PSOM 4.0.1 Logging

In this section we collect all the logs about events that has happened. This type of logs are written once the event has finished and do not represent a live photograph of the state of the service or of the appliance. They are mostly useful for debugging user's issues and to provide support.

figure 1. the static logs menu

As you can see in figure 1. the static logs menu all these logs are grouped under the "LOGGING" label into the main menu.

4.0.1.1 Auditing

The **Auditing** page shows any change to the configuration performed via web console.

AuditLogEvent List

Timestamp	Username	Change	Object Type	Object ID
2013-10-09 13:11:46.000		INSERT	User's role	
2013-10-09 13:11:27.000		INSERT	User's role	
2013-10-09 11:02:31.000		INSERT	Asterisk Setting	937
2013-10-09 11:02:31.000		INSERT	Asterisk Setting	936
2013-10-09 11:02:30.000		INSERT	Setting	84
2013-10-09 11:02:30.000		DELETE	Setting	83
2013-10-08 17:16:35.000		INSERT	Asterisk Setting	935
2013-10-08 17:16:35.000		INSERT	Asterisk Setting	934
2013-10-08 17:16:34.000		INSERT	Setting	83
2013-10-08 17:16:34.000		DELETE	Setting	82
2013-10-08 16:34:51.000		UPDATE	privateserver.SmsSenderConfig	5
2013-10-08 15:45:58.000		UPDATE	Repository	11
2013-10-08 15:44:52.000		UPDATE	Repository	11
2013-10-08 15:44:52.000		INSERT	Asterisk Setting	933
2013-10-08 15:44:52.000		INSERT	Asterisk Setting	932
2013-10-08 15:44:51.000		INSERT	Setting	82
2013-10-08 15:44:51.000		DELETE	Setting	81
2013-10-08 15:18:58.000		INSERT	Asterisk Setting	931
2013-10-08 15:18:58.000		INSERT	Asterisk Setting	930
2013-10-08 15:18:57.000		INSERT	Setting	81
2013-10-08 15:18:57.000		DELETE	Setting	80
2013-10-08 15:12:21.000		UPDATE	Account	737
2013-10-08 15:12:02.000		UPDATE	privateserver.SmsSenderConfig	10
2013-10-08 15:12:02.000		UPDATE	privateserver.SmsSenderConfig	9
2013-10-08 15:12:02.000		UPDATE	privateserver.SmsSenderConfig	8
2013-10-08 15:12:02.000		UPDATE	privateserver.SmsSenderConfig	7
2013-10-08 15:10:40.000		UPDATE	Account	737
2013-10-08 15:10:32.000		INSERT	privateserver.SmsSenderConfig	10
2013-10-08 15:10:32.000		INSERT	privateserver.SmsSenderConfig	9
2013-10-08 15:10:32.000		INSERT	privateserver.SmsSenderConfig	8

1 2 3 4 5 6 7 8 9 10 .. 78 Next Filter Pure and Unfiltered!

CSV

figure 2. List of Audit Events Log List

As shown above in figure 2. List of Audit Events Log List, we can collect several informations:

- Timestamp: the exact time the event happened.

> ⚠ Please remember that the timestamp is related to the appliance's time configuration. It's local time which gets reported.

- Username: the user who performed the change is traced by the username used in the login.
- Change: the type of changes performed.
- Object Type
- Object ID

These informations can be used for security checks as for post-issue analysis.

4.0.1.2 Web Sessions

In the **web sessions** it's possible to read the list of all the access made to the web console.

figure 3. List of Web Session Logs

Just click on the **Web Sessions** entry in the **main menu** and you get a list as in figure figure 3. List of Web Session Logs. Fields shown are:

- **Date Created**: this is when the event occurred
- **Event Type**: there are several type of Events (see specifications below)
- **Principal**: the username used to log in.
- **IP Address**: the IP address from which the connection has been performed.

> ⓘ If the user listed is "anonymousUser" then this is an event triggered by the system. This is specifically true for events like the "SESSION_TIMEOUT".

Event Types

Here follows a list of all the event types logged in this table:

- **SESSION_TIMEOUT** : pretty self explanatory
- **LOGOUT** : The user performed a logout
- **SUCCESS_INTERACTIVE** / **SUCCESS** : these two events always come together and indicates a login has successfully performed. The former one points out the login happened through the web interface ("interactive"), the latter is a generic successful login log.
- **FAIL_USER_NOT_FOUND** : the login used was not found among the users configured.
- **FAIL_CREDENTIALS_EXPIRED** : the password or the credentials used are set as expired
- **FAIL_BAD_CREDENTIALS** : wrong password

4.0.1.3 Call Detailed Records

Despite its name the CDR is a debugging and quality assurance facility. It saves all the calls status, meaning it is very useful to understand is something is going wrong with you Secure Call Service.

To access the CDR you must click on the **Call Detailed Record** in the main menu. You'll get the "Cdr List" page which includes all the calls recorded.

figure 4. Call Detailed Record List

If the table is empty, please place a phone call between the two SIp Accounts. Then come back on this page and check that the call has been correctly registered. The shown fields are:

- **Call Date**: when the call has been placed.
- **Caller Number**: The virtual phone number used to place the call.
- **Caller**: The caller description, if any.
- **Caller Group**: The caller's group description, if any.
- **Callee Number**: The virtual phone number called.
- **Callee**: The callee description, if any.
- **Callee Group**: The callee's group description, if any.
- **Total Duration**: how long the call lasted. Time elapsed from the moment the "Call Button" is pressed on the caller's client to the one in which the communication is closed at all.
- **Call Duration**: how long the call lasted, just the voice.
- **Disposition**: which result the call had.
- **Hangup Cause**: how long the call lasted.
- **Call id**: a unique number that identifies the call.
- **Call type**: nature of the call performed.
- **Call route**: if the call transited on a trunk this field would shown the trunk's name.

Relevant fields in CDR

The **Disposition** is very important because it tells you the exit code of each call. Possible codes are:

1. **ANSWERED**: the call was taken by the callee
2. **NO_ANSWER**: nobody picked up the phone to answer (mostly this is a time out code)
3. **BUSY**: the callee refused the call
4. **FAILED**: for some reason the call was not able to be placed

The **Hangup Cause** describes in details what caused the hangup. Possible causes are:

1. **NO_ROUTE_DESTINATION**: The callee is not reachable because of network issues.
2. **NORMAL_CLEARING**: Call closed normally.
3. **USER_BUSY**: The user is busy.
4. **NO_ANSWER**: The callee didn't answer and the call was closed for timeout reason.

5. **CALL_REJECTED**: The callee rejected the call, as by pressing the "hold" button.
6. **BEARERCAPABILITY_NOTAVAIL**: The caller and the callee were using different security models that are not compatible.
7. **NO_USER_RESPONSE**: The same as in "NO_ANSWER", but this happens when the call goes out by Trunk.

The **Call Type** identifies the nature of the call. Used with the Call id fields this value is useful to trace down complex calls like three-way calls, conferences or transferred calls. Possible values are:

1. **CALL**: Usual call between two persons
2. **3-WAY**: Usual call (as above) with an added person during the call
3. **CONFERENCE**: Conference call
4. **TRANSFER**: Usual call that is transferred to a third person

The **Call id** is a number that identifies uniquely the call. It can be used to trace a call if it changes its nature (ie becoming a three-way call) or to group all the participants (ie in a conference room).

ⓘ It's possible to avoid any call record, as a Privacy option. To enable the "No-CDR" option, you just have to set the CDR Period to "DAY" and the duration to '0' (zero).

⊘ The above statement about the CDR Privacy option is true only for the DIRECT calls (which are the calls between two users). Conferences and 3-way calls will be logged as ever, though.

4.0.1.4 SIP Sessions

The **SIP Sessions** page show the activities each Account did with the server.

figure 5. List of SIP Session Activities

To get this list just click on the **SIP Sessions** entry in the main menu. The activities are listed by date and they give you a detailed overview of the SIP status for each one. These logs are very useful for debugging the networking issues on the client side.

✅ In figure 5. List of SIP Session Activities you can read that a **NETWORK_ERROR** has caused the DISCONNECT event for one user.

The **Event** column lists the SIP events:

- **CONNECT:** PrivateGSM client opened a connection to PrivateServer. This usually means the client has been activated
- **REGISTER:** The Account has been correctly registered and is now **on line**
- **UNREGISTER:** The Account has been correctly unregistered and is now **off line**
- **DISCONNECT:** PrivateGSM client closed the connection. This usually means the client has been stopped.

Each one has its **Details** column which explains the exact message provided by the PrivateServer.
CONNECT/DISCONNECT event are bound to a remote address, not directly related to a specific VoIP account. A periodic background task analyses the SIP session logs and, when possible, reconcile them binding these events to a specific account.
Reconciliation is very useful while debugging SIP session for a specific user: clicking on username field will show a filtered list of SIP session events.

4.0.1.5 App log configuration

As the web interface is made from a Java application in Tomcat, then you can set the logging levels that are going to be written in the catalina.out log file.

⊘ This is an advanced feature usually useful only for debugging purposes. It's not recommended to change the default settings.

---

figure 6. Set Logging levels

In order to change a log level you have first to select a component of the appliance from the left drop-down menu (please check the figure 6. Set Logging levels). Then you can set a logging level from the right drop-down menu.

When you're done just press the **Submit** button and the catalina.out log file would change its log level accordingly.

ⓘ

4.0.1.6 Install messages

> ⓘ **NEW FEATURE!**
>
> This log is available starting with the current version of PrivateServer.

Install messages are one amongst the numerous means for installing PrivateGSM on the customers' mobile device.

> ⚠ To make the Install messages work fine you have to configure the Application download URL.

What we accomplish sending an Installation message is to help the customer to find out the correct edition of PrivateGSM application (either Professional or Enterprise) and automatically install it.

figure 7. list of the install messages sent

main features of the logs of the install messages are:

- **Timestamp**: the exact date & time when the message has been sent
- **Account**: to who the message has been sent to
- **Platform**: which mobile platform
- **Variant**: this could be either:
    - **ZRTP**: this is the **Professional** Edition
    - **SDES**: this is the **Enterprise** Edition
- **Sent**: if the message results to be effectively sent or if there were issues in sending it
- **Sent by**: the User who sent the Install Message.
- **Virtual phone number**: the number of the mobile device where the text has been sent to

figure 8. details of one install message

Clicking on the **Timestamp** field it becomes possible to reveal details about each message. You can have an example in figure 8. details of one install message.

ⓘ

> ⓘ **Tip**
>
> Clicking on the **Account** field brings you straight to the Account's detail.

The most important one among them is no doubt the **Message link**. That's because it's possible to use it in order to create a message to be sent by other means. This composed message would have the same validity of any text message sent to deliver an Install Message to the user.

> ⓘ **Tip**
>
> An example of a good use of this field in the day-by-day business is copying it into an email and sent it to the user, asking him to open the email on his mobile and then click on the link inside the email. In this way you has a service manager would have a perfect replacement for the text message as a mean for delivering the Install Message.

4.0.1.7 Provisioning messages

> ⓘ **NEW FEATURE!**
>
> This log is available starting with the current version of PrivateServer.

Provisioning messages are basically the mean for delivering the download URI of the provisioned configuration. Long story short: whenever you push an automatic activation by sending the provisioning message you're sending a text message that contains the URI to the user's configuration. Part of this configuration is created by the Provisioning Profile, part of it is taken by the Account configuration. Nevertheless, it's always a configuration file to be downloaded and then installed in your PrivateGSM.

figure 9. Log list of the provisioning messages

Clicking on the **Provisioning Messages** link brings you to the logs list shown in figure 9. Log list of the provisioning messages. Here you have all the primary informations about the automatic activation performed:

- the **User** who **sent** each provisioning message
- the **timestamp** pointing out when each message has been sent
- the **Virtual phone number** where the message has been sent to
- the status of the message: either it's been **Consumed** or not. This means if the customer has ever clicked on the link inside the text message
- the **Account** to which the message has been sent to
- if the text message has been **Sent** or not, meaning if any problem arose during the delivery and the text never left the server.
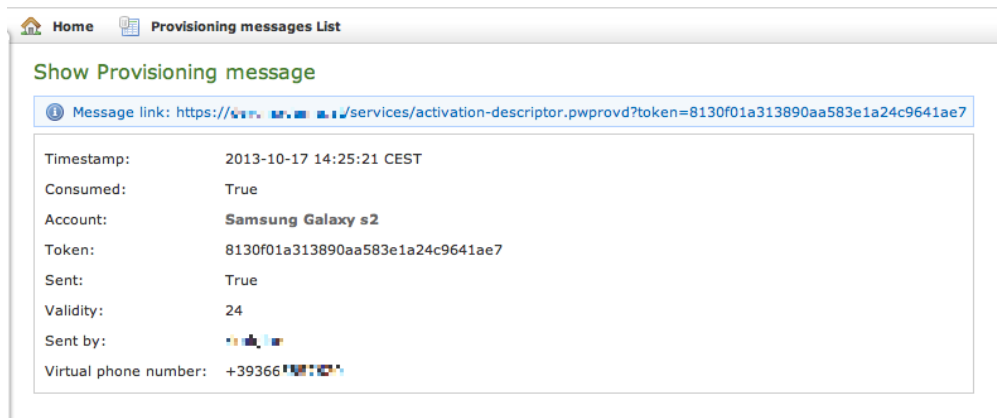
figure 10. detail of provisioning message

Clicking on the **Timestamp** field it becomes possible to reveal details about each message. You can have an example in figure 10. detail of provisioning message.

> ⓘ **Tip**
>
> Clicking on the **Account** field brings you straight to the Account's detail.

Three are the contents shown in the detail form that are not present in the list above:

1. The **Message link**
2. The **Token**
3. The **Validity**

The most important one among them is no doubt the **Message link**. That's because it's possible to use it in order to create a message to be sent by other means. This composed message would have the same validity of any text message sent to deliver a provisioning profile to the user.

> ⓘ **Tip**
>
> An example of a good use of this field in the day-by-day business is copying it into an email and sent it to the user, asking him to open the email on his mobile and then click on the link inside the email. In this way you has a service manager would have a perfect replacement for the text message as a mean for delivering the Provisioning Profile.

PSOM 4.0.2 Real-Time management