Letter & Conference API:

This API will return the following information:

1. **Letters**
    a. **To be printed** – letters that are about to be printed
    b. **Suppressed** – letters that are suppressed by the end user
    c. **Printed** – letters that have been printed and mailed
    d. **Pre-loaded** – letters that have been pre-loaded by the district
2. **Conferences**
    a. **To be scheduled** – list of students that need a conference
    b. **Scheduled** – list of conferences that are scheduled
    c. **To be closed** – list of conferences that have occurred and still need to be closed
    d. **Closed** – list of conferences that are closed
    e. **To be rescheduled** – list of students that have been removed from a conference and still need to be rescheduled into a new conference.

The A2A API will perform requests and provide responses from an HTTP protocol known as a **Re**presentational **S**tate **T**ransfer (REST).   The REST APIs can be access via web browser and/or programmatically. For example to say hello to your mom,

https://a2aapi.sia-us.com/api/data/hello/mom

Security:

- Each API will require a userName and password passed or posted to the REST URL. For example, the following URL arguments would be added to the API's URL

    ?userName=myUserName&password=myPassword

- Each district will have their own private URL. For example, for district 'district' the base URL would be: https://a2aapi.sia-us.com/api/data/district/letter
- If the username and district don't match the eSolutions security then no data will be returned.
- A user will be disabled if he/she tries to log in with the same number of invalid attempts as if he/she tried from the UI.
    o The user will need to be re-enabled in the Users Administration page.

Common API Arguments:

The following Arguments can either be posted and/or added to the URL:

   userName (required) – the user's user name

password (required) - the user's password

format (default json) – preferred return format (html, json, or XML)

page(default 1)– the page of data to return

page_size (default 10 and max 100) – the number of records to return per page

For example,

> https://a2aapi.sia-us.com/api/data/myDistrictCode/letter/ALL/?userName=myUserName&password=myPassword&page=1&page_size=2&format=xml

Letter API:

Base URL: https://a2aapi.sia-us.com/api/data/{districtCode}/letter/{letterType}/{studentCode}

{districtCode} (required) – this part of the URL is the district's district code

{letterType} (required) – this tells the API which letters to return. This can only be one of the following:

> **SUMMARY** – return a summary of the district's letters
>
> **ALL** – return all letters
>
> **TO_BE_PRINTED** – only return letters that are scheduled to be printed
>
> **SUPPRESSED** – only return letters that are suppressed
>
> **PRINTED** – only return letters that have been printed
>
> **PRE_LOADED** – only return letters that have been pre-loaded

{studentCode} (optional) – this optionally tells the API only return letters for a particular student. NOTE: This argument is ignored when returning the "SUMMARY"

Example URLs,

> https://a2aapi.sia-us.com/api/data/zzAnthony/letter/SUMMARY
>
> https://a2aapi.sia-us.com/api/data/zzAnthony/letter/ALL
>
> https://a2aapi.sia-us.com/api/data/zzAnthony/letter/ALL/1234

https://a2aapi.sia-us.com/api/data/zzAnthony/letter/PRINTED

https://a2aapi.sia-us.com/api/data/zzAnthony/letter/PRINTED/1234

Sample output:

Summary:

```
{
    "letter_api": {
        "letter_L1_to_be_printed_count": 4,
        "letter_L2_to_be_printed_count": 3,
        "letter_L3_to_be_printed_count": 8,
        "letter_EEA2_to_be_printed_count": 1,
        "letter_T1_to_be_printed_count": 1,
        "letter_L1_suppressed_count": 1,
        "letter_L3_suppressed_count": 1,
        "letter_L1_printed_count": 17,
        "letter_L2_printed_count": 14,
        "letter_L3_printed_count": 1,
        "letter_EEA1_printed_count": 2
    },
    "log": {
        "execution_id": 1438980321497,
        "access": "GRANTED",
        "user_id": 5808,
        "district": "zzAnthony",
        "date": "2015-08-07 13:45:21.497",
        "api": "letter",
        "type": "summary",
        "execution_time": 22143
    }
}
```

All Letters:

```
{
    "letter_api": {
        "navigation": {
            "letter_type": "ALL",
            "total_records": 53,
            "page": 1,
            "page_size": 10
        },
        "letters": [
            {"letter": {
                "id": 1079814,
                "letter": "L1",
                "site": "300",
                "letter_date": "2015-08-06",
                "status": "PR",
                "student": {
                    "last_name": "Abberero",
                    "first_name": "Gina",
```

HEADQUARTERS
5200 Golden Foothill Parkway
El Dorado Hills, CA 95762
MAIN   800.487.9234
FAX   888.487.6441

NORTHERN CALIFORNIA
2151 Larkspur Lane, Unit B
Redding, CA 96002
MAIN   800.487.9234
FAX   866.752.7767

SOUTHERN CALIFORNIA
790 East Colorado Blvd, Suite 700
MAIN   800.487.9234
FAX   866.573.8570

```
            "grade": "7",
            "site": "300",
            "student_code": "1000013",
            "notes": [{"note": {
                "text": "Spoke with Gina's parents
regarding missing too much school and they assured us that this
will never ever ever happen again.",
                "created_date": "2015-09-09",
                "created_by": "YoMomma@sia-us.com"
            }}]
```

Conference API:

Base URL: https://a2aapi.sia-us.com/api/data/{districtCode}/conference/{conferenceType}/{studentCode}

{districtCode} (required) – this part of the URL is the district's district code

{conferenceType} (required) – this tells the API which conferences to return. This can only be one of the following:

SUMMARY – return a summary of the district's conferences

TO_BE_SCHEDULED – only return students that need a conference scheduled

TO_BE_CONDUCTED – only return conferences that are scheduled

TO_BE_CLOSED – only return conferences that need to be closed

TO_BE_RESCHEDULED - only return students that need to be rescheduled

CLOSED – only return conferences that have been closed

{studentCode} (optional) – this optionally tells the API only return conferences for a particular student. NOTE: This argument is ignored when returning the "SUMMARY"

Example URLs,

https://a2aapi.sia-us.com/api/data/zzAnthony/conference/SUMMARY

https://a2aapi.sia-us.com/api/data/zzAnthony/conference/TO_BE_SCHEDULED

https://a2aapi.sia-us.com/api/data/zzAnthony/conference/TO_BE_SCHEDULED/1234

Sample output:

Summary:

```
{
    "conference_api": {
        "conference_to_be_scheduled_count": 6,
        "conference_to_be_conducted_count": 1,
        "conference_to_be_closed_count": 6,
        "conference_to_be_rescheduled_count": 0,
        "conference_closed_count": 1,
        "my_conference_scheduled_count": 0,
        "my_conference_to_be_closed_count": 0,
        "my_conference_closed_count": 0
    },
    "log": {
        "execution_id": 1438980478795,
        "access": "GRANTED",
        "user_id": 5808,
        "district": "zzAnthony",
        "date": "2015-08-07 13:47:58.795",
        "api": "conference",
        "type": "summary",
        "execution_time": 22094
    }
}
```

Conferences to be closed:

```
{
    "conference_api": {
        "navigation": {
            "conference_type": "TC",
            "total_records": 4,
            "page": 1,
            "page_size": 10
        },
        "conferences": [
            {"conference": {
                "id": 86547,
                "location": "Principal's Office",
                "date": "2015-08-12 10:00:00.0",
                "num_students": 5,
                "closed": false,
                "students": [
                    {"student": {
                        "last_name": "Perez",
                        "first_name": "Clifford",
                        "grade": "6",
                        "site": "300",
                        "student_code": "1000015",
                        "letter_code": "L2",
                        "conference_status": "",
                        "notes": []
                    }},
                    {"student": {
                        "last_name": "Abberero",
```

```
                    "first_name": "Gina",
                    "grade": "7",
                    "site": "300",
                    "student_code": "1000013",
                    "letter_code": "L2",
                    "conference_status": "",
                    "notes": []
            }},
            {"student": {
                    "last_name": "Norman",
                    "first_name": "Jared",
                    "grade": "8",
                    "site": "300",
                    "student_code": "1000011",
                    "letter_code": "L2",
                    "conference_status": "",
                    "notes": []
            }},
```

Sample Java Program to access the APIs

```java
package myPackage;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;

public class SampleApi {
      public static void main(String[] args) {
            try {
                  String districtCode = "myDistrict";
                  String api = "conference";
                  String apiType = "summary";
                  String userName = "myUserName";
                  String password = "myPassword";
                  String format = "json";

                  final String apiURL = "https://a2aapi.sia-us.com/api/data/" + districtCode + "/" + api + "/" + apiType;
                  final URL url = new URL(apiURL);
                  HttpURLConnection conn = (HttpURLConnection) url.openConnection();

                  HashMap<String, String> params = new HashMap<String, String>();

                  StringBuffer postData = null;
                  byte [] postDataBytes = null;

                  /**
                   * Prepare the form parameters
                   */
                  params.put("userName", userName);
                  params.put("password", password);
```

```java
            params.put("format", format);

            if (params != null && params.size() > 0) {
                for (String param : params.keySet()) {
                    if (postData == null) {
                        postData = new StringBuffer();
                    } else {
                        postData.append("&");
                    }

        postData.append(URLEncoder.encode(param, "UTF-
8")).append("=").append(URLEncoder.encode(params.get(param), "UTF-8"));
                }
            }
            postDataBytes = postData.toString().getBytes("UTF-
8");

            /**/

            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
            conn.setRequestProperty("Accept",
"application/json");
            if (postDataBytes != null) {
                conn.setRequestProperty("Content-Length",
String.valueOf(postDataBytes.length));
                conn.getOutputStream().write(postDataBytes);
            }

            if (conn.getResponseCode() != 200) {
                throw new RuntimeException("Failed : HTTP
error code : "
                            + conn.getResponseCode());
            }

            /**
             * Output the API response
             */
            BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));

            String output;
            System.out.println("API DATA: \n");
            while ((output = br.readLine()) != null) {
                System.out.println(output);
            }

            conn.disconnect();
        } catch (Exception e) {
            System.err.println("problem calling api: " +
e.getMessage());

            e.printStackTrace(System.err);
        }
```

```
            }
}
```

Sample PYTHON Program to access the APIs

```
####  PYTHON SI&A Letter API TESTER V1.0 ####

####   August 10, 2015 ####


from urllib2 import Request, urlopen, URLError


districtCode = 'zzqa'

userName = 'qa1@sia-us.com'

password = 'password'

outputFormat = 'json' ### Acceptable format types = json, xml or html

apiURL = 'https://a2aapi.sia-us.com/api/data/'

api = 'letter'  ### Acceptable api Types = letter or conference

page = '1' ### (Default is 1) - the page of data to return

pageSize = '100' ### number of Records to return per page (default 10 and max 100)

apiType = 'pr' ### Acceptable apiType = all, GE, SU, PR, NP, PL, summary

### summary RETURNS Letter Summary

### all    RETURNS All Letters

### GE     RETURNS To be Printed

### SU     RETURNS Suppressed Letters

### PR     RETURNS Printed Letters

### NP     RETURNS NO Print Letters

### PL     RETURNS Preloaded Letters

studentCode = '' ### Student Code this optionally tells the API only return letters for a
particular student.  NOTE: This argument is ignored when returning the "SUMMARY"
```

```python
### studentCode default is '' : usage: '/1000008' ###




request = Request(apiURL + districtCode + '/' + api + '/' + apiType + studentCode +
'?userName='

          + userName + '&password=' + password +  '&page=' + page +'&page_size=' +
pageSize + '&format=' + outputFormat)



try:

        response = urlopen(request)

        myData = response.read()

        print myData

except URLError, e:

    print 'No Data was returned. Got an error code:', e
```