

Ловко обойдите трассировку входа в систему агента с использованием журналов

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Запустите рабочее место агента](#)

[Учетные данные входа в систему агента](#)

[SystemInfo](#)

[API REQUEST](#)

[Установите соединение BOSH](#)

[Вход агента](#)

[Выполните вход в систему](#)

[Коды выхода из системы, коды причины, PhoneBook](#)

Введение

Этот документ описывает процесс, вовлеченный во вход в систему агента через систему Изящества с файлами журнала. Важно понять поток сообщений между другими Компонентами Изящества, Сервером Интеграции компьютерной телефонии (CTI) и компьютером клиента так, чтобы можно было успешно решить проблемы.

Предварительные условия

Требования

Cisco рекомендует ознакомиться с Изяществом Cisco и приглашением команды CLI голосовой операционной системы (VOS).

Используемые компоненты

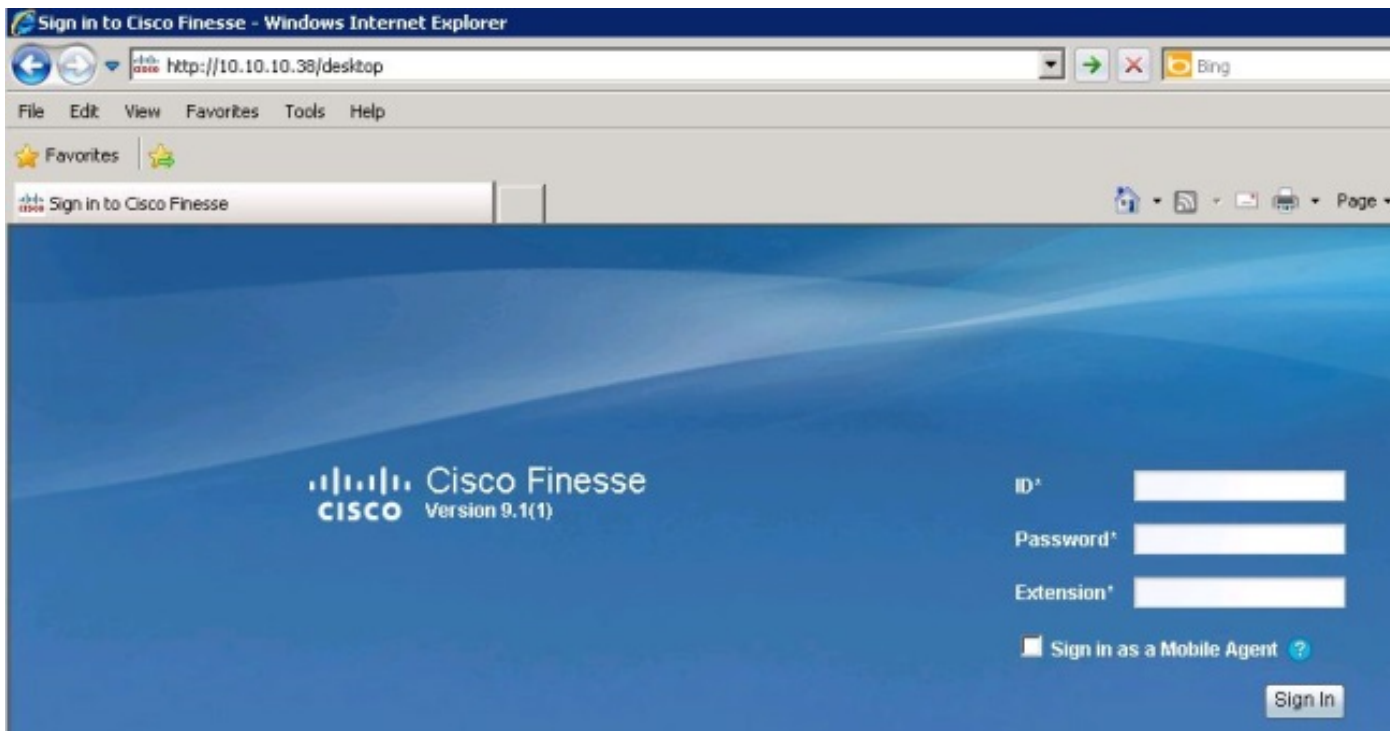
Сведения в этом документе основываются на Версии 9.1 (1) Изящества Cisco.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить

потенциальное воздействие всех команд до их использования.

Запустите рабочее место агента

Для запуска рабочего места агента скопируйте этот URL в web-браузер: **http://server> изящества <your / рабочий стол**. В Версии 9.1 Изящества, HTTP или HTTPS поддерживается.



Изящество использует Tomcat в качестве Web-сервера. При запуске web-браузера запрос выполнен для Ухищрения для представления рабочего места агента вам. Cisco Tomcat **localhost_access_log** команда показывает запрос загрузить рабочее место агента.

```
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
```

Учетные данные входа в систему агента

Теперь, когда рабочее место агента было представлено, вы вводите свои учетные данные входа в систему. Прежде чем Изящество может передать запрос регистрации в системе к CTI Server, клиент должен установить Соединение Двухнаправленных потоков по синхронному HTTP (BOSH). Для установления Соединения BOSH клиент сначала запрашивает Сведения о системе от Сервера Изящества.

SystemInfo

Рабочий стол клиента сделал Представительную государственную Передачу (REST) запрос Прикладного программного интерфейса (API) к этому URL: `/finesse/api/SystemInfo`. Примите во внимание `nocache =`. Этот уникальный идентификатор используется для отслеживания этого запроса через систему. **Возвращенный с status=200** указывает, что был успешно получен запрос.

```
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
```

Если у вас нет `clientlogs`, но необходимо отследить запрос, можно искать `Tomcat localhost_access_log` для определения, когда остальное запрос API было сделано и определять местоположение уникального идентификатора.

```
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 передает этот запрос API Репозитория web - приложения (WAR) API REST Изящества. Для обнаружения журналов API REST Изящества ищите, веб-сервисы Изящества регистрируют или меткой времени или `nocache ID` для определения местоположения `API_REQUEST`. Этот журнал показывает `REQUEST_START`, `REQUEST_URL`, `REQUEST_END` и `elapsed_time`, который система взяла для завершения запроса.

```
%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
```

Содержание возвратилось к клиенту остальными, запрос API для получения Сведений о системе показывают здесь. Эта информация расположена в клиенте (агент) журналы.

```
%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
```

Установите соединение BOSH

SystemInfo показывает Основные и Вторичные серверы Изящества, статус Изящества как **IN_SERVICE**, **xmppDomain** и **xmppPubSubDomain**. У клиента теперь есть достаточно информации для установления соединения BOSH.

```
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!
```

Клиент успешно подписан на Объект Изящества (узел)/**finesse/api/User/2001**, как только установлено соединение BOSH.

Когда соединение BOSH клиента установлено, журнал веб-сервисов получает сообщение **PRESENCE_NOTIFICATION** от клиента. Этот **PRESENCE_TYPE** только указывает, что клиент доступен для получения **Событий XMPP** и не имеет никакого отношения к доступности агента в Унифицированном предприятии Contact Center (UCCE). Помните, что агент еще не регистрируется.

Примечание: Вы только видите сообщения **PRESENCE_TYPE**, когда клиент устанавливает соединение BOSH или когда разъединено соединение BOSH клиента. Когда соединение BOSH клиента разъединено, показы **PRESENCE_TYPE** как недоступное.

Вот Событие уведомления в журнале веб-сервисов:

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

Вход агента

Теперь, когда клиент установил соединение BOSH, процесс входа в систему начинается. Клиент делает другой запрос API REST для получения информации о текущем пользователе. Для выполнения этого запроса перейдите к этому URL:**/finesse/api/User/2001** и вводят **method=GET**.

Поскольку это - другой Запрос API, **nocache ID** является другим. Так, для отслеживания этого запроса необходимо использовать этот новый 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,
```

Можно найти этот запрос в Tomcat **localhost_access_log** в случае необходимости. Вот то, как вы находите его в журнале веб-сервисов:

```
%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_identifer=null][request_method=user.GET][request_parameters=2001]:
Request from client to webservice api
```

Вот запрос в журнале Сервисов уведомлений. Примите во внимание **HTTP/1.1 200 OK**.

Примечание: Журнал Уведомления Cisco для получения информации только. При включении регистрации Уведомления Изящества Cisco она влияет на производительность.

```
>> "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]"
```

Теперь, когда Сервис уведомлений имеет запрос, он размещает информацию для этого пользователя. Вот POST от Журнала Сервиса уведомлений, который переходит к клиенту:

```
>> "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]"
```

Это **Событие XMPP**, которое является агентом **2001** в данном примере, передается всем клиентам подписки. JavaScript в клиенте получает **Событие XMPP**, и Событие передается Гаджету в клиенте. Вот clientlogs, которые показывают содержание ответа:

```
Commencing sign-in process18:40:05: Container : [ClientServices] User: requestId=
'undefined', Murl='/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>
```

Выполните вход в систему

Теперь клиент готов выполнить вход в систему. Заметьте **RequestID**. RequestID передается в органе по запросу. Вы используете этот RequestID для придержива запроса регистрации в системе **остальным API> CTI> API REST> Сервис уведомлений> Ответ** назад клиенту. Этим запросом является PUT, что означает, что клиент запрашивает ОБНОВЛЕНИЕ или изменение к его текущему состоянию.

```
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>
```

API REST Изящества получает этот запрос от клиента. Затем API передает **SetAgentStateReq** к CTI Server.

```
%CCBU_http-8080-7-6-API_REQUEST: %[REQUEST_URL=User/2001][agent_id=2001]
[request_identifier=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 Server получает запрос.

```
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
```

Как только в агента входят со статусом **NOT_READY**, CTI Server передает **AGENT_STATE-СОБЫТИЕ** для Ухищрения.

```
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
```

Вот журнал веб-сервисов, который получил событие от CTI Server. Помните наблюдение **Необработанного сообщения** от CTI Server сначала, и затем вы видите **Декодируемое сообщение**.

```
%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
```

Теперь, когда Изящество получило **AgentStateEvent** от CTI Server, событие должно быть **Публиковано** в Сервисе уведомлений так, чтобы клиент получил **ОБНОВЛЕНИЕ**.

Единственный способ для агента знать, что его/ее состояние изменилось, путем получения этого **События XMPP**. Изящество преобразовывает **AgentStateEvent** в **XMPP** и передает **XMPP** к Сервису уведомлений. Заметьте, что Событием является **PUT**, и RequestID находится в Информационном наполнении.

```
%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
```

Здесь, Сервис уведомлений получает **ОБНОВЛЕНИЕ**. Даже при том, что SAID сообщения, **подведенные к маршрутизированному пакету к JID**, сообщение, что событие было опубликовано, передаются пользователю.

```
RoutingTableImpl: Failed to route packet to JID: 2001@uccefinesse138.vmlod.cvp/
User packet: <message from="pubsub.uccefinesse138.vmlod.cvp" to=
"2001@uccefinesse138.vmlod.cvp/ User" id="/finesse/api/User/
2001__2001@uccefinesse138.vmlod.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;
```

Вот тело сообщения:

```
&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;
```

```
<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></notification></item></items></event></message>
```

Как прежде, сообщение XMPP получено клиентом и передано к Гаджету клиента. Заметьте, что клиент получает событие с исходным RequestID в сообщении.

```
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>
```

Теперь в клиента успешно входят.

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

Коды выхода из системы, коды причины, PhoneBook

Теперь клиент должен получить специфичные для агента данные, такие как Выход из системы кодирует, Коды причины и Телефонная книга. Вот запрос о той информации, сделанной клиенту.

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

Та же логика применяется к этим запросам. Следует иметь в виду, что Коды причины Изящества и PhoneBook сохранены в базе данных Изящества, не в UCCE.