Partner API - User Guide

September 09, 2015

Contents:

Introduction

The Voxy API enables partners to programmatically manage data and services in the Voxy platform and to query the platform for data.

This document provides an overview of how to use the API and includes examples. The *API Reference Guide* provides specific information about each API endpoint.

The API was designed using the principles of REST.

Common values used in API calls:

- external_id : A unique ID for the student in the partner's system that is used by Voxy to associate the user created in Voxy. We recommend that this ID be something similar to a student registration number. This ID must be unique to the partner and be passed for all API calls that relate to that user

- HTTP header AUTHORIZATION: A value made by concatenating the values of two variables ("Voxy $API_KEY:$REQUEST_SIGNATURE") where:

    - $API_KEY is the unique API_KEY sent to each partner by e-mail (used to identify the institution in Voxy)

    - $REQUEST_SIGNATURE is a hash sha256

        - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

The URLs for the API:

- Testing environment: https://web-stage.voxy.com/partner_api/
- Production environment: https://www.voxy.com/partner_api/

The URLs for the web application for students:

- Testing environment: https://web-stage.voxy.com/
- Production environment: https://www.voxy.com/

The examples in this document are written in Python running on Django.

Simple flow

- For new users in Voxy:

    1) Create the user in Voxy with an expiration date (POST /partners/users/${external_id})
    2) Retrieve the authentication token (SSO) (GET /partners/users /${external_id}/auth_token)
    3) Redirect the user to response (SSO) → ['actions'] ['start']

- For users already registered in Voxy:
    1) Retrieve the authentication token (SSO) (GET /partners/users/${external_id}/auth_token)
    2) Redirect the user to response (SSO) → ['actions'] ['start']

- Check whether a user is already registered in Voxy or if the user's account has expired (GET /partners/users/external_id}/auth_token)
    - If the user is registered and has a valid account expiration date (the account has not expired), the response will be:

```
HTTP_STATUS=200
{
   'auth_token':'eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0_T',
   'actions':{

'start':'https://web-stage.voxy.com/u?auth_token=eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0
_T'
 }
```

    - If the user is registered but his/her account has expired account (account expiration date <today's date), the response will be:

```
HTTP_STATUS = 403
{"error_message": "Access for the user with the id = '{id}' expired expired_date on {} "}
```

    - If the user has not been registered, the return will be:

```
HTTP_STATUS = 403
{" error_message ": " user does not exist "}
```

Detailed flow

1) Check if user is already registered in Voxy or the account is expired (GET /partners/users/${external_id}/auth_token)

2) If the user has not been registered:

```
HTTP_STATUS = 403 {"error_message": "user does not exist"}
```

   a) Create the user in Voxy (POST /partners/users/${external_id})
      i) Setting the account expiration date is optional using the expiration_date parameter
   b) The expiration date of the account can be set or changed (PUT /partners/users/${external_id})
   c) Retrieve the authentication token (SSO) (GET /partners/users/${external_id}/auth_token)
   d) Redirect user to response → ['actions'] ['start']

3) If user is registered and not expired (date of account expiration > today):

```
HTTP_STATUS = 200
```

   a) Rertieve the authentication token (SSO) (GET /partners/users/${external_id}/auth_token)
   b) Redirect user to ['actions'] ['start']

4) If user is registered, but expired (account expiration date <today's date):

```
HTTP_STATUS = 403 {"error_message": "Access for the user with the id = '{id}' expired on expired_date {}"}
```

   a) Display a message informing the user that the account has expired

How to calculate the value of the header AUTHORIZATION

POST / PUT / GET / DELETE

A header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"`

where:

- $API_KEY is the unique API_KEY sent to each partner by e-mail (used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256
  - $API_SECRET sent to the partner concatenated with the url_encode of all the parameter values of the BODY and/or QUERY_STRING_PARAMs ordered alphabetically by parameter name

Below is an example of the method for calculating the header $REQUEST_SIGNATURE in Python 3:

```
from hashlib import sha256
from urllib.parse import urlencode

# partner information:
API_KEY ='yourapikey'
API_SECRET = 'yourapisecret'

# # utility methods:
def get_header(response_data):
    return 'Voxy {key}:{request_signature}'.format(
        key=API_KEY,
        request_signature=generate_signature(response_data)
    )

def generate_signature(response_data):
    to_hash = urlencode(sorted(response_data.items()))
    before_hash = API_SECRET + to_hash
    hashed = sha256(before_hash.encode('utf-8'))
    return hashed.hexdigest()

#all the parameters being passed in the body of the request in question:
body_params = {'first_name': 'John', 'email_address': 'john@university.com'}

headers = {}
headers ['AUTHORIZATION'] = get_header(body_params)
```

resulting in:

```
{'AUTHORIZATION' 'Voxy $API_KEY:$REQUEST_SIGNATURE'}
{'AUTHORIZATION' 'Voxy yourapikey:f5de859dd563a4868a83883e3acc24ebaaa000e727bbdb5eeaf9cf04e4217974'}
```

Example 1: Creating a user with an expiration date

POST /partners/users /${external_id} - Passing optional parameters

HTTP_METHOD:
`POST`
URL:
https://web-stage.voxy.com/partner_api/partners/users/123456
HEADERS:
```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
}
```
BODY:
```
{'email_address': 'aluno.sobrenome@universidade.br', 'expiration_date': '2015-12-31', 'first_name': 'Aluno', 'native_language' : 'pt'}
```

RETURN: `HTTP_STATUS=201 (created)`

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and is used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of:
  - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python to request up:

```
from hashlib import sha256
from django.utils.http import urlencode


api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {'email_address':'aluno.sobrenome@universidade.br', 'expiration_date': '2015-12-31', 'first_name': 'Aluno', 'native_language':'pt'}


def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected  $to_hash: 'email_address=aluno.sobrenome%40universidade.br&expiration_date=2015-12-31&first_name=Aluno&native_language=pt'
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()


#SIGNATURE_TOKEN expected for the example: '205a00943ca3428245fc3638b4022f43a5d79041d663335183800e9c0b650ab9'
```

Example 2: Creating a user without an expiration date

POST /partners/users/${external_id}
Passing only required parameters

HTTP_METHOD:
`POST`
URL:
https://web-stage.voxy.com/partner_api/partners/users/123456
HEADERS:
```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
}
```
BODY:
```
{'email_address': 'aluno.sobrenome@universidade.br', 'first_name': 'Aluno', 'native_language': 'pt'}
```

RETURN: `HTTP_STATUS=201 (created)`

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
  - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for request above:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {'email_address': 'aluno.sobrenome@universidade.br', 'first_name': 'Aluno', 'native_language': 'pt'}
```

```
def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: 'email_address=aluno.sobrenome%40universidade.br&first_name=Aluno&native_language=pt'
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()

#SIGNATURE_TOKEN expected for the exemple: 'a69703678f626c6b82c0fa37e5cd850716e23af9da5c2da1e74755d70e46ec0c'
```

Example 3: Changing / setting account expiration date

PUT /partners/users/${external_id}

HTTP_METHOD:

PUT

URL:

https://web-stage.voxy.com/partner_api/partners/users/123456

HEADERS:

```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
}
```

BODY:

```
{'expiration_date': '2015-12-31'}
```

RETURN: HTTP_STATUS=200

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
  - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for the request above:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {'expiration_date': '2015-12-31'}

def generate_signature(api_secret, body_params):
  to_hash = urlencode(sorted(body_params.items()))
  #expected for $to_hash: 'expiration_date=2015-12-31'
  hashed = sha256(api_secret + to_hash)
  return hashed.hexdigest()

#SIGNATURE_TOKEN expected for example: 'b1ec2a4743ac134002c35f5db13304e1d6178ad00b2407d9860c3b3e6ac11fd0'
```

Example 4: Fetching the login URL and authentication token

(SSO) GET /partners/users/${external_id}/auth_token

For existing users

HTTP_METHOD:

GET

URL:

https://web-stage.voxy.com/partner_api/partners/users/123456/auth_token

HEADERS:

```
 { 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
 }
```

BODY:

```
{}
```

RETURN: HTTP_STATUS=200

```
{
   'auth_token':'eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0_T',
   'actions':{
 'start':'https://web-stage.voxy.com/u?auth_token=eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0_T'
   }
}
```

The user is automatically logged into the platform Voxy by visiting the URL ['actions'] ['start']:

https://web-stage.voxy.com/u?auth_token=mOhbHQZle4fDz86ORGfmNwVsmkn7XSggvF_QO2ZEXQY645YMgjFik4ePakExOscF0Z5fQbty

** It is mportant to mention that auth_token can be used only once. This means that the partner will have to make a request for a new auth_token using this call whenever a user is logging into Voxy. Therefore, this URL should not be stored in a database.

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python to above request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}


def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()


#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 5: Verifying that the is already registered

GET /partners/users/${external_id}/auth_token

HTTP_METHOD:
`GET`
URL:
`https://web-stage.voxy.com/partner_api/partners/users/99999/auth_token`
HEADERS:
`{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}`
BODY:
`{}`

- If the user was already registered and has valid account expiration date (not expired) the return will be:

```
RETURN: HTTP_STATUS=200
{
   'auth_token':'eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0_T',
   'actions':{
 'start':'https://web-stage.voxy.com/u?auth_token=eTJeh7FLE4zkVAvOO_VQ0yoCUHriNnJtHZLiGxg0sY8VeuZtmOdNADZaEeCNRgkHjP3cf0_T'
 }
```

- If the user is already registered but has an expired account (account expiration date < today's date), the return will be:

```
RETURN: HTTP_STATUS=403
{"error_message": “Access for the user with the id='{id}' expired on {expired_date}”}
```

- If the user has not been registered, the return will be:

```
RETURN: HTTP_STATUS=403
{"error_message": "user does not exist"}
```

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}


def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
```

```
 #expected $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example:  '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 6: Changing account information

PUT /partners/users/${external_id}

User parameters that can be updated:

- first_name
- expiration_date
- native_language
- Phone_number
- email

Example 7: Adding credits for private lessons (tutoring) to a user account

POST /partners/users/${external_id}/entitlements

HTTP_METHOD:
POST
URL:
https://web-stage.voxy.com/partner_api/partners/users/${external_id}/entitlements
HEADERS:

```
 { 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
 }
```

BODY:
{'credits': 5}

RETURN: HTTP_STATUS=200

Note that credits is an integer data type.

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"`
where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
  - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for this request:

```
from hashlib import sha256
from django.utils.http import urlencode

api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {'credits': 5}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   print to_hash
   #expected for $to_hash: 'credits=5'
   hashed = sha256(api_secret + to_hash)
   print hashed.hexdigest()
   return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '0037223c823f379c40830a1581914cb18e35efb43c3c359a7ca53971fff81f8e'
```

Example 8: Creating segments

(POST /partners/segments/

Segments can be used to group users for reporting purposes. Examples: classes, levels, campus, plans, courses...

HTTP_METHOD:
POST
URL:
https://web-stage.voxy.com/partner_api/partners/segments/

HEADERS:
```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
}
```
BODY:
{'label':'nome-unico-do-segmento'}

RETURN: HTTP_STATUS=201 (created)

Note that labels (segment names) cannot contain special characters or accents.

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"`
where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of $REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {'label':'nome-unico-do-segmento'}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected for $to_hash: 'label=nome-unico-do-segmento'
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '083e019f649678162d5df14941ae7c96fe8ed1f3b333f38a73a6ee0432d60d38'
```

Example 9: Adding a user to a segment

POST /partners/segments/${segment_label}/users/${external_user_id}

HTTP_METHOD:
`POST`
URL:
https://web-stage.voxy.com/partner_api/partners/segments/nome-unico-do-segmento/users/123456
HEADERS:
```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'
}
```
BODY:
`{}`

RETURN: `HTTP_STATUS = 201 (created)`

In this case:

- ${segment_label} must be a valid segment already registered on Voxy through POST /partners/segments/
    - In this example: nome-unico-do-segmento
- ${external_user_id} must be a valid user already registered in Voxy through POST /partners/users/${external_id}
    - In this exemplo: 123456

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"`
where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation for the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```

```
#SIGNATURE_TOKEN expected for this exemplo: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 10: Adding a user to a segment on demand

If the segment does not already exist, it is automatically created and the user is added
POST /partners/segments/${segment_label}/users/${external_user_id}

HTTP_METHOD:
`POST`
URL:
https://web-stage.voxy.com/partner_api/partners/segments/nome-novo-segmento/users/123456
HEADERS:
`{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'`
BODY:
`{}`

RETURN: `HTTP_STATUS=201 (created)`

In this case:

- ${segment_label} does not need to be a previously created segment
- ${external_user_id} must be a valid user previously registered through POST /partners/users/${external_id}
    - In this example: 123456

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"`
where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation method of the $ REQUEST_SIGNATURE in Python for the request

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected $to_hash: ''
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()

#SIGNATURE_TOKEN expeted for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 11: Listing all segments already registered

GET /partners/segments/

HTTP_METHOD:
`GET`
URL:
https://web-stage.voxy.com/partner_api/partners/segments/
HEADERS:
`{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'`
`}`
BODY:
`{}`

RETURN: `HTTP_STATUS=200`

```
[
{"label": "nome-unico-do-segmento", "user_ids": [123456, 99999]},
{"label": "nome-unico-do-segmento-2", "user_ids": []}
]
```

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 12: Removing a User From a Segment

DELETE /partners/segments/${segment_label}/users/${external_user_id}

HTTP_METHOD:
`DELETE`
URL:
https://web-stage.voxy.com/partner_api/partners/segments/nome-unico-do-segmento/users/123456
HEADERS:
`{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}`
BODY:
`{}`

RETURN: `HTTP_STATUS=200`

In this case:

- ${segment_label} must be a valid segment already registered in Voxy
    - In this example: nome-unico-do-segmento
- ${external_user_id} must be a valid user already registered in Voxy
    - In this example: 123456

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for the request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected $to_hash: ''
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 13: Listing All Users Who Belong to a Segment

GET /partners/segments/${segment_label}

HTTP_METHOD:
GET
URL:
https://web-stage.voxy.com/partner_api/partners/segments/nome-unico-do-segmento
HEADERS:
```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}
```
BODY:
```
{}
```

RETURN: HTTP_STATUS=200
```
{"label": "nome-unico-do-segmento", "user_ids": [99999]},
```

In this case, the segment must be a valid segment previously created.

Recall that the header AUTHORIZATION is a combination of strings in the following format "Voxy $API_KEY:$REQUEST_SIGNATURE" where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected $to_hash: ''
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 14: List All Segments That a User Belongs To

GET /partners/users/${external_id}

HTTP_METHOD:
GET
URL:
https://web-stage.voxy.com/partner_api/partners/users/123456/
HEADERS:
```
{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}
```
BODY:
```
{}
```

RETURN: HTTP_STATUS=200
```
{
   "segments":[
      "segmento1", "segmento2"
   ]
}
```

Recall that the header AUTHORIZATION is a combination of strings in the following format "Voxy $API_KEY:$REQUEST_SIGNATURE" where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
   to_hash = urlencode(sorted(body_params.items()))
   #expected $to_hash: ''
```

```
   hashed = sha256(api_secret + to_hash)
   return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 15: Retrieving a User's Account Information

GET /partners/users/${external_id}

HTTP_METHOD:

```
GET
```

URL:

```
https://web-stage.voxy.com/partner_api/partners/users/123456/
```

HEADERS:

```
{ 'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}
```

BODY:

```
{}
```

RETURN: HTTP_STATUS=200

```
{
   "first_name":"Bob",
   "email_address":"bob@example.com"
   "native_language":"pt",
   "level":5,
   "expiration_date":"2016-01-01",
   "tutoring_credits":99,
   "segments":[
      "segmento1", "segmento2"
   ],
   "phone_number":"+5511912349876",
}
```

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}


def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()

#SIGNATURE_TOKEN expected for this example: '5c31e5c0145780b8b7534aafa8a23545ca61acd5ea6b6199e8bed34481e5d11e'
```

Example 16: Retrieving an individual user's current and historical unit progress and performance data

GET /partners/users/${external_id}/units

HTTP_METHOD:

```
GET
```

URL:

```
https://web-stage.voxy.com/partner_api/partners/users/123456/units
```

HEADERS:

```
{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}
```

BODY:

```
{}
```

RETURN: HTTP_STATUS=200

see sample response here: https://gist.github.com/borekwa/07d3262b1a6f143d73e7

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}


def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```

Example 17: Retrieving an individual user's unit progress and performance data within specific date filters

GET /partners/users/${external_id}/units

HTTP_METHOD:

`GET`

URL:

`https://web-stage.voxy.com/partner_api/partners/users/123456/units`

HEADERS:

`{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}`

BODY:

```
{
    start_date: '2015-09-01',
    end_date: '2015-10-01'
}
```

RETURN: HTTP_STATUS=200

see sample response here: https://gist.github.com/borekwa/07d3262b1a6f143d73e7

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}


def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```

Example 18: Retrieving current and historical Unit progress and performance data for all users in a segment

GET /partners/segments/${segment_label}/users/units

HTTP_METHOD:

`GET`

URL:

`https://web-stage.voxy.com/partner_api/partners/segments/acme-center/users/units`

HEADERS:

`{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}`

BODY:

```
{
```

```
    start_date: ‘2015-09-01’,
    end_date: ‘2015-10-01’
}
```

RETURN: HTTP_STATUS=200
see sample response here: https://gist.github.com/borekwa/76d71b81fcd3f53e681c

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```

Example 19: Retrieving group tutoring sessions for a segment (supports optional start_date and end_date filters)

GET /partners/segments/${segment_label}/users/group_sessions

HTTP_METHOD:
`GET`
URL:
`https://web-stage.voxy.com/partner_api/segments/${segment_label}/users/group_sessions`
HEADERS:
`{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}`
BODY:

```
{
    start_date: ‘2015-09-01’,
    end_date: ‘2015-10-01’
}
```

RETURN: HTTP_STATUS=200
see sample response here: https://gist.github.com/borekwa/1dcf88fc61e5ea000895

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```

Example 20: Retrieving attendance and feedback data for an individual group tutoring session

GET /partners/group_sessions/${group_session_id}/feedback

HTTP_METHOD:

```
GET
```

URL:

```
https://web-stage.voxy.com/partner_api/segments/${segment_label}/users/group_sessions
```

HEADERS:

```
{'AUTHORIZATION': 'Voxy ${API_KEY}:${REQUEST_SIGNATURE}'}
```

BODY:

```
{}
```

RETURN: HTTP_STATUS=200

```
see sample response here:
```

Recall that the header `AUTHORIZATION` is a combination of strings in the following format `"Voxy $API_KEY:$REQUEST_SIGNATURE"` where:

- $API_KEY is the API_KEY sent to the partner by e-mail (and used to identify the institution in Voxy)
- $REQUEST_SIGNATURE is a hash sha256 of
    - $API_SECRET sent to the partner concatenated with the url_encode of all the key value pairs of the request data ordered alphabetically by key name

Here is an example of the calculation of the $ REQUEST_SIGNATURE in Python for this request:

```
api_secret = 'Mvp1co0erZK8U8sEbF6IqE54'
body_params = {}

def generate_signature(api_secret, body_params):
 to_hash = urlencode(sorted(body_params.items()))
 #expected for $to_hash: ''
 hashed = sha256(api_secret + to_hash)
 return hashed.hexdigest()
```