

Kalliope[®] PBX V4

Documentazione API

per CDR

1 Introduzione alle API REST Kalliope

1.1 Descrizione generale

KalliopePBX V4 offre un set di API REST che possono essere invocate da sistemi esterni per compiere azioni di tipo dispositivo (es. creazione di un interno, generazione di un backup e download, ecc.) o di consultazione (download del registro chiamate, ecc.).

L'accesso alle API può avvenire tramite HTTP o HTTPS, previa autenticazione, secondo il meccanismo descritto di seguito.

1.2 Meccanismo di autenticazione

Il meccanismo di autenticazione delle API REST utilizza le stesse credenziali utente valide per l'accesso web. In particolare, ogni richiesta deve contenere un HTTP header custom, monouso, il cui nome è **X-authenticate**.

La procedura di autenticazione è one-way, ossia non richiede l'invio di un challenge da parte del server, ma tutte le informazioni necessarie all'autenticazione sono generate esclusivamente lato client.

La proprietà di uso singolo dell'header di autenticazione previene gli attacchi per duplicazione, garantendo che la conoscenza dell'header X-authenticate di una certa richiesta non permetta di effettuarne un'altra illecita.

1.3 Costruzione dell'X-authenticate header

L'header è composto da una serie di dati in chiaro ed un digest costruito utilizzando tali dati, unitamente alla password dell'utente che effettua la richiesta. Di seguito un esempio di un X-authenticate header completo:

```
X-authenticate: RestApiUsernameToken Username="admin", Domain="default",  
Digest="+Pjg7Tb3v98XnL6iJVv+v5hwhYjdzQ2tIWxvJB2cE40=",  
Nonce="bfb79078ff44c35714af28b7412a702b", Created="2016-04-29T15:48:26Z"
```

I dati che sono utilizzati (e trasmessi nella richiesta) nella costruzione dell'header sono:

- **Username:** lo username dell'utente (es. admin)
- **Domain:** il dominio del tenant di appartenenza dell'utente. Nel caso di sistemi mono-tenant, il dominio predefinito è "default"
- **Nonce:** una stringa casuale esadecimale generata dal client, di lunghezza almeno pari a 8 caratteri. Ad ogni richiesta è necessario rigenerare un nuovo *nonce*, in quanto il sistema tiene traccia di tutti quelli utilizzati in precedenza, per un periodo pari al tempo di validità del *nonce*, che è pari a 5 minuti. Dopo 5 minuti è possibile riutilizzare un *nonce* utilizzato in precedenza; la protezione da attacchi basati su replica è in questo caso garantita dal timestamp di creazione del *nonce*
- **Created:** indica il timestamp di creazione del *nonce*; se l'ora di ricezione della richiesta differisce dall'ora impostata sul KalliopePBX per più di 5 minuti, la richiesta viene considerata invalida e rifiutata. Questo garantisce che il riuso di un X-authenticate header con lo stesso *nonce* non sia possibile né entro i 5 minuti del tempo di vita di un *nonce* (per la protezione da replica implementata) né oltre tale tempo di vita. Il formato della stringa del timestamp (riportato al timezone UTC) è "YYYY-MM-DDThh:mm:ssZ".

Il campo **Digest** è un hash dei dati sopra descritti e della password dell'utente, e si calcola applicando la funzione di hash **sha256** alla concatenazione delle seguenti stringhe:

Nonce, digestPassword, Username, Domain e Created (concatenazione delle stringhe senza delimitatori):

```
Digest = sha256(Nonce + digestPassword + Username + Domain + Created)
```

Dato che sul KalliopePBX le password utente non sono salvate in chiaro ma sotto forma di hash non reversibile, per la generazione del Digest è necessario utilizzare l'hash della password con la stessa procedura utilizzata nel PBX (indicata come **digestPassword**) e non la password in chiaro.

La procedura di calcolo della **digestPassword** richiede, oltre alla password dell'utente, una ulteriore stringa denominata **salt**, univoca per tenant. Il salt può essere ottenuto dal KalliopePBX mediante una API REST apposita, invocabile in modo anonimo (senza autenticazione).

La formula di calcolo della **digestPassword** è¹:

```
digestPassword = sha256(password{<salt>})
```

Ad esempio, supponendo che la password dell'utente sia "admin" e il salt sia uguale a "b5a8fdcf2f8d5acdada33c4a072a97d7a", la **digestPassword** risultante è:

```
digestPassword = sha256(admin{b5a8fdcf2f8d5acdada33c4a072a97d7a}) =  
dd7b0be7fa37d6cbaf0b842bf7532f229cb79ab8d54d509c2aa7eea27a53cd5e
```

Come esempio, con i dati seguenti:

```
Nonce: bfb79078ff44c35714af28b7412a702b  
digestPassword:  
dd7b0be7fa37d6cbaf0b842bf7532f229cb79ab8d54d509c2aa7eea27a53cd5e  
Username: admin  
Domain: default  
Created: 2016-04-29T15:48:26Z
```

si ottiene:

```
Digest =  
base64(sha256binary(bfb79078ff44c35714af28b7412a702bdd7b0be7fa37d6cbaf0b842bf7  
532f229cb79ab8d54d509c2aa7eea27a53cd5eadmindefault2016-04-29T15:48:26Z))
```

che vale **+PJg7Tb3v98XnL6iJVv+v5hwhYjdzQ2tIWxvJB2cE40=**.

¹ I caratteri { e } fanno parte della stringa di cui si esegue lo sha256 e devono essere inseriti

NOTA: la funzione sha256_{binary}(..) indica la funzione sha256(..) che restituisce l'output in formato **binario** e non in formato esadecimale.

La stringa completa risultante da inserire all'interno dell'header X-authenticate è quindi quella indicata all'inizio del paragrafo e riportata di seguito:

```
X-authenticate: RestApiUsernameToken Username="admin", Domain="default",  
Digest="+PJg7Tb3v98XnL6iJVv+v5hwhYjdzQ2tIWxvJB2cE40=",  
Nonce="bfb79078ff44c35714af28b7412a702b", Created="2016-04-29T15:48:26Z"0
```

1.4 Elenco completo API

La lista dettagliata e aggiornata delle API è consultabile fino alla versione firmware 4.8.x tramite l'interfaccia web di KalliopePBX all'indirizzo **http://KALLIOPE_IP_ADDRESS/api/doc**.

A partire dalla versione firmware **4.9.4** è disponibile una **collection Postman** che integra anche il codice per aggiungere automaticamente l'header di autenticazione richiesto (è necessario solo impostare l'indirizzo IP del PBX e le credenziali username/password dell'utente con cui invocare l'API). È possibile scaricare il file di collection da questo link:

https://www.kalliopepbx.com/wiki/KalliopePBX%20API%204.9.4.postman_collection.json.zip

1.5 Classi per la generazione degli header di autenticazione

Di seguito un elenco di progetti in vari linguaggi di programmazione per la generazione e validazione degli header di autenticazione.

- **phpRestApiUtils** (<https://github.com/NetResultsIT/phpRestApiUtils>): implementazione in PHP
- **RegEx** (<https://regex101.com/r/IEgLV/1>): espressione regolare utile per validare gli header generati

2 API REST CDR

Le API REST per l'accesso al CDR permettono di filtrare per periodo di tempo o per singola chiamata restituendo l'informazione in uno dei vari formati supportati.

Le richieste possono essere fatte sia in formato GET che in formato POST, in quest'ultimo caso il corpo della richiesta può essere sia in formato XML che JSON.

2.1 URL e parametri

L'URL di base per le richieste al CDR è ***http[s]://<indirizzo IP KalliopePBX>/rest/cdr*** al quale possono essere aggiunti ulteriori parametri nel seguente modo:

```
/rest/cdr/{format}
/rest/cdr/{format}/{years}
/rest/cdr/{format}/{years}/{months}
/rest/cdr/{format}/{years}/{months}/{days}
```

Il parametro **{format}** è obbligatorio e descrive il formato di output dell'API. Il valore di questo parametro deve essere uno dei seguenti:

Valore	Descrizione
summary	Viene restituita una riga per ogni chiamata con le informazioni aggregate
detailed	Per ogni chiamata vengono restituite anche tutte le righe di dettaglio della stessa
blues_out	Formato richiesto dal software Blues
v3_compat	Formato di compatibilità con KalliopePBX V3

Tutti gli altri parametri sono opzionali, se non viene specificato nessun altro parametro la ricerca sarà effettuata sul mese corrente.

Il parametro **{years}** può essere un anno nel formato **YYYY**, oppure un range di anni nel formato **YYYY-YYYY**.

Il parametro **{months}**, analogamente al parametro **{years}**, può essere un mese nel formato **MM**, oppure un range di mesi nel formato **MM-MM**.

Anche il parametro **{days}** rispecchia la stessa logica, può essere un singolo giorno nel formato **DD**, oppure un range di giorni nel formato **DD-DD**.

I parametri sono posizionali all'interno dell'URL quindi se vogliamo specificare il parametro **{days}** dovranno essere obbligatoriamente specificati anche i parametri **{years}** e **{months}**.

ESEMPIO: Supponiamo ad esempio di voler ricercare le chiamate tra il giorno 12/01/2016 ed il giorno 15/02/2016 ed avere un formato dettagliato di queste. L'URL al quale effettuare la richiesta GET sarà il seguente:

```
http[s]://<indirizzo IP KalliopePBX>/rest/cdr/detailed/2016/01-02/12-15
```

2.2 Content-Type della risposta per i vari formati richiesti

A seconda del formato richiesto - **summary**, **detailed**, **blues_out** o **v3_compat** - il Content-Type della risposta sarà diverso da quello richiesto tramite l'header *Accept*.

Di seguito una tabella riassuntiva dei Content-Type possibili per i vari formati richiesti.

format	Content-Type
summary	Sono supportati i formati di output sia in formato XML che JSON che CSV
detailed	Sono supportati i formati di output sia in formato XML che JSON che CSV
blues_out	Il formato Blues è un testo semplice per cui in questo caso il Content-Type della risposta sarà sempre text/plain indipendentemente dal formato richiesto con l'header Accept
v3_compat	Il formato di compatibilità con KalliopePBX V3 può essere restituito al client sia in formato XML che JSON che CSV

2.3 Esempi di output nei vari formati

XML:

```

<?xml version="1.0"?>
<cdr>
  <call>
    <id>1463997154.0</id>
    <source>local_exten</source>
    <start_time>2016-01-12 11:52:34</start_time>
    <answer_time></answer_time>
    <end_time>2016-01-12 11:52:34</end_time>
    <account_code>206</account_code>
    <caller>...</caller>
    <gateway_name></gateway_name>
    <called>050...</called>
    <status>FAILED</status>
    <answered_by></answered_by>
    <bill_secs>0</bill_secs>
    <duration>0</duration>
    <destination></destination>
  </call>
  ...
  <call>
    ...
  </call>
</cdr>

```

JSON:

```

[
  {
    "id": "1463997154.0",
    "source": "local_exten",

```

```
"start_time": "2016-01-12 11:52:34",  
"answer_time": "",  
"end_time": "2016-01-12 11:52:34",  
"account_code": "206",  
"caller": "...",  
"gateway_name": "",  
"called": "050...",  
"status": "FAILED",  
"answered_by": "",  
"bill_secs": 0,  
"duration": 0,  
"destination": ""  
},  
...  
{  
...  
}  
]
```

CSV:

```
#id,source,start_time,answer_time,end_time,account_code,caller,gateway_name,  
called,status,answered_by,bill_secs,duration,destination  
1463997154.0,local_exten,2016-01-12 11:52:34,,2016-01-12  
11:52:34,206,,,,,050...,FAILED,,0,0,  
...
```

2.4 Richieste POST

Con i soli parametri dell'URL la minima granularità di ricerca possibile è di un giorno perché non è possibile specificare orari di ricerca.

Questo limite è facilmente superabile con una richiesta POST.

ESEMPIO: Supponiamo di voler ricercare le chiamate del giorno 12/01/2016 tra le 15:00 e le 16:00, per farlo è sufficiente inviare nel corpo della richiesta POST il seguente XML:

```
<?xml version="1.0"?>  
<kpbx_request>  
<cdr>  
<begin>2016-01-12 15:00:00</begin>
```

```
<end>2016-01-12 16:00:00</end>  
</cdr>  
</kpbx_request>
```

Nel caso delle richieste POST l'unico parametro tra quelli dell'URL preso in considerazione è **{format}**, tutti gli altri anche se inseriti non concorreranno alla creazione dei filtri di ricerca.

Oltre ai tag **<begin>** ed **<end>** è possibile specificare il tag **<unique_id>** all'interno del quale indicare uno **unique id** specifico da ricercare. Questo tag può essere utilizzato in congiunzione agli altri due, in questo caso la politica di ricerca sarà 'AND' ovvero lo specifico unique id sarà ricercato all'interno del periodo richiesto.

Anche nel caso di richieste POST se non viene indicato l'intervallo temporale la ricerca sarà effettuata sul mese corrente.

È possibile specificare il corpo della richiesta anche in formato JSON come nell'esempio seguente:

JSON:

```
{  
  "cdr":{  
    "begin":"2018-05-08 12:00:00",  
    "end":"2018-05-08 14:00:00"  
  }  
}
```