trivum documentation de l’API

trivum API

trivum Interface HTTP . . . 1
1. Commandes . . . 2
1.1. ZoneCommand . . . 2
1.2. Définir la source de la zone . . . 4
1.3. Définir l’attribut de zone . . . 5
1.4. trivum Favoris . . . 5
1.5. Playlists trivum . . . 6
1.6. TuneIn Favoris . . . 6
1.7. Presets FM . . . 6
1.8. Statut et contrôle du NAS . . . 6
1.9. Direction du groupe . . . 7
1.10. Pagination . . . 9
2. Sélection musicale interactive . . . 9
3. Obtenir le statut de la zone . . . 10
3.1. Synchrone . . . 10
3.2. Asynchrone . . . 11
3.3. Annexe : exemple schématique d’une application client Visu . . . 12
3.3.1. Application à un seul fil . . . 12
3.3.2. Exemple d’application à deux threads . . . 13
4. Guide d’intégration simple . . . 13
4.1. État général de la zone . . . 14
4.2. Statut de zone unique . . . 15
4.3. Exécuter une commande . . . 16

trivum technologies GmbH <info@trivum.com> v0.1, 2025-02-27 :title-logo-image: image::../images/trivum-logo.svg[pdfwidth=150,align=right]

trivum Interface HTTP

25-juillet-2023

L’interface HTTP trivum prend des requêtes qui peuvent être testées facilement par un navigateur Web et renvoie des réponses au format XML.

1. Commandes

1.1. ZoneCommand

Permet de faire des choses simples comme désactiver une zone ou changer de volume.

```
/xml/zone/runCommand.xml?zone=@zoneId&command=commandNumber
```

zoneId

L'identifiant d'une zone. Pour la liste des ID possibles, consultez la configuration Web sous `Automation /trivum API` ou consultez ci-dessous l'exemple getAll.xml.

Certains actionneurs peuvent ne pas adresser la première zone par @0 mais par @1 en raison de fichiers de configuration internes non utilisés. Pour résoudre ce problème, vous pouvez réinitialiser toute la configuration en : Système/Sauvegarder/Restaurer/Réinitialiser toutes les données de configuration

Au lieu de `@0`, le nom de la zone peut être donné. S'il contient des caractères spéciaux, réécrivez-les en utilisant `%` :

/xml/zone/runCommand.xml?zone=living%20room&command=…

numéro de commande multiple

Ceci est une commande numérique avec ces valeurs possibles:

```
ZONECMD_POWER_OFF             001
ZONECMD_MUTE                  002   toggle mute
ZONECMD_POWER_TOGGLE          006
ZONECMD_POWER_ON              007
ZONECMD_MUTE_ON               680
ZONECMD_MUTE_OFF              681
ZONECMD_VOLUME_INC            003
ZONECMD_VOLUME_DEC            004
ZONECMD_VOLUME_INC2           009
ZONECMD_VOLUME_DEC2           010
ZONECMD_VOLUME_INC5           011
ZONECMD_VOLUME_DEC5           012
ZONECMD_ALLOFF                015
ZONECMD_SNOOZE                017
ZONECMD_USE_PREV_SOURCE       029   see Zones / zone / KNX/HTTP sources
ZONECMD_JOIN                  030
ZONECMD_UNJOIN                031
ZONECMD_USE_NEXT_SOURCE       041   see Zones / zone / KNX/HTTP sources
ZONECMD_DEFAULT_LINEIN        048   play default line input of zone
ZONECMD_DEFAULT_STREAMING     050   play default streamer of zone
ZONECMD_DEFAULT_TUNER         051   play default tuner of zone
```

```
ZONECMD_VOLUME_DEC_1          080
ZONECMD_VOLUME_DEC_10         089
ZONECMD_VOLUME_INC_1          090
ZONECMD_VOLUME_INC_10         099
ZONECMD_START_MACRO_1         200
ZONECMD_START_MACRO_100       299
MULTIKEY_BASIC_FORWARD        400   skip to next  track, preset
MULTIKEY_BASIC_BACKWARD       401   skip to prev. track, preset
MULTIKEY_BASIC_FASTFORWARD    402
MULTIKEY_BASIC_FASTBACKWARD   403
MULTIKEY_BASIC_PLAYPAUSE      406
MULTIKEY_PLAY                 431
MULTIKEY_PAUSE                432
MULTIKEY_STOP                 433
MULTIKEY_STATION_DOWN         490
MULTIKEY_STATION_UP           491
MULTIKEY_NEXT_ALBUM           493
MULTIKEY_PREVIOUS_ALBUM       494
MULTIKEY_NEXT_PLAYLIST        495
MULTIKEY_PREVIOUS_PLAYLIST    496
ZONECMD_START_PAGING_1        500   (use 500 + paging id)
ZONECMD_START_PAGING_32       531
ZONECMD_STOP_PAGING_1         550
ZONECMD_STOP_PAGING_32        581
ZONECMD_STOP_PAGING_ALL       599
ZONECMD_PRESET_1              600
ZONECMD_PRESET_7              606
ZONECMD_GROUP_START_1         621
ZONECMD_GROUP_START_8         628
ZONECMD_GROUP_STOP            630
ZONECMD_GROUP_STOP_1          631
ZONECMD_GROUP_STOP_8          638
ZONECMD_GROUP_STOP_ALL        639
ZONECMD_STREAMING_NOPLAY      641
ZONECMD_VOLUME_00             900
ZONECMD_VOLUME_99             999
ZONECMD_ROOM_VOLUME_00       1000
ZONECMD_ROOM_VOLUME_99       1099
```

Exemples

appeler	fonction
`/xml/zone/getAll.xml`	Répertorier tous les ID de zone possibles.
`/xml/zone/get.xml?zone=@0`	Obtenir le statut d'une seule zone. Les paramètres facultatifs sont : &addSourceBasicData &addSourceStatusData
`/xml/zone/getSelection.xml?grouped`	Liste les zones avec les informations complètes du groupe.

appeler	fonction
/xml/zone/runCommand.xml?zone=@0&command=50	Passez la première zone au streaming par défaut.
/xml/zone/runCommand.xml?zone=@0&command=1	Éteignez la première zone.
/xml/zone/runCommand.xml?zone=@0&command=15	Désactiver toutes les zones.
/xml/zone/runCommand.xml?zone=@0&command=680	Désactiver le son
/xml/zone/runCommand.xml?zone=@0&command=681	Désactiver le son

1.2. Définir la source de la zone

Sélectionnez une source de zone par nom abrégé

/xml/zone/set.xml?zone=@0&source=@shortSourceName

shortSourceName

texte	agir	remarque
a a1 a3	première entrée analogique première entrée analogique troisième entrée analogique	Selon le modèle d'appareil, 0 à 8 Les entrées analogiques sont prises en charge. 10231} f2
premier trivum favori deuxième trivum favori		y y2
première liste de lecture trivum deuxième liste de lecture trivum		i i2
premier préréglage de réglage deuxième préréglage de réglage		s
source de flux par défaut de la zone	lecture sélection récente	t
tuner FM par défaut de la zone	lecture fréquence récente	l

Exemples

Appel API	remarque
/xml/zone/set.xml?zone=@0&source=@a1	passer à la première entrée analogique
/xml/zone/set.xml?zone=@0&source=@t	passer au tuner FM par défaut de la zone et lire la fréquence récente

`/xml/zone/set.xml?zone=@0&source=@p3`	basculer vers le tuner FM par défaut de la zone et écouter la station préréglée 3
`/xml/zone/set.xml?zone=@0&source=@f2`	passer au streaming par défaut de la zone et jouer trivum favori 2
`/xml/zone/set.xml?zone=@0&source=@i5`	passer au streaming par défaut de la zone et lire le préréglage de la webradio TuneIn 5
`/xml/zone/set.xml?zone=@1@source=@n`	C4 uniquement : utiliser une source par emplacement de carte n. (n >= 0)
`/xml/zone/runCommand.xml?zone=@0&command=15`	Désactiver toutes les zones.

1.3. Définir l'attribut de zone

Changer les valeurs de base dans une zone, comme le volume, la sourdine, la balance ou la basse.

Appel API	remarque
`/xml/zone/set.xml?zone=@0&volume=10`	régler le volume (0 ... 100)
`/xml/zone/set.xml?zone=@0&action=1`	identique à `/xml/zone/runCommand.xml` pour exécuter une commande numérique, dans ce cas `ZONECMD_POWER_OFF (1)`
`/xml/zone/set.xml?zone=@0&balance=0`	régler la balance, de -15 (complètement à gauche) à 15 (complètement à droite)
`/xml/zone/set.xml?zone=@0&bass=-5`	définir la réduction ou l'amélioration des basses, de -15 à 15
`/xml/zone/set.xml?zone=@0&treble=5`	régler la réduction ou l'amélioration des aigus, de -15 à 15

1.4. trivum Favoris

Pour créer des favoris trivums:

- lire du contenu musical, comme un album NAS
- puis sélectionnez ⋯ en haut à droite
- puis sélectionnez "Ajouter aux favoris trivum".

Obtenez la liste des favoris de trivum:

`/api/v1/trivum/favorite.xml`

Jouez à un favori trivum:

`/xml/zone/set.xml?source=@f1&zone=@0`

Vous pouvez également ajouter des options:

options	remarque

`&sequence=random-sequential`	sélectionner une piste de démarrage aléatoire
`&séquence=aléatoire-aléatoire`	jouer dans un ordre aléatoire permanent

1.5. Playlists trivum

Obtenez la liste des listes de lecture trivum:

`/api/v1/trivum/playlist.xml`

Lire une playlist trivum:

`/xml/zone/set.xml?source=@y1&zone=@0`

Vous pouvez également ajouter des options:

options	remarque
`&sequence=random-sequential`	pour commencer à une piste aléatoire
`&séquence=aléatoire-aléatoire`	pour lire uniquement des pistes aléatoires

1.6. TuneIn Favoris

Ceux-ci peuvent également être créés par ⋯ en haut à droite pendant la lecture d'une station TuneIn.

Obtenez la liste des favoris TuneIn:

`/api/v1/tunein/favorite.xml`

Jouez un favori TuneIn:

`/xml/zone/set.xml?source=@i1&zone=@0`

1.7. Presets FM

Liste des préréglages FM:

`/xml/system/getTunerStationList.xml`

Sur C4, cela affiche la liste des préréglages FM à l'échelle du système, mais aucun préréglage local stocké par carte tuner FM.

1.8. Statut et contrôle du NAS

Appel API	remarque
`/xml/system/getMusicCenterStatus.xml`	obtenir le statut de la bibliothèque NAS
`/xml/system/scanMusicCenterShares.xml`	relancer l'analyse complète du NAS

1.9. Direction du groupe

Les groupes peuvent être créés, modifiés ou supprimés par un appel:

```
/xml/zone/createGroup.xml?zone=zVisu&oldgroup=zMaster&members=++----------
```

Paramètres:

nom	remarque
zVisu	index de la zone courante de la visualisation client
zMaster	index du maître de groupe dont la musique doit être utilisée (si les deux zones diffusent actuellement des sources différentes)
+/-	caractères indiquant graphiquement quelles zones doivent faire partie d'un groupe. par exemple, avec un système à 4 zones, tapez 4 caractères ou moins (se remplit automatiquement avec -).

Exemple : la deuxième zone rejoint la lecture de la première zone

- la première zone lit un flux, la deuxième zone lit le tuner FM, toutes les autres zones sont éteintes.
- la deuxième zone doit être ajoutée à un groupe avec la première zone,
  et elle doit reprendre la musique de la première zone (le flux).

`/xml/zone/createGroup.xml?zone=1&oldgroup=0&members=++--`

Résultat: la deuxième zone commence à jouer le même flux que la première zone.

Exemple : la première zone rejoint la lecture de la deuxième zone

- la première zone lit un flux, la deuxième zone lit le tuner FM, toutes les autres zones sont éteintes.
- la première zone doit être ajoutée à un groupe avec la deuxième zone, et elle doit reprendre la musique de la deuxième zone (le tuner).

`/xml/zone/createGroup.xml?zone=0&oldgroup=1&members=++--`

Résultat: la première zone commence à jouer le même tuner FM que la deuxième zone.

Cela signifie que si les deux zones diffusent des sources différentes,
"oldgroup" décide de la musique à diffuser une fois le groupe rejoint.

Exemple: la deuxième zone doit quitter le groupe

`/xml/zone/createGroup.xml?zone=0&oldgroup=0&members=+---`

Le changement de + à - dans la liste des membres est pertinent.

Modifier le niveau de volume au sein d'un groupe

Au sein d'un groupe, les zones n'utilisent normalement pas de niveaux de volume isolés, mais un changement de volume affecte tous les membres du groupe.
Cette interdépendance est gérée par l'appel :

`/xml/zone/setVolume.xml`

Par défaut, cet appel ne se contentera pas de **définir** un niveau de volume absolu, mais il **avance un peu** dans la direction d'un volume cible donné. Il est préférable de l'utiliser avec un bouton + ou - dans votre visualisation.

Appel API	remarque
`/xml/zone/setVolume.xml?id=@0&volume=0`	Diminuez le volume du groupe pour l'ensemble du groupe. id est n'importe quel ID de zone du groupe. Le volume de tous les membres de la zone sera diminué de quelques pas.
`/xml/zone/setVolume.xml?id=@0&volume=99`	Augmentez le volume du groupe pour l'ensemble du groupe. Le volume de tous les membres de la zone augmentera de quelques niveaux.
`/xml/zone/setVolume.xml?id=@0&groupMemberVolume=99`	Augmentez le volume d'une seule zone par étape, sans affecter les autres membres du groupe.
`/xml/zone/setVolume.xml?id=@0&groupMemberVolume=0`	Diminue le volume d'une seule zone par étape, sans affecter les autres membres du groupe.
`/xml/zone/setVolume.xml?id=@0&stop`	Arrêtez immédiatement l'augmentation du volume.
`/xml/zone/setVolume.xml?id=@0&groupMemberVolume=50&absolute`	Réglez le volume **absolu** pour une seule zone, isolée des autres membres du groupe. (À utiliser avec précaution.)

Pour obtenir les nouvelles informations sur le niveau de volume au sein d'un groupe, effectuez un appel getChanges et consultez la liste d'état du volume.

`/xml/zone/getChanges.xml?zone=@0&visuid=90&apiLevel=2&now`

Exemple de sortie, si groupé, sous zone/statut:

```
<zone>
 ...
 <status>
 <volume>17</volume>  - volume of zone making the getChanges call
 ...
 <group>
```

```
  <zone>0</zone>
  <volume>17</volume> - volume for zone id 0
 </group>
 <group>
  <zone>1</zone>
  <volume>26</volume> - volume for zone id 1
 </group>
 <group>
   ...
 </group>
 <groupMembers>2</groupMembers>
 </status>
</zone>
```

Pour une explication complète de getChanges, voir Obtenir le statut de la zone.

1.10. Pagination

Les paginations doivent être configurées dans la configuration Web. Ensuite, les appels suivants peuvent être utilisés :

Lancer la pagination

`/xml/paging/start.xml`

Paramètres

nom	description
`id`	ID de radiomessagerie, 0 - 31
`volume`	facultatif, 5 - 100. s'il n'est pas fourni, le niveau de volume d'appel configuré est utilisé.
`autostoptime`	en option, 5 - 100 secondes. si non fourni, les paramètres d'arrêt configurés sont utilisés.

Exemple

`/xml/paging/start.xml?id=0&volume=10&autostoptime=10`

Une pagination s'arrête automatiquement après le temps défini, mais vous pouvez l'arrêter plus tôt en appelant :

`/xml/paging/stop.xml?id=0`

2. Sélection musicale interactive

Commence avec:

`/xml/system/getWebTouchMenu.xml?which=music&zone=@0&visuid=90`

Cela produit des enregistrements comme :

```
<row>
  <type>action</type>
  <mode>menu</mode>
  <action>/xml/system/getWebTouchMenu.xml?which=trivumFavorites&amp;keypad=4</action>
  <icon>/imgs/visuIconServiceFavorites_128px.png</icon>
  <text>trivum_20favorites</text>
</row>
```

puis, par enregistrement:

- décoder et afficher le champ de texte dans votre visualisation.
  `_20` signifie un caractère avec le code Ascii 0x20 (un espace).
- Si vous appuyez sur cette touche, appelez l'URL d'action et affichez le niveau de menu suivant.

Ne comptez pas sur la disponibilité permanente de niveaux de menu spécifiques. En particulier, les menus fournis par les services de musique peuvent changer avec le temps.

3. Obtenir le statut de la zone

3.1. Synchrone

Interrogez l'état d'une zone avec un court appel d'API :

`/xml/zone/getChanges.xml?zone=@0&visuid=90&apiLevel=2&now`

Paramètres

nom	fonction
`visuid`	un nombre de 1 à 99 pour identifier votre instance de visualisation externe. dans ce document d'API, visuid=90 est utilisé pour les requêtes de test.
`apiLevel`	doit toujours être 2. cela produira des objets xml `button` sous `keypad/basic`.
`maintenant`	indique au serveur de renvoyer immédiatement le nouvel état de la zone et de fermer la connexion. sans `&now` l'appel bloquerait jusqu'à un timeout, ou jusqu'à un changement dans les informations d'état de la zone.

`reload=1`	si deux visualisations accèdent au même serveur avec le même visuid une erreur "utilisé deux fois" peut apparaître. dans ce cas, la visu la plus récente devrait ajouter `&reload=1` lors du premier appel, pour dire clairement qu'il s'agit de la visualisation la plus récente.

A propos des unités de contrôle (visualisations)

Si vous envoyez des requêtes avec `visuid=90`, un objet *Control Unit* avec l'ID 90 est créé sur le serveur.

Vous pouvez obtenir la liste des unités de contrôle actuelles dans la configuration Web, sous *Unités de contrôle*.

Après le premier accès, l'unité est répertoriée comme "Non configuré". Dès que vous modifiez sa configuration, par exemple en mettant l'option "Off par un appui court sur power", elle s'appelle *Configured*, et les nettoyages ultérieurs de la liste des Control Units ne supprimeront pas celle-ci.

S'il n'y a pas de demandes pour cette unité, après un certain temps, elle sera répertoriée sous "unités de contrôle actuellement inactives".

3.2. Asynchrone

Cela signifie qu'un appel HTTP ne reviendra pas immédiatement, mais il se bloquera jusqu'à ce que quelque chose change.

Exemple:

`/xml/zone/getChanges.xml?zone=@0&visuid=90&apiLevel=2`

Notez que `& now` est manquant. Ce qui suit se produira:

au premier appel API :

Une unité de contrôle avec ID 90 est créée et liée à la première zone.
L'appel d'API revient immédiatement, avec les données d'état complètes de la zone.

sur tous les autres appels d'API :

L'unité de contrôle 90 existante est réutilisée. L'appel d'API peut bloquer, jusqu'à :

- un timeout est atteint (10 secondes environ). dans ce cas, vous obtenez une réponse du type : `<rows><system><timeout>1</timeout>`
- ou jusqu'à ce que quelque chose change, par exemple le volume dans la zone.

si (de nombreuses) données d'état ont changé sur le serveur entre deux appels getChanges, l'appel peut ne pas bloquer du tout, mais renvoyer immédiatement le nouvel état.

lorsque vous recevez un délai d'expiration, réexécutez simplement getChanges immédiatement.

cela signifie que vous pouvez exécuter getChanges à l'infini, dans une boucle, par exemple dans un thread d'E/S séparé. Étant donné qu'une requête ne revient que sur les modifications, cela ne causera aucun problème de charge avec le serveur.

lorsque vous ne recevez pas de délai d'attente, c'est-à-dire que l'appel revient immédiatement ou après quelques secondes (dès que quelque chose a changé), traitez les données d'état, puis relancez la requête getChanges.

3.3. Annexe : exemple schématique d'une application client Visu

3.3.1. Application à un seul fil

Cela nécessite que vous puissiez tester, dans votre langage de programmation, si les données de réponse pour un socket existent (via l'appel select()).

Fil principal

- commencer : envoyer `/xml/zone/getChanges.xml?visuid=90&now`
- début de la boucle : mise à jour de l'interface graphique.
  - traiter les événements d'entrée de l'utilisateur.
  - envoyer des commandes synchrones comme :
    `/xml/zone/runCommand.xml?…`
    recevoir une réponse, vérifier rc ET traiter les données d'état xml
    (comme pour les réponses getChanges)
  - vérifier si les données de réponse existent pour l'appel getChanges en cours
    (en code C : appel select() sur le socket)
    SI les données existent à partir du serveur trivum :
    - Recherchez `<userdata name="rc">0</userdata>`.
      Si NON présent
      _ traiter l'erreur et attendre quelques secondes.
      Sinon si PAS un délai d'attente
      _ traiter la réponse xml (données d'état)
      Endif
      appel asynchrone (envoyer simplement)
      `/xml/zone/getChanges.xml&visuid=90&onlyChanges`
      Endif
  - si aucune donnée du serveur n'arrive dans la minute 1
    - appel asynchrone (il suffit d'envoyer)
      `/xml/zone/getChanges.xml&visuid=90&onlyChanges`
      endif
  - réexécuter la boucle

3.3.2. Exemple d'application à deux threads

Peut être utilisé si vous préférez exécuter le blocage des réceptions sur les sockets dans un thread d'E/S séparé.

Fil principal

- mettre à jour l'interface graphique.
- traiter les événements d'entrée de l'utilisateur.
- envoyer des commandes synchrones comme :
  `/xml/zone/runCommand.xml?…`
  recevoir une réponse, vérifier rc ET traiter les données d'état xml
  (comme pour les réponses getChanges)
- recevoir les données d'état et les erreurs de Status Thread.
- relancez cette boucle.

Fil d'état

- SI sur la première boucle :
  - envoyer `/xml/zone/getChanges.xml?visuid=90&now`
    SINON
  - envoyer `/xml/zone/getChanges.xml?visuid=90&onlyChanges`
- recevoir une réponse (ceci est bloqué jusqu'à 10 secondes)
- Recherchez `<userdata name="rc">0</userdata>`.
  S'il n'est PAS présent, il y a une erreur.
  Assurez-vous de ne pas simplement réexécuter la boucle en cas d'erreur,
  mais attendez au moins quelques secondes et informez le fil principal.
- Recherchez `<rows><system><timeout>1</timeout>`.
  SI ceci est présent
  - relancer la boucle immédiatement.
    AUTREMENT
  - traitez les données d'état de la réponse,
    et copiez les nouvelles données d'état dans Main Thread.
- relancez cette boucle.

4. Guide d'intégration simple

Une approche très simple, étape par étape, avec quelques exemples de données de réponse.

Si vous souhaitez intégrer trivum dans un serveur domotique, vous souhaitez normalement interroger deux choses :

4.1. État général de la zone

```
/xml/zone/getAll.xml
```

Par exemple, avec Crestron, il devrait être possible d'appeler ceci par le client TCP, avec un appel comme

```
SendString(tcpClient, "GET /xml/zone/getAll.xml HTTP/1.1\r\n\r\n");
```

Il produit un aperçu XML :

```
<rows>
  <zone>
    <class>zone</class>
    <id>0</id>
    <description>Room 1</description>
    <status>on</status>
    <volume>0</volume>
    <groupmaster>255</groupmaster>
  </zone>
  <zone>
    <class>zone</class>
    <id>1</id>
    <description>Room 2</description>
    <status>off</status>
    <volume>15</volume>
    <groupmaster>255</groupmaster>
  </zone>
  <zone>
    <class>zone</class>
    <id>2</id>
    <description>Room 3</description>
    <status>off</status>
    <volume>15</volume>
    <groupmaster>255</groupmaster>
  </zone>
  ...
```

Avec un client non compatible XML comme Crestron, le traitement peut ressembler à ceci (exemple non testé) :

```
STRING response[5000];
STRING zoneNames[4][20];
INTEGER pos, startPos, endPos, zoneCount;
zoneCount = 0;
pos = 1;
```

```
WHILE ((pos = Find("<description>", response, pos)) > 0) {
    startPos = pos + 13;
    endPos = Find("</description>", response, startPos);
    IF (endPos > startPos) {
        zoneNames[zoneCount] = Mid(response, startPos, endPos - startPos);
        zoneCount = zoneCount + 1;
    }
    pos = endPos + 14;
}
```

4.2. Statut de zone unique

```
/xml/zone/get.xml?zone=@0&addSourceBasicData&addSourceStatusData
```

produit par exemple :

```
<rows>
  <runtime>
    <class>zone</class>
    <id>0</id>
    <source>
      <description>SC344m_201008153.66</description>
      <status>
        <sourceStatusInfoCount>0</sourceStatusInfoCount>
        <streamStatus>5</streamStatus>
        <streamLength>0</streamLength>
        <streamPosition>331</streamPosition>
        <artist>Jazeek</artist>
        <album></album>
        <track>LV</track>
        <service>webradio</service>
        <imageURL>http://cdn-
profiles.tunein.com/s45087/images/logod.jpg?t=2</imageURL>
        <playerIP>127.0.0.1</playerIP>
        <playerPort>1333</playerPort>
        <info>
          <key>WebRadio</key>
          <value></value>
          <size>0</size>
        </info>
        <info1></info1>
        <infoPlayingSource>visuIconPlayingTuneIn_128px.png</infoPlayingSource>
        <info2>LV_20_2F_20Jazeek</info2>
        <info>
          <key>Artist</key>
          <value>Jazeek</value>
          <size>0</size>
        </info>
```

```
        <info>
          <key>Track</key>
          <value>LV</value>
          <size>0</size>
        </info>
        <info>
          <key>Status</key>
          <value>Playing 24.0 kHz</value>
          <size>1</size>
        </info>
        <info>
          <key>Service</key>
          <value>TuneIn</value>
          <size>1</size>
        </info>
        <info>
          <key>TuneIn favorite</key>
          <value>Favorite station</value>
          <size>1</size>
        </info>
        <infoCount>6</infoCount>
        <mostImportantInfo>0</mostImportantInfo>
        <infoLevel>0</infoLevel>
      </status>
    </source>
    <outputcount>1
    </outputcount>
    <output1>
      <description>SC344m_201008153.66</description>
    </output1>
    <status>on</status>
    <volume>0</volume>
  </runtime>
  ...
</rows>
```

4.3. Exécuter une commande

Une fois les identifiants de zone connus, jouez quelque chose comme dans les exemples suivants. Ils utilisent principalement un numéro de commande multiple comme indiqué plus haut dans ce document.

call	function
`/xml/zone/set.xml?zone=@0&source=@f1`	jouer le premier favori dans la première zone
`/xml/zone/runCommand.xml?zone=@0&command=48`	jouer la ligne d'entrée
`/xml/zone/runCommand.xml?zone=@0&command=9`	augmenter le volume de 2 étapes
`/xml/zone/runCommand.xml?zone=@0&command=10`	réduire le volume de 2 étapes

call	function
/xml/zone/runCommand.xml?zone=@0&command=6	activer/désactiver la zone
/xml/zone/runCommand.xml?zone=@0&command=2	activer/désactiver le son
/xml/zone/runCommand.xml?zone=@0&command=51	lire le tuner
/xml/zone/runCommand.xml?zone=@0&command=400	passer à la piste de liste de lecture suivante ou au préréglage du tuner
/xml/zone/runCommand.xml?zone=@0&command=401	passer à la piste de liste de lecture précédente ou au préréglage du tuner
/xml/zone/runCommand.xml?zone=@0&command=41	lire la première ou la suivante source KNX/HTTP de la zone. (définir jusqu'à 8 de ces sources dans la configuration.)
/xml/zone/runCommand.xml?zone=@0&command=29	lire la source KNX/HTTP précédente. (cela peut être n'importe quelle action, comme lire un favori, une entrée de ligne ou une station du tuner.)