

使用日誌的Finesse代理登入跟蹤

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[啟動Agent Desktop](#)

[代理登入憑據](#)

[系統資訊](#)

[API REQUEST](#)

[建立BOSH連線](#)

[代理登入](#)

[執行登入](#)

[註銷代碼、原因代碼、電話簿](#)

簡介

本文檔介紹通過Finesse系統使用日誌檔案登入代理所涉及的過程。瞭解不同Finesse元件、電腦電話整合(CTI)伺服器 and 客戶端案頭之間的消息流非常重要，這樣您才能成功排除故障。

必要條件

需求

思科建議您瞭解Cisco Finesse和語音作業系統(VOS)CLI命令提示資訊。

採用元件

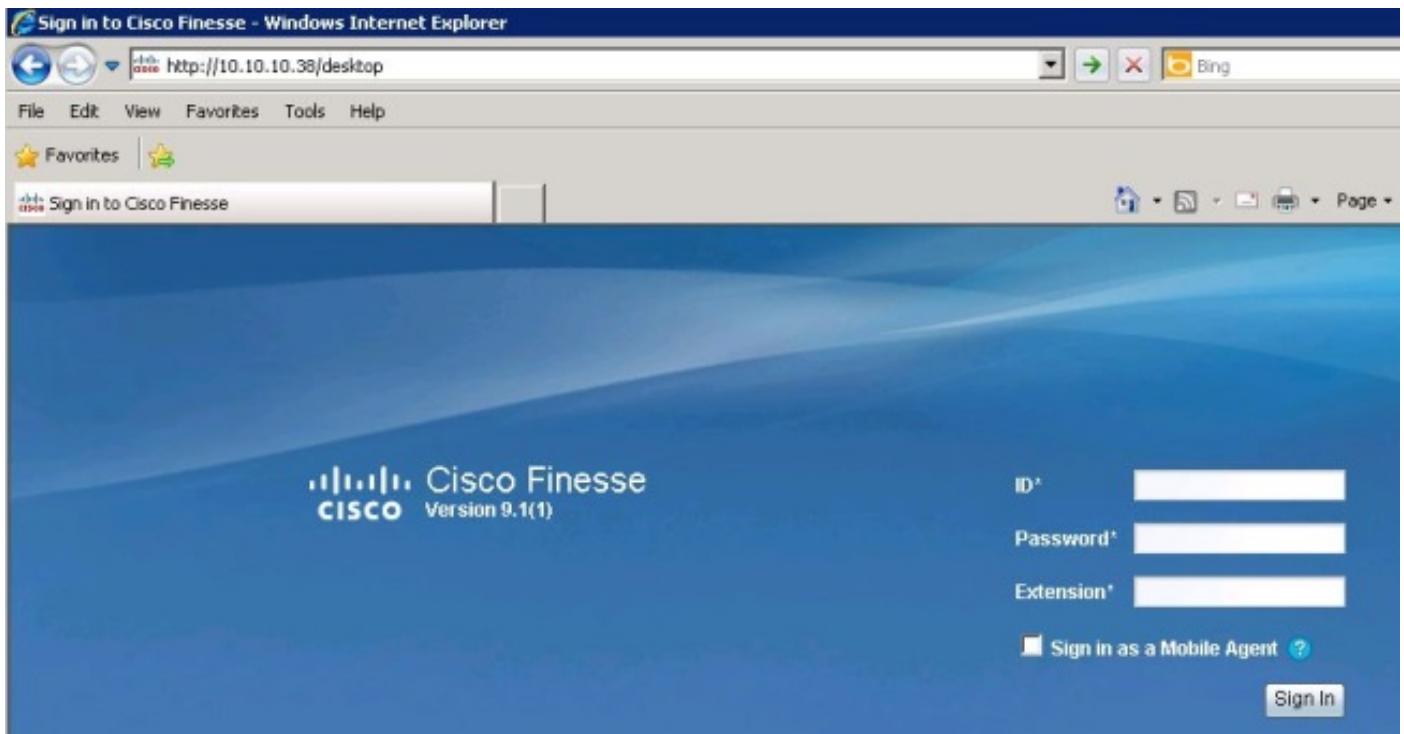
本檔案中的資訊是根據Cisco Finesse版本9.1(1)。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

啟動Agent Desktop

若要啟動Agent Desktop，請將此URL複製到Web瀏覽器中：<http://<your finesse server>/desktop>。

在Finesse版本9.1中，支援HTTP或HTTPS。



Finesse使用Tomcat作為Web伺服器。當您啟動Web瀏覽器時，會向Finesse提出請求，以向您顯示代理案頭。Cisco Tomcat `localhose_access_log`命令顯示載入Agent Desktop的請求。

```
10.10.10.211 10.10.10.211 - - 80 GET / HTTP/1.1 302 - 141
10.10.10.211 10.10.10.211 - - 80 GET /desktop/container/ HTTP/1.1 200 4541 185
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/css/base.css
HTTP/1.1 200 3093 7
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/css/login.css
HTTP/1.1 200 2185 0
10.10.10.211 10.10.10.211 - - 80 GET /desktop/js/Logon.js HTTP/1.1 200 1745 1
10.10.10.211 10.10.10.211 - - 80 GET /desktop/js/utilities/Cookies.js HTTP/1.1
200 2390 0
10.10.10.211 10.10.10.211 - - 80 GET /desktop/thirdparty/jquery/js/jquery.tools.
min.js HTTP/1.1 200 15699 1
10.10.10.211 10.10.10.211 - - 80 GET /desktop/thirdparty/jquery/js/jquery-1.5.
min.js HTTP/1.1 200 84523 7
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/images/sprite_
buttons.png HTTP/1.1 200 3297 0
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/images/help.png
HTTP/1.1 200 830 0
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/images/cisco_logo.
png HTTP/1.1 200 760 0 200 2205 1
10.10.10.211 10.10.10.211 - - 80 GET /desktop/theme/finesse/images/bg.jpg
HTTP/ 1.1 200 32222 4
```

代理登入憑據

現在已顯示Agent Desktop，請輸入您的登入憑據。在Finesse向CTI伺服器傳送登入請求之前，客戶端需要建立通過同步HTTP(BOSH)雙向流連線。為了建立BOSH連線，客戶端首先從Finesse伺服器請求系統資訊。

系統資訊

客戶端案頭向此URL發出表示狀態傳輸(REST)應用程式程式設計介面(API)請求：
:/finesse/api/SystemInfo。請注意nocache=。此唯一ID用於通過系統跟蹤此請求。返回的
status=200表示已成功收到請求。

```
Container : [ClientServices] SystemInfo: requestId='undefined', Making REST  
request: method=GET, url='/finesse/api/SystemInfo?nocache=1366756802163'18:40:03:  
Container : [ClientServices] SystemInfo: requestId='undefined', Returned  
with status=200
```

如果您沒有客戶端日誌，但需要跟蹤請求，則可以搜尋Tomcat localhost_access_log，以確定何時發出REST API請求，並找到唯一識別符號。

```
127.0.0.1 127.0.0.1 - - 80 GET /finesse/api/SystemInfo ?nocache=1366756802163  
HTTP/1.1 200 336 120 10.10.10.211 10.10.10.211 2001 - 80 GET /gadgets/makeRequest  
?refresh=3600&url=http%3A%2F%2Flocalhost%2Ffinesse%2Fapi%2FSystemInfo%3Fnocache%  
3D1366756802163&httpMethod=GET&headers=Authorization%3DBasic%2520MjAwMToyMDAx%  
26locale%3Den_US&postData=&authz=&st=&contentType=TEXT&numEntries=3&getSummaries  
=false&signOwner=true&signViewer=true&gadget=undefined&container=default&  
bypassSpecCache=&getFullHeaders=false HTTP/1.1 200 659 596
```

API_REQUEST

Tomcat將此API請求傳送到Finesse REST API Web應用程式儲存庫(WAR)。要查詢Finesse REST API日誌，請通過時間戳或nocache ID搜尋Finesse Webservices日誌以查詢API_REQUEST。此日誌顯示REQUEST_START、REQUEST_URL、REQUEST_END以及系統完成請求所用的elapsed_time。

```
%CCBU_http-8080-7-6-REQUEST_START: %[method_name=GET][parameter_name=  
{ nocache=[1366756802163], }][resource_name=/SystemInfo][usr=]: Request start  
%CCBU_http-8080-7-6-API_REQUEST: %[REQUEST_URL=SystemInfo][agent_id=][request_  
identifier=][request_method=systemInfo.GET][request_parameters=]: Request from  
client to webservice api  
%CCBU_http-8080-7-6-REGISTER_API_STATS_OBJECT: %[resource_name=com.cisco.ccbu:  
category=WebAppStats,component0=SystemInfo-GET]: Registered new api stats object  
for new request type. %CCBU_http-8080-7-6-REQUEST_END: %[elapsed_time=98]: Request complete  
此處顯示了由REST API請求返回給客戶端以檢索系統資訊的內容。此資訊位於客戶端 (代理) 日誌  
中。
```

```
content='<SystemInfo>  
<primaryNode>  
<host>UCCEFINESSE91.vmload.cvp</host>  
</primaryNode>  
<secondaryNode>  
<host>UCCEFINESSE138.vmload.cvp</host>  
</secondaryNode>  
<status>IN_SERVICE</status>  
<xmppDomain>UCCEFINESSE138.vmload.cvp</xmppDomain>  
<xmppPubSubDomain>pubsub.UCCEFINESSE138.vmload.cvp</xmppPubSubDomain>  
</SystemInfo>'
```

建立BOSH連線

SystemInfo顯示主Finesse伺服器 and 輔助Finesse伺服器、Finesse的狀態為IN_SERVICE、xmppDomain和xmppPubSubDomain。客戶端現在有足夠的資訊來建立BOSH連線。

```
18:40:03: Container : PageServices.init().onLoad: System info status: IN_SERVICE
18:40:03: Container : PageServices.init(): Establishing BOSH connection...
18:40:03: Container : PageServices.init(): Starting timeout and poller...
18:40:04: Container : [ClientServices] MasterPublisher._connInfoHandler()
- Connection status: connecting
18:40:04: Container : [ClientServices] MasterPublisher._connInfoHandler()
- Connection status: disconnected
18:40:04: Container : PageServices._onDisconnect(): retryCount=0, retrying...
18:40:04: Container : [ClientServices] MasterPublisher._connInfoHandler()
- Connection status: connecting
18:40:05: Container : [ClientServices] MasterPublisher._connInfoHandler()
- Connection status: connected
18:40:05: Container : PageServices.onLoad(): BOSH established!
```

在BOSH連線建立後，客戶端成功訂閱了Finesse對象 (節點) /finesse/api/User/2001。

當客戶端的BOSH連線建立後，Web服務日誌會收到來自客戶端的PRESENCE_NOTIFICATION消息。此PRESENCE_TYPE僅表示客戶端可用於接收XMPP事件，且與Unified Contact Center Enterprise(UCCE)中的代理可用性無關。請記住，座席尚未登入。

附註：僅當客戶端建立BOSH連線或客戶端的BOSH連線斷開時，才會看到PRESENCE_TYPE消息。當客戶端的BOSH連線斷開時，PRESENCE_TYPE顯示為不可用。

以下是Web服務日誌中的通知事件：

```
%CCBU_Smack Listener Processor (1)-6-PRESENCE_NOTIFICATION_RECIEVED:
%[FROM JID=2001@uccefinesse138.vmlod.cvp/desktop]
[PRESENCE_TYPE=available]: Finesse received a presence notification
```

代理登入

現在，客戶端已建立BOSH連線，登入過程開始。客戶端發出另一個REST API請求以獲取當前使用者資訊。若要提出此要求，請導覽至此URL:/finesse/api/User/2001和enter method=GET。

由於這是不同的API請求，因此nocache ID不同。因此，為了跟蹤此請求，您需要使用此新ID。

```
Container : PageServices.onLoad(): BOSH established! Commencing sign-in process
Container : [ClientServices] User: requestId='undefined', Making REST request:
method=GET, url='/finesse/api/User/2001?nocache=1366756805180
'18:40:05: Container : [ClientServices] User: requestId='undefined',
Returned with status=200,
```

如果需要，您可以在Tomcat localhost_access_log中找到此請求。以下是在Web服務日誌中查詢它的方式：

```
%CCBU_http-8080-7-6-REQUEST_START: %[method_name=GET][parameter_name={ nocache=1366756805180, }][resource_name=/User/2001][usr=2001]: Request start
```

```
%CCBU_http-8080-7-6-API_REQUEST: %[REQUEST_URL=User/2001][agent_id=2001]
[request_identifier=null][request_method=user.GET][request_parameters=2001]:
Request from client to webservice api
```

以下是Notification Services日誌中的請求。注意HTTP/1.1 200 ok。

附註：思科通知日誌僅供參考。如果啟用Cisco Finesse通知日誌記錄，則會影響效能。

```
>> "GET /finesse/api/User/2001 HTTP/1.1[\r][\n]"
Adding Host request header
>>"Authorization: Basic MjAwMToyMDAx[\r][\n]"
>>"User-Agent: Jakarta Commons-HttpClient/3.1[\r][\n]"
>>"Host: localhost:8080[\r][\n]"
>> " [\r][\n]"
<<"HTTP/1.1 200 OK[\r][\n]"
<<"HTTP/1.1 200 OK[\r][\n]"
<<"Pragma: No-cache[\r][\n]"
<<"Cache-Control: no-cache[\r][\n]"
```

既然通知服務已收到該請求，它就會發佈此使用者的資訊。下面是傳送到客戶端的通知服務日誌中的POST：

```
Cookie accepted: "$Version=0; JSESSIONIDSSO=C11F62C59D0D0438CDEDEEB0DB12AA0B;
$Path=/"
Cookie accepted: "$Version=0; JSESSIONID=25FE81BD7DB73280A07B4CA4138E7680;
$Path=/finesse"
Buffering response body
<<"<User>[\n]"
<<" <dialogs>/finesse/api/User/2001/Dialogs</dialogs>[\n]"
<<" <extension></extension>[\n]"
<<" <firstName>Mickey</firstName>[\n]"
<<" <lastName>Mouse</lastName>[\n]"
<<" <loginId>2001</loginId>[\n]"
<<" <loginName>mmouse</loginName>[\n]"
<<" <roles>[\n]"
<<" <role>Agent</role>[\n]"
<<" </roles>[\n]"
<<" <state>LOGOUT</state>[\n]"
<<" <stateChangeTime></stateChangeTime>[\n]"
<<" <teamId>5000</teamId>[\n]"
<<" <teamName>Minnies_Team</teamName>[\n]"
<<" <uri>/finesse/api/User/2001</uri>[\n]"
<<"</User>"
```

此XMPP事件(在本例中為代理2001)將傳送到所有訂閱客戶端。客戶端的JavaScript接收XMPP事件，並將該事件傳送到客戶端內的小工具。以下是顯示響應內容的客戶端日誌：

```
Commencing sign-in process18:40:05: Container : [ClientServices] User: requestId=
'undefined', Maulr='/finesse/api/User/2001?nocache=1366756805180'18:40:05:
Container : [ClientServices] User: requestId='undefined', Returned with status=200,
content='<User> king REST request: method=GET,
<dialogs>/finesse/api/User/2001/Dialogs</dialogs>
<extension></extension>
<firstName>Mickey</firstName>
<lastName>Mouse</lastName>
<loginId>2001</loginId>
<loginName>mmouse</loginName>
<roles>
<role>Agent</role>
```

```
</roles>
<state>LOGOUT</state>
<stateChangeTime></stateChangeTime>
<teamId>5000</teamId>
<teamName>Minnies_Team</teamName>
<uri>/finesse/api/User/2001</uri>
</User>
```

執行登入

現在，客戶端已準備好執行登入。請注意RequestID。在請求正文中傳送RequestID。使用此RequestID可以按照登入請求返回REST API > CTI > REST API > Notification Service > Response 回客戶端。此請求是PUT，這意味著客戶端正在請求更新或更改其當前狀態。

```
Container : SignIn.handleUserLoad(): Performing login: extn=2003 18:40:05:
Container : [ClientServices] User: requestId='6e210ca9-5786-43bc-babf-64a397a6057f',
</data>
<event>PUT</event>
<requestId>6e210ca9-5786-43bc-babf-64a397a6057f</requestId>
<source>/finesse/api/User/2001</source>
</Update>
```

Finesse REST API從客戶端接收此請求。然後，API向CTI伺服器傳送SetAgentStateReq。

```
%CCBU_http-8080-7-6-API_REQUEST: %[REQUEST_URL=User/2001][agent_id=2001]
[request_identifer=6e210ca9-5786-43bc-babf-64a397a6057f][request_method=
user.PUT][request_parameters= extension:2003 state:LOGIN]: Request from
client to webservice api
%CCBU_http-8080-7-6-REGISTER_API_STATS_OBJECT: %[resource_name=com.cisco.ccbu:
category=WebAppStats,component0=User-[id]-PUT]: Registered new api stats object
for new request type.
%CCBU_http-8080-7-6-REQUEST_END: %[elapsed_time=8]: Request complete
%CCBU_pool-5-thread-4-6-MESSAGE_TO_CTI_SERVER: %[cti_message=Invoke id :20 ,
agentstate : 0, workmode : 0, reason code: -15532, forceflag :1, agentcapacity:
0, agentext: 2003, agentid: 2001][cti_message_name=SetAgentStateReq]:
Message going to the backend cti server
```

CTI伺服器收到請求。

```
Trace: AGENT_EVENT: ID=2001 Periph=5001 Ext=2003 Inst=2003 Sig=
Trace: SkgState=LOGIN SkgDuration=0 OverallState=NOT_READY OverallDuration=0
Reason=0
Trace: MRDID=1 NumTasks=0 MaxTaskLimit=1 AgtMode=1 AGTAvailabilityStatus=0
ICMAgtID=5001
Trace: SkTgtID=5001 SkGrpNo=0x0 SkGrpID=5006 NumLines=0 CurLine=0 ClientStatus=
0x0 Direction=0
```

代理以NOT_READY狀態登入後，CTI伺服器會將AGENT_STATE-EVENT傳送到Finesse。

```
MsgType:AGENT_STATE_EVENT (MonitorID:0 PeripheralID:5001 SessionID:0x0
PeripheralType:EnterpriseAgent SkillGroupState:LOGIN StateDuration:0
SkillGroupNumber:85881 SkillGroupID:5000 SkillGroupPriority:0 AgentState:
NOT_READY EventReasonCode:0 MRDID:1 NumTasks:0 AgentMode:1 MaxTaskLimit:1
ICMAgentID:5001 AgentAvailabilityStatus:0 NumFltSkillGroups:0 Direction:0
ClientSignature:"AgentID:"2001" AgentExtension:"2003" AgentInstrument:"2003"
RemaskNumMasks:1 RemaskInstrument:"2003" RemaskExtension:"2003" RemaskCallId:
0xffffffff RemaskFunctionFlag:<0x38> <LogoutCodeReq,NotRdyCodeReq,WrapDataReq>
```

```
RemaskCallMask:<0x21000000> <MC,Emerg> RemaskAgentMask:<0x0a000000> <
Logout,Avail> )Trace: AGENT_EVENT: ID=2001 Periph=5001 Ext=2003 Inst=2003 Sig=
Trace: SkgState=LOGIN SkgDuration=0 OverallState=NOT_READY OverallDuration=0
Reason=0 Trace: MRDID=1 NumTasks=0 MaxTaskLimit=1 AgtMode=1
AGTAvailabilityStatus=0 ICMagtID=5001
```

以下是從CTI伺服器接收事件的Web服務日誌。請記住，您首先會看到來自CTI Server的RAW消息，然後會看到Decoded消息。

```
%CCBU_CTIMessageEventExecutor-0-6-DECODED MESSAGE FROM CTI_SERVER: %[cti_message
=CTIAgentStateEvent [skillGroupState=0 (LOGIN), stateDuration=0, skillGroupNumber
=85881, skillGroupPriority=0, agentState=2 (NOT_READY), eventReasonCode=0,
numFltSkillGroups=0,CTIClientSignature=, agentID=2001, agentExtension=2003,
agentInstrument=2003, agentID_Long=null, duration=null, nextAgentState=null,
fltSkillGroupNumberList=[], fltSkill GroupIDList=[], fltSkillGroupPriorityList=[],
fltSkillGroupStateList=[]]CTIMessageBean [invokeID=null, msgID=30, timeTracker=
{"id":"AgentStateEvent", "CTI_MSG_RECEIVED":1366756808374,
"CTI_MSG_DISPATCH":1366756808375}, msgName=AgentStateEvent, deploymentType=CCE]]
[cti_response_time=1]: Decoded Message to Finesse from backend cti server
```

現在，Finesse從CTI伺服器收到AgentStateEvent，需要將該事件發佈到Notification服務，以便客戶端接收更新。座席知道其狀態已更改的唯一方法是接收此XMPP事件。Finesse將AgentStateEvent轉換為XMPP，並將XMPP傳送到通知服務。請注意，事件是PUT，而RequestID在負載中。

```
%CCBU_pool-5-thread-5-6-XMPP_PUBLISH_ASYNCHRONOUS: %[NodeId=/finesse/api/User/
2001] [Payload=<Update><data><user><dialogs>/finesse/api/User/2001/Dialogs
</dialogs><extension>2003</extension><firstName>Mickey</firstName><lastName>
Mouse</lastName><loginId>2001</loginId><loginName>mmouse</loginName>
<reasonCodeId>-1</reasonCodeId><roles><role>Agent</role></roles><state>NOT_READY
</state><stateChangeTime>2013-04-23T22:40:08Z</stateChangeTime><teamId>5000
</teamId><teamName>Minnies_Team</teamName><uri>/finesse/api/User/2001
</uri></user></data><event>PUT</event><requestId>6e210ca9-5786-43bc-babf-
64a397a6057f </requestId><source>/finesse/api/User/2001</source></Update>]:
```

Publishing XMPP Message Asynchronously

此處，通知服務接收更新。即使消息說無法將資料包路由到JID，也會向使用者傳送一條已發佈事件的消息。

```
RoutingTableImpl: Failed to route packet to JID: 2001@uccefinesse138.vmlload.cvp/
User packet: <message from="pubsub.uccefinesse138.vmlload.cvp" to=
"2001@uccefinesse138.vmlload.cvp/ User" id="/finesse/api/User/
2001__2001@uccefinesse138.vmlload.cvp__VI1B2"><event xmlns=
"http://jabber.org/protocol/pubsub#event"><items node="/finesse/api/User/2001">
<item id="1su0Keff8M2irdS"><notification xmlns="http://jabber.org/protocol/pubsub">
&lt;Update&gt;
```

以下是訊息正文：

```
&lt;data&gt;
&lt;user&gt;
&lt;dialogs&gt;/finesse/api/User/2001/Dialogs&lt;/dialogs&gt;
&lt;extension&gt;2003&lt;/extension&gt;
&lt;firstName&gt;Mickey&lt;/firstName&gt;
&lt;lastName&gt;Mouse&lt;/lastName&gt;
&lt;loginId&gt;2001&lt;/loginId&gt;
&lt;loginName&gt;mmouse&lt;/loginName&gt;
&lt;reasonCodeId&gt;-1&lt;/reasonCodeId&gt;
&lt;roles&gt;
&lt;role&gt;Agent&lt;/role&gt;
&lt;/roles&gt;
```

```
<state>NOT_READY</state>
<stateChangeTime>2013-04-23T22:40:08Z</stateChangeTime>
<teamId>5000</teamId>
<teamName>Minnies_Team</teamName>
<uri>/finesse/api/User/2001</uri>
</user>
</data>
<event>PUT</event>
<requestId>6e210ca9-5786-43bc-babf-64a397a6057f</requestId>
<source>/finesse/api/User/2001</source>
</Update></notification></item></items></event></message>
```

與以前一樣，XMPP消息由客戶端接收並傳送到客戶端的小工具。請注意，客戶端收到消息中帶有原始RequestID的事件。

```
Returned with status=202, content='18:40:05: Container : [ClientServices]
MasterPublisher._eventHandler() - Received event on node '/finesse/api/User/
2001': <Update>
<data>
<user>
<dialogs>/finesse/api/User/2001/Dialogs</dialogs>
<extension>2003</extension>
<firstName>Mickey</firstName>
<lastName>Mouse</lastName>
<loginId>2001</loginId>
<loginName>mmouse</loginName>
<reasonCodeId>-1</reasonCodeId>
<roles>
<role>Agent</role>
</roles>
<state>NOT_READY</state>
<stateChangeTime>2013-04-23T22:40:08Z</stateChangeTime>
<teamId>5000</teamId>
<teamName>Minnies_Team</teamName>
<uri>/finesse/api/User/2001</uri>
</user>
</data>
<event>PUT</event>
<requestId>6e210ca9-5786-43bc-babf-64a397a6057f</requestId>
<source>/finesse/api/User/2001</source>
</Update>
```

現在，客戶端已成功登入。

```
Container : SignIn._triggerLoggedIn(): Successfully logged in!18:40:05
```

註銷代碼、原因代碼、電話簿

現在，客戶端需要檢索特定於座席的資料，如註銷代碼、原因代碼和電話簿。以下是向客戶端請求該資訊。

```
Container : SignIn._triggerLoggedIn(): Successfully logged in!18:40:05:
Container : [ClientServices] Dialogs: requestId='undefined', Making REST
request: method=GET, url='/finesse/api/User/2001/Dialogs?nocache=
1366756805355?
```

```
18:40:05: Container : [ClientServices] User: requestId='undefined',
Making REST request: method=GET, url='/finesse/api/User/2001/ReasonCodes?
```



```
category=LOGOUT&nocache=1366756805356'18:40:05: Container : [ClientServices]
User: requestId='undefined', POST_DATA=''18:40:05: Container : _displayUserData
(): User's current state is: NOT_READY

'18:40:05: Container : [ClientServices] User: requestId='undefined', Making REST
request: method=GET, url='/finesse/api/User/2001/ReasonCodes?category=NOT_READY&
nocache=1366756805358

18:40:05: Container : [ClientServices] User: requestId='undefined', POST_DATA=
''18:40:05: Header : The client logger has been initialize for the header
18:40:05: Header : _displayUserData(): User's current state is: NOT_READY

18:40:05: Header : Container._initGadgetContainer(): Initializing gadget
container.
18:40:05: Header : FailoverMonitor.startListening(): Listening for triggers
18:40:05: Header : PageServices.stopTimeoutPoller(): Cancelling connection
timeout and poller...
18:40:05: Header : [ClientServices] id=2001: TypeError: 'this._listenerCallback
[...].callback' is null or not an object
```

同樣的邏輯也適用於這些請求。請記住，Finesse原因代碼和PhoneBook儲存在Finesse資料庫中，而不是UCCE中。