

SEMATIME 4.0 API DOCUMENTATION

INTRODUCTION

This is a very simple API that allows organizations to integrate their Sematime accounts with other systems such as an ERP software. The API exposes the following functions:

- a. Sending a simple message
- b. Sending multiple unique messages (REST only)
- c. Getting detailed delivery reports
- d. Scheduling a message for later sending
- e. Sending exam results (for schools)
- f. Sending fee balances (for school)
- g. Fetching your account info

The API is REST based and JSON is used as the default and only data exchange format. We make use of both HTTP POST & HTTP GET methods to mediate between the servers. We also understand that not everyone is conversant with how REST APIs work so we have provided a way of using the API by passing HTTP GET URL parameters. For example, to send a message using URL parameters.

<https://apis.sematime.com/v1/1452764545xxx/messages/single.url?text=The quick brown dog jumped over the lazy boy.&recipients=0768028101&AuthToken=1ad63cf06775416ba1bb413a51694xxx>

PREREQUISITES

These are the things you will need before you can start integrating with our API.

1. Developer Key

This is a unique code that is shared with the organization you wish to integrate with for them to grant you API access to their account. If you don't have one already, go to <https://apis.sematime.com/developers/> and signup. Your key will be sent via email.

2. AuthToken

This is a unique code that will be emailed to you once the organization has granted you API access. It will be sent to your developer email address together with account ID.

3. Account ID

This is a code that helps us identify each and every organization you have integrated with. It will be sent to you via email together with the AuthToken.

We have also prepared a Postman collection of some of the functions of the API available here <https://drive.google.com/open?id=15PCwqpF7pGttOOowRM8WX2FwrExphgPN>

AUTHORIZATION

This is handled via simple token based authorization. Every request to the API must include the authorization token.

- i. For developers using the REST architecture, your authorization token should be passed as a HTTP header called 'AuthToken'
- ii. For developers using URL parameters, your authorization token should be passed as a query parameter called 'AuthToken'

Base URL

All resources referenced in the API have the following base URL: <https://apis.sematime.com/v1/{accountId}>

PAGINATION

Some GET requests can generate responses that contain lots of records. We provide the option of limiting response sizes through the use of two URL parameters:

- a. **rowCount** – indicates the number of records you want to retrieve. Defaults to 20
- b. **lastOffset** – indicates the index of the row where you would wish to start the retrieval. This value is always specified by the API server. Defaults to 0(zero)

API RESPONSE CODES

Remember to check the HTTP status code returned to know if your request was successful or not. HTTP code 200(OK) means you were successful anything else means there was an error and that you should check the accompanying description to know the infringement.

Status Code	Name	Description
200	OK	Description
400	Bad Request	Improperly formatted request.
401	Unauthorized	Request could not be authenticated due to incorrect login credentials or auth token.
403	Forbidden	Request was accepted but could not be processed. For example, when you do not have enough sms units to send a text
404	Not Found.	Resource you are requesting could not be found.
500	Internal Server Error	Server is unable to process your request possibly due to a problem on our side.

SENDING A SIMPLE MESSAGE

Sending a message is pretty simple and straight forward. We only need you to send us the following parameters:

Parameter	Data Type	Constraint	Description
message	String	required	The message you want to send.
recipients	String	required	a comma separated string that contains phone numbers. It can be formatted as follows, "0736189xxx, John<0706129xxx>"

1. Using REST

Make a HTTP POST request to <https://apis.semotime.com/v1/{accountId}/messages/single> where {accountId} is the ID of the account you received during API authorization. The server expects:

a. Headers

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.

- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'application/json'
- b. **POST request body**

Your POST request body should be a JSON string containing the details of the message to send. For example,

```
{
  "text": "The big brown fox jumps over the lazy dog.",
  "recipients": "0706129xxx,James<0737189xxx>"
}
```

2. Using URL parameters

You only need to make a HTTP GET call to <https://apis.semotime.com/v1/{accountId}/messages/single.url> and pass the AuthToken, message and recipients as query parameters. For example,

<https://apis.semotime.com/v1/145276454xxx/messages/single.url?text=The quick brown dog jumped over the lazy boy&recipients=0768028101&AuthToken=1ad63cf06775416ba1bb413a51694xxx>

Regardless of the method used to send the message, you will get back a response that looks like the following if your message was successful.

```
{
  "signature": "Call:0706129100\n",
  "sender": {},
  "pending": 1,
  "totalSmsCount": 1,
  "delivered": 0,
  "smsType": "general",
  "engagements": 0,
  "id": 1562314576226,
  "text": "The quick brown dog jumped over the lazy boy.",
  "failed": 0
}
```

If your request was not successful, you will get a response containing the error message. For example,

```
{
  "description": "Your request is missing some parameters."
}
```

SENDING MULTIPLE UNIQUE MESSAGES (REST ONLY)

There are times when you need to send many different but unique messages to your recipients. For example, fee balances reminders where each recipient has a unique account number, name and amount.

Make a HTTP POST request to <https://apis.semotime.com/v1/{accountId}/messages/multi> where {accountId} is the ID of the account you received during API authorization. The server expects:

- c. **Headers**

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'application/json'

b. **POST request body**

Your POST request body should be a JSON string containing the details of the message to send. For example,

```
{ "messages": [
  {
    "recipient": "0724310714",
    "text": "Dear JAMES WOKABI, A/c No:170154418, Prev Reading:24664, Curr Reading: 25217, Cons:553 Act., Monthly Bill Kshs.:0, Total Outstanding:0, Billing Month:October 2019, Due date:31-10-2019."
  },
  {
    "recipient": "0724310714",
    "text": "Dear SYLVESTER MUNGUTI MASU, A/c No:170153713, Prev Reading:806, Curr Reading:811, Cons:5 Act., Monthly Bill Kshs.:370, Total Outstanding:-1710, Billing Month:October 2019, Due date:31-10-2019"
  }
],
  "description": "October 2019 bills"
}
```

You will get back a response similar to the one received when you send a simple single message as shown above.

GETTING DETAILED DELIVERY REPORTS

The response you get when a message has been sent is not detailed enough. In fact, it only contains a count of those that were delivered, failed and pending. You need to send another request to get a detailed delivery report.

The reason we do it this way is because delivery reports keep on being updated by the mobile service providers periodically. A message that was in ‘pending’ status may only get ‘delivered’ 2-3 hours later if the subscriber was out of network coverage at the time of sending.

To get a deliver report, you only need the ID of the message. Usually the message ID will part of the parameters you receive when you send a message. See the above section on sending messages.

1. Using REST

Make a HTTP GET request to <https://apis.sematetime.com/v1/{accountId}/messages/{messageId}/delivery> where {accountId} is the ID of the account you received during API authorization and {messageId} is the ID of the message whose report you want. The server expects:

a. **Headers**

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be ‘application/json’

2. Using URL parameters

You only need to make a HTTP GET call to

<https://apis.sematetime.com/v1/{accountId}/messages/{messageId}/delivery.url>

and pass the AuthToken as a query parameter where {accountId} is the ID of the account you received during API authorization and {messageId} is the ID of the message whose report you want.

Regardless of the method you use, here is an example of the response you can expect.

```
{
  "summary": [
    {
      "count": 1,
      "status": "Delivered"
    },
    {
      "count": 0,
      "status": "Pending"
    },
    {
      "count": 0,
      "status": "Undeliverable"
    },
    {
      "count": 0,
      "status": "Expired"
    },
    {
      "count": 0,
      "status": "Rejected"
    }
  ],
  "reports": [
    {
      "senderId": "Sematime",
      "phoneNumber": "0768028xxx",
      "studentNumber": "",
      "smsCount": 1,
      "name": "",
      "sentAt": 1562314576807,
      "message": "The quick brown dog jumped over the lazy boy.\nCall:0706129xxx",
      "error": {
        "reason": "No Error",
        "permanent": false,
        "name": "OK",
        "description": "The message has been delivered.",
        "id": 0
      },
      "gatewayId": "262314576727353xxxx",
      "deliveredAt": 1562314583600,
      "status": {
        "reason": "Delivered to handset",
        "name": "Delivered",
        "description": "Message has been successfully processed.",
        "id": 5
      }
    }
  ]
}
```

SCHEDULING A MESSAGE FOR LATER SENDING

Sometime you want to have a message sent on a future date and time without you actually being present to send it manually. For example, you may need a message to be sent tomorrow at midnight.

You will need to send us the following parameters.

Parameter	Data Type	Constraint	Description
Text	String	required	The message you want to send.
recipients	String	required	a comma separated string that contains phone numbers. It can be formatted as follows, "0736189xxx, 0706129xxx"
scheduledDate	Long	required	The date and time in milliseconds when you want the message to be sent Learn more - https://currentmillis.com

1. Using REST

Make a HTTP POST request to <https://apis.semotime.com/v1/{accountId}/messages/schedule> where {accountId} is the ID of the account you received during API authorization. The server expects:

a. Headers

Your request must be accompanied by the following HTTP headers:

- AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'application/json'

b. POST request body

Your POST request body should be a JSON string containing the details of the message to send. For example,

```
{
  "text": "The big brown fox jumps over the lazy dog.",
  "recipients": "0706129xxx,James<0737189xxx>",
  "scheduledDate": 1562331240000
}
```

2. Using URL parameters

You only need to make a HTTP GET call to <https://apis.semotime.com/v1/{accountId}/messages/schedule.url> and pass the AuthToken, message, recipients and scheduledDate as query parameters. For example,

<https://apis.semotime.com/v1/145276454xxx/messages/schedule.url?text=The quick brown dog jumped over the lazy boy&recipients=0768028xxx&AuthToken=1ad63cf06775416ba1bb413a51694xxx&scheduledDate= 1562331480000>

Regardless of the method you use to schedule the message, you will get back a response that looks like this:

```
{
  "description": "Successfully scheduled for sending on 05/07/2019 03:58 PM"
}
```

SENDING EXAM RESULTS

This is a little different than sending a simple message. For starters, we will need you to have an Excel file containing the exam results to send. Your file needs to have the scores per subject, position, marks and grade for each student.

Download a sample Excel file here - https://drive.google.com/open?id=1F9cqIN_4zlu7Lohbmqq_PFoCQBw2HhRJ

POS	ADM	NAME	ORG CL	ENG	KIS	MATH	BIO	PHY	CHEM	HIST	GEO	CRE	AGR	COMP	FREN	GER	B/ST	ENTRY	TTL	AVG
1	4497	MUHOHO EVANSON GITONGA	E	9	23	9	12	10	28	27			26				13	9	157	17
2	00001	NDIRANGU LUCAS MWANGI	w	20	19	12	25	16	26			16				20		8	154	19
3	12602/16	KANYI JOHN NDEGWA	w	19	24	18	11	10	25	26							18	8	151	19
4	12540/16	KIHARA NGAMAU	E	17	22	13	10	15	25	33							12	8	147	18
5	12423/16	GACHOKA JOSEPH KAROBIA	w	17	14	15	12	16	26	20			22					8	142	18
6	12521/16	NDERITU NEWTON NDEGWA	E	16	16	16	14	15	19			21	24					8	141	18
7	13297/18	WALTER NJIRU NJERU	E	20	24	14	6	8	15	33			15					8	135	17
8	12456/16	KARENGO IAN DEINHARD	E	11	21	12	19	10	21	25							15	8	134	17
9	12610/16	MWEMA BORNVENTURE MACHIRA	w	10	25	14	14	15	14	25		16						8	133	17
10	12391/16	NGUGI VICTOR WANJAGI	E	13	29	16	14	13	20	25								7	130	19
11	12417/16	WACHIRA SAMUEL MUKIRI	S	20	11	5	8	5	29	31				21				8	130	16
12	12510/16	GITHINJI FELIX WANJAU	N	13	18	15	15	5	23		22				18			8	129	16
13	12567/16	MUMIN ABDUL KARIM	E	13	20	6	11	12	20			23		23				8	128	16
14	12449/16	MAINA JEFF MURIITHI	w	17	15	7	18	10	29		17		14					8	127	16
15	12572/16	NDERITU JOHN MUNYI	w	17	26	9	7	4	26	12				24				8	125	16
16	12930/17	IAN JAMES KIARIE MAINA	S	15	18	9	4	13	21	21		22						8	123	15

Sending exam results is a two-step process and whether you are using REST architecture or the URL parameters method you must follow the steps below.

- Upload Excel file for processing
- Provide additional information

In addition to the Excel file, you will also need to send us the following parameters:

Parameter	Data Type	Constraint	Description
examName	String	required	The name of the exam whose results you are sending
studentIdColumn	String	required	The name of the column that contains student IDs
studentNameColumn	String	required	The name of the column that contains student names
positionColumn	String	required	The name of the column that contains positions
marksColumn	String	required	The name of the column that contains the marks scored
gradeColumn	String	optional	The name of the column that contains the grades
classAverage	String	optional	The average marks for this exam

classSize	Integer	optional	Number of students who sat this exam
bestScore	String	optional	The marks/grade scored by best student

1. Uploading Excel file using REST

Make a HTTP POST request to <https://apis.semamtime.com/v1/{accountId}/exams/upload> where {accountId} is the ID of the school you received during API authorization. The server expects:

a. Headers

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'multipart/form-data'

b. POST request body

Your POST request body should contain the Excel file that you wish to send. The name of the file should be 'file'. For example,

POST /v1/1452764545xxx/exams/upload HTTP/1.1

Host: apis.semamtime.com

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="file"; filename="/Users/uncleboni/Downloads/Premock.xlsx

-----WebKitFormBoundary7MA4YWxkTrZu0gW--

If successful, you will get a response that looks like this:

```
{
  "fileContents":
    "POS,ADM,NAME,ORG,CL,ENG,KIS,MATH,BIO,PHY,CHEM,HIST,GEO,CRE,AGR,COMP,FREN,GER,
    B/ST,ENTRY,TTL,AVG,
    1,4497,MUHORO EVANSON GITONGA,E,9,23,9,12,10,28,27,,,26,,,13,9,157,17
    2,00001,NDIRANGU LUCAS MWANGI,w,20,19,12,25,16,26,,,16,,,20,8,154,19
    3,12602/16,KANYI JOHN NDEGWA,w,19,24,18,11,10,25,26,,,,,,18,8,151,19
    4,12540/16,KIHARA NGAMAU,E,17,22,13,10,15,25,33,,,,,,12,8,147,18
    5,12423/16,GACHOKA JOSEPH KAROBIA,w,17,14,15,12,16,26,20,,,22,,,,,8,142,18"
}
```

As you can see we are going to respond back to you with the contents of the Excel file converted to a comma separate values(CSV) string. In the second step of sending exam results you will need to send us back the exact file contents.

2. Uploading Excel file using the URL parameters method

As you will note, this process is very similar to the one we use in REST. There is a subtle difference in the endpoint URL and also in the response returned. While the REST architecture method returns the file contents, this method returns the URL of the Excel file after we have uploaded it to our servers.

Make a HTTP POST request to <https://apis.semamtime.com/v1/{accountId}/exams/upload.url> where {accountId} is the ID of the school you received during API authorization. Remember to add the AuthToken as a query parameter. For example,

<https://apis.semamtime.com/v1/1452764545xxx/exams/upload.url?AuthToken=1ad63cf06775416ba1bb413xxx>

The server expects:

a. **Headers**

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'multipart/form-data'

b. **POST request body**

Your POST request body should contain the Excel file that you wish to send. The name of the file should be 'file'. For example,

[POST /v1/1452764545xxx/exams/upload.url](https://apis.semamtime.com/v1/1452764545xxx/exams/upload.url) HTTP/1.1

Host: apis.semamtime.com

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="file"; filename="/Users/uncleboni/Downloads/Premock.xlsx

-----WebKitFormBoundary7MA4YWxkTrZu0gW

If successful, you will get a response that looks like this:

```
{
  "fileUrl": "https://s3.amazonaws.com/web.semamtime.com/1452764545xxx/exams/Premock.xlsx"
}
```

In the second step of sending exam results you will need to send us back the URL of the file exactly as it appears in the response.

3. **Sending exam results using REST**

Make a HTTP POST request to <https://apis.semamtime.com/v1/{accountId}/exams/send> where {accountId} is the ID of the school you received during API authorization. The server expects:

a. **Headers**

Your request must be accompanied by the following HTTP headers:

- iii. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- iv. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'application/json'

b. **POST request body**

Your POST request body should be a JSON string containing the details of the exam to send. For example,

```
{
  "fileContents": "POS,ADM,NAME,ORG,CL,ENG,KIS,MATH,BIO,PHY,CHEM,HIST,GEO,CRE,AGR,COMP,FREN,GER,B/ST,ENTRY,TTL,AVG\n1,4497,MUHORO EVANSON GITONGA,E,9,23,9,12,10,28,27,,26,,,13,9,157,17\n2,00001,NDIRANGU LUCAS MWANGI,w,20,19,12,25,16,26,,16,,,20,,8,154,19 \n3,12602/16,KANYI JOHN NDEGWA,w,19,24,18,11,10,25,26,,,,,18,8,151,19 \n4,12540/16,KIHARA NGAMAU,E,17,22,13,10,15,25,33,,,,,12,8,147,18 \n5,12423/16,GACHOKA JOSEPH KAROBIA,w,17,14,15,12,16,26,20,,22,,,8,142,18",
  "examName": "PreMocks",
}
```

```

"studentIdColumn": "ADM",
"studentNameColumn": "Name",
"positionColumn": "POS",
"marksColumn": "TTL",
"classAverage": "480",
"classSize": 35,
"bestScore": "600"
}

```

4. Sending exam results using URL parameters

You only need to call <https://apis.sematetime.com/v1/{accountId}/exams/send.url> where {accountId} then pass the fileUrl, examName, studentIdColumn, studentNameColumn, positionColumn, marksColumn, gradeColumn, classAverage, etc. For example,

```

https://apis.sematetime.com/v1/1452764545xxx/exams/send.url?
AuthToken=1ad63cf06775416ba1bb413a51694xxx&fileUrl=https://s3.amazonaws.com/web.sematetime.com/
1452764545xxx/exams/
Premock.xlsx&studentIdColumn=ADM&studentNameColumn=NAME&positionColumn=POS&marksColumn=TTL&e
xamName=PreMocks

```

If the exam results are sent successfully, you get a response that looks similar to this:

```

{
  "attachment": {},
  "signature": "",
  "sender": {},
  "pending": 1,
  "totalSmsCount": 2,
  "delivered": 0,
  "smsType": "exams",
  "engagements": 0,
  "id": 1562341614769,
  "text": "PreMocks results exam results",
  "failed": 0
}

```

SENDING FEE BALANCES

Sending fee balances is very similar to sending exam results. To begin with, we'll need an Excel file that contains the student ID, name and fee balance amount.

Download a sample Excel file here –

<https://drive.google.com/open?id=1CGju-0eaDzuj6bTSj44WnMVZ00-XRVna>

Adm No.	Name	Amount		
4497	Suleiman Shakoba	2500		
1101	Dennis Milo	3500		
1102	Rachel Wanjiku	5090		
1103	Abraham Nicko	7800		

Sending fee balances is also a two-step process and whether you are using REST architecture or the URL parameters method you must follow the steps below.

- a. Upload Excel file for processing
- b. Provide additional information

In addition to the Excel file, you will also need to send us the following parameters:

Parameter	Data Type	Constraint	Description
studentIdColumn	String	required	The name of the column that contains student IDs
studentNameColumn	String	required	The name of the column that contains student names
amountColumn	String	required	The name of the column that contains fee balance amounts
additionalInfo	String	optional	A message appended to the fee balance message. e.g instructions on how to pay

1. Uploading Excel file using REST

Make a HTTP POST request to <https://apis.semotime.com/v1/{accountId}/fees/upload> where {accountId} is the ID of the school you received during API authorization. The server expects:

c. Headers

Your request must be accompanied by the following HTTP headers:

- iii. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- iv. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'multipart/form-data'

d. POST request body

Your POST request body should contain the Excel file that you wish to send. The name of the file should be 'file'. For example,

`POST /v1/1452764545xxx/fees/upload HTTP/1.1`

`Host: apis.semotime.com`

`Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW`

`Content-Disposition: form-data; name="file"; filename="/Users/uncleboni/Downloads/Balances.xlsx`

`-----WebKitFormBoundary7MA4YWxkTrZu0gW--`

If successful, you will get a response that looks like this:

```
{
  "fileContents": "Adm No., Name,Amount,,\n4497,Suleiman Shakoba,2500,,\n1101,Dennis Milo,
3500,,\n1102,Rachel Wanjiku,5090,,\n1103,Abraham Nicko,7800,,\n"
}
```

Our API responds back with the contents of the file formatted in comma separated values(CSV) format. We will need the file contents when sending the fee balances.

2. Uploading Excel file using the URL parameters method

The difference between uploading the Excel file through the URL parameters method and the REST architecture method is in the endpoint URL and also in the response returned. While the REST architecture method returns the file contents, this method returns the URL of the Excel file after we have uploaded it to our servers.

Make a HTTP POST request to <https://apis.sematetime.com/v1/{accountId}/fees/upload.url> where {accountId} is the ID of the school you received during API authorization. Remember to add the AuthToken as a query parameter. For example,

<https://apis.sematetime.com/v1/1452764545xxx/fees/upload.url?AuthToken=1ad63cf06775416ba1bb413xxx>

The server expects:

c. Headers

Your request must be accompanied by the following HTTP headers:

- iii. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- iv. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'multipart/form-data'

d. POST request body

Your POST request body should contain the Excel file that you wish to send. The name of the file should be 'file'. For example,

[POST /v1/1452764545xxx/fees/upload.url HTTP/1.1](#)

[Host: apis.sematetime.com](#)

[Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW](#)

[Content-Disposition: form-data; name="file"; filename="/Users/uncleboni/Downloads/Balances.xlsx](#)

[-----WebKitFormBoundary7MA4YWxkTrZu0gW](#)

If successful, you will get a response that looks like this:

```
{
  "fileUrl": "https://s3.amazonaws.com/web.sematetime.com/1452764545xxx/fees/Balances.xlsx"
}
```

PS: You will use this file URL in the second step of sending fee balances.

3. Sending fee balances using REST

Make a HTTP POST request to <https://apis.sematetime.com/v1/{accountId}/fees/send> where {accountId} is the ID of the school you received during API authorization. The server expects:

c. Headers

Your request must be accompanied by the following HTTP headers:

- v. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- vi. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be 'application/json'

d. POST request body

Your POST request body should be a JSON string containing the details of the exam to send. For example,

```
{
  "fileContents": "Adm No.,Name,Fee Balance,\n4497,Suleiman Shakoba,2500,\n1101,Dennis Milo,
3500,\n1102,Rachel Wanjiku,5090,\n1103,Abraham Nicko,7800,\n",
  "studentIdColumn": "Adm No.",
  "studentNameColumn": "Name",
  "amountColumn": "Amount"
}
```

4. Sending fee balances using URL parameters

You only need to call <https://apis.semotime.com/v1/{accountId}/exams/send.url> where {accountId} then pass the fileUrl, studentIdColumn, studentNameColumn and amountColumn. For example,

```
https://apis.semotime.com/v1/1452764545xxx/fees/send.url?
AuthToken=1ad63cf06775416ba1bb413a51694xxx&fileUrl=https://s3.amazonaws.com/web.semotime.com/
145276454xxx/exams/Balances.xlsx&studentIdColumn=Adm
No.&studentNameColumn=Name&amountColumn=Amount
```

If you are successful, you will get back a response similar to:

```
{
  "attachment": {},
  "signature": "",
  "sender": {},
  "pending": 0,
  "totalSmsCount": 0,
  "delivered": 0,
  "smsType": "fees",
  "engagements": 0,
  "id": 1562345095663,
  "text": "Fee balances",
  "failed": 0
}
```

FETCHING YOUR ACCOUNT INFO

Make a HTTP GET request to <https://apis.semotime.com/v1/{accountId}/accounts> where {accountId} is the ID of the account you received during API authorization. The server expects:

a. Headers

Your request must be accompanied by the following HTTP headers:

- i. **AuthToken** – a mandatory header containing your API key as gotten from your credentials.
- ii. **Content-Type** – a header containing the content type of the data you are sending to the API. It should always be ‘application/json’

You will get back a response formatted as follows.

```
{
  "country": {},
  "settings": {},
  "timezone": "(GMT+3:00) Africa/Nairobi",
  "schoolType": "secondary school",
  "active": true,
  "lastActiveDate": 1574329793553,
  "logoUrl": "https://shulepay.nyc3.digitaloceanspaces.com/logo.png",
}
```

```
{  
  "population":264,  
  "emailAddress":"info@sematime.com",  
  "phoneNumber":"0706129100",  
  "balance":3655,  
  "name":"St. Anthony High School",  
  "registrationDate":1452764545XXX,  
  "id":"1452764545XXX"  
}
```

GETTING HELP

If you are struggling to get the API to work with your programming language, you need to import our API collections into your Postman then use the code generation feature in Postman to get code snippets in your language.

1. Sematime 4.0 API Postman collection
https://drive.google.com/file/d/15PCwqpF7pGttOOowRM8WX2FwrExpHgPN/view?usp=drive_open
2. How to generate code snippets in Postman
https://learning.getpostman.com/docs/postman/sending_api_requests/generate_code_snippets/

You may contact Sematime on 0706129100 and ask to speak to our API lead. Alternatively, you can chat with him on Skype - @githinjibn

Happy coding.