time	Source	Destination	Protocol	Length	Info
15	3.620726	127.0.0.1	127.0.0.1	HTTP	725	POST /EchoApp/connect HTTP/1.1 (application/json)
17	3.627282	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=658 Ack=385 win=354 Len=0 TSval=2241866196 TSecr=2241866196
19	3.627307	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=658 Ack=390 win=354 Len=0 TSval=2241866196 TSecr=2241866196
21	3.627384	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=658 Ack=775 win=365 Len=0 TSval=2241866196 TSecr=2241866196
23	3.627457	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=658 Ack=777 win=365 Len=0 TSval=2241866196 TSecr=2241866196
25	3.628183	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=658 Ack=782 win=365 Len=0 TSval=2241866197 TSecr=2241866197
39	3.670936	127.0.0.1	127.0.0.1	HTTP	760	POST /EchoApp/ConnectionStatusEvent HTTP/1.1 (application/json)
41	3.676215	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=1350 Ack=1166 win=376 Len=0 TSval=2241866245 TSecr=2241866245
43	3.676283	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=1350 Ack=1171 win=376 Len=0 TSval=2241866245 TSecr=2241866245
45	3.676344	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=1350 Ack=1577 win=384 Len=0 TSval=2241866245 TSecr=2241866245
47	3.676409	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=1350 Ack=1579 win=384 Len=0 TSval=2241866245 TSecr=2241866245
49	3.676996	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=1350 Ack=1584 win=384 Len=0 TSval=2241866246 TSecr=2241866246
66	3.762686	127.0.0.1	127.0.0.1	HTTP	1066	POST /EchoApp/RegistrationStatusEvent HTTP/1.1 (application/json)
68	3.766600	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=2348 Ack=1968 win=384 Len=0 TSval=2241866335 TSecr=2241866335
70	3.766654	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=2348 Ack=1973 win=384 Len=0 TSval=2241866336 TSecr=2241866336
72	3.766719	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=2348 Ack=2683 win=384 Len=0 TSval=2241866336 TSecr=2241866336
74	3.766851	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=2348 Ack=2685 win=384 Len=0 TSval=2241866336 TSecr=2241866336
76	3.767358	127.0.0.1	127.0.0.1	TCP	68	33642->9091 [ACK] Seq=2348 Ack=2690 win=384 Len=0 TSval=2241866336 TSecr=2241866336

Outgoing call

Outgoing call uses the following REST requests:

call - outgoing SIP call

CallStatusEvent - update the outgoing call status

hangup - hangup the call

log.pcap [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `ip.src==127.0.0.1 && tcp.dstport==9091` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
114	8.571338	127.0.0.1	127.0.0.1	HTTP	1889	POST /EchoApp/call HTTP/1.1 (application/json)
119	8.577284	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=1822 Ack=385 win=384 Len=0 TSval=23272
120	8.577292	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=1822 Ack=390 win=384 Len=0 TSval=23272
121	8.577298	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=1822 Ack=1941 win=384 Len=0 TSval=23272
123	8.577363	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=1822 Ack=1943 win=384 Len=0 TSval=23272
125	8.577961	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=1822 Ack=1948 win=384 Len=0 TSval=23272
145	8.682350	127.0.0.1	127.0.0.1	HTTP	992	POST /EchoApp/callStatusEvent HTTP/1.1 (application/json)
147	8.687835	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=2746 Ack=2332 win=384 Len=0 TSval=23272
149	8.687856	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=2746 Ack=2337 win=384 Len=0 TSval=23272
151	8.687927	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=2746 Ack=2981 win=384 Len=0 TSval=23272
153	8.687993	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=2746 Ack=2983 win=384 Len=0 TSval=23272
155	8.688554	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=2746 Ack=2988 win=384 Len=0 TSval=23272
176	8.754441	127.0.0.1	127.0.0.1	HTTP	992	POST /EchoApp/callStatusEvent HTTP/1.1 (application/json)
179	8.758047	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=3670 Ack=3372 win=384 Len=0 TSval=23272
181	8.758053	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=3670 Ack=3377 win=384 Len=0 TSval=23272
182	8.758058	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=3670 Ack=4021 win=384 Len=0 TSval=23272
185	8.758517	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=3670 Ack=4028 win=384 Len=0 TSval=23272
210	8.987719	127.0.0.1	127.0.0.1	HTTP	2022	POST /EchoApp/callStatusEvent HTTP/1.1 (application/json)
212	8.992842	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=5624 Ack=4412 win=384 Len=0 TSval=23272
214	8.992901	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=5624 Ack=4417 win=384 Len=0 TSval=23272
216	8.992992	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=5624 Ack=6090 win=384 Len=0 TSval=23272
218	8.993050	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=5624 Ack=6092 win=384 Len=0 TSval=23272
220	8.993508	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=5624 Ack=6097 win=384 Len=0 TSval=23272
1698	16.309715	127.0.0.1	127.0.0.1	HTTP	463	POST /EchoApp/hangup HTTP/1.1 (application/json)
1703	16.315267	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6019 Ack=6481 win=384 Len=0 TSval=23272
1704	16.315282	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6019 Ack=6485 win=384 Len=0 TSval=23272
1706	16.315347	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6019 Ack=6609 win=384 Len=0 TSval=23272
1708	16.315405	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6019 Ack=6611 win=384 Len=0 TSval=23272
1710	16.315947	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6019 Ack=6616 win=384 Len=0 TSval=23272
1734	16.353704	127.0.0.1	127.0.0.1	HTTP	675	POST /EchoApp/callStatusEvent HTTP/1.1 (application/json)
1739	16.357702	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6626 Ack=7334 win=384 Len=0 TSval=23272
1754	16.367611	127.0.0.1	127.0.0.1	TCP	68	34266-9091 [ACK] Seq=6626 Ack=7338 win=384 Len=0 TSval=23272

File: "C:\tmp\6\log.pcap" 326 kB 00:00:17 Pac... Profile: Default

Sending and receiving SIP messages

When sending and receiving messages the following REST requests are used:

sendMessage - send a message

MessageStatusEvent - update the message status

OnMessageEvent - receive the incoming message

log.pcap [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `ip.src==127.0.0.1 && tcp.dstport==9091` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
33	10.169410	127.0.0.1	127.0.0.1	TCP	68	34248-9091 [FIN, ACK] Seq=1 Ack=1 win=384 Len=0 TSval=2326163678 TSecr=2325203574
35	10.169710	127.0.0.1	127.0.0.1	TCP	76	34257-9091 [SYN] Seq=0 win=32792 Len=0 MSS=16396 SACK_PERM=1 TSval=2326163678 TSecr=2326163678
37	10.169725	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=1 Ack=1 win=32896 Len=0 TSval=2326163678 TSecr=2326163678
38	10.169863	127.0.0.1	127.0.0.1	HTTP	536	POST /EchoApp/sendMessage HTTP/1.1 (application/json)
41	10.175721	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=469 Ack=385 win=33920 Len=0 TSval=2326163685 TSecr=2326163684
43	10.175798	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=469 Ack=389 win=33920 Len=0 TSval=2326163685 TSecr=2326163685
45	10.175912	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=469 Ack=581 win=34944 Len=0 TSval=2326163685 TSecr=2326163685
47	10.175981	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=469 Ack=583 win=34944 Len=0 TSval=2326163685 TSecr=2326163685
49	10.176571	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=469 Ack=588 win=34944 Len=0 TSval=2326163685 TSecr=2326163685
63	10.209948	127.0.0.1	127.0.0.1	HTTP	1193	POST /EchoApp/MessageStatusEvent HTTP/1.1 (application/json)
68	10.214131	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=1594 Ack=1819 win=37760 Len=0 TSval=2326163723 TSecr=23261637
70	10.214612	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=1594 Ack=1826 win=37760 Len=0 TSval=2326163724 TSecr=23261637
83	10.270728	127.0.0.1	127.0.0.1	HTTP	1282	POST /EchoApp/OnMessageEvent HTTP/1.1 (application/json)
87	10.274535	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=2808 Ack=2215 win=39424 Len=0 TSval=2326163783 TSecr=23261637
89	10.274577	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=2808 Ack=3152 win=41344 Len=0 TSval=2326163783 TSecr=23261637
99	10.314686	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=2808 Ack=3157 win=41344 Len=0 TSval=2326163824 TSecr=23261637
108	10.445610	127.0.0.1	127.0.0.1	HTTP	779	POST /EchoApp/MessageStatusEvent HTTP/1.1 (application/json)
110	10.448970	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=3519 Ack=3541 win=43136 Len=0 TSval=2326163958 TSecr=23261639
112	10.449025	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=3519 Ack=3546 win=43136 Len=0 TSval=2326163958 TSecr=23261639
114	10.449087	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=3519 Ack=3974 win=45056 Len=0 TSval=2326163958 TSecr=23261639
116	10.449147	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=3519 Ack=3976 win=45056 Len=0 TSval=2326163958 TSecr=23261639
118	10.449670	127.0.0.1	127.0.0.1	TCP	68	34257-9091 [ACK] Seq=3519 Ack=3981 win=45056 Len=0 TSval=2326163959 TSecr=23261639

File: "C:\tmp\5\log.pcap" 67 kB 00:00:13 Packets: 138 · Displayed: 22 (15,9%) · Load ti... Profile: Default

Possible problems

Problems with REST traffic not flowing can be caused by various web server integration issues. For example, if the traffic does not go when the 'connect' REST method is invoked, connection cannot be successfully established. If `ConnectionStatusEvent` or `RegistrationStatusEvent` do not invoke, the web server cannot receive connection status and SIP registration status notifications. REST error handling is thoroughly described in the REST hooks section.

Troubleshooting

Make sure your web server correctly responds to REST / HTTP requests and the port of your web server is open in Firewall. Make sure the WCS server app is correctly configured for the REST URL of the web server, and [server logs](#) contains no errors. Besides, you can turn off using REST completely with the [remove app-rest-method](#) command. In this case there will be no requests to the web server and corresponding errors. And the web client will have to pass all necessary data to itself as described in the REST hooks section.