



**EXPERIMENT MANAGEMENT SYSTEM
APPLICATION PROGRAMMING INTERFACE (API)**

Last updated 5 January 2025

Copyright © 2025 Sona Systems, Ltd., All Rights Reserved

Table of Contents

Introduction	5
Intended Audience.....	5
About This Documentation.....	5
Obtaining Technical Support.....	5
Release Schedule	5
Improvements and Suggestions	6
API Introduction	7
API Access and URL.....	7
Authentication	7
Request Requirements	9
Accessing the API using HTTP GET	9
Accessing the API using SOAP.....	9
Request Examples for Common Programming Languages.....	10
Performance Considerations and Acceptable Use	16
Testing Tools	16
Security Concerns	16
API Functions	17
SonaConvertUserToSSOByPersonID	17
SonaConvertUserToSSOByUserID.....	19
SonaGetCoursesByPersonID	21
SonaGetPersonInfoByName	23
SonaSignUpUserToATimeslot	27
SonaGetSignUpsForTimeslot	29
SonaGetSignupsByDateRange	31
SonaGetStudyScheduleList	33
SonaGetStudyList	36
SonaGetResearcherIDByExperimentID.....	40
SonaGetPrescreenSurveyDataByPersonID	42

SonaGetPrescreenSurveyDataByDateRange	45
SonaGetPrescreenDataByExperimentID.....	49
SonaGetPrescreenDataByTimeslotID	53
SonaGrantCreditBySignupID	57
SonaFixStudyCreditAssignmentMismatch.....	59
SonaGetPersonInfoByPersonID	61
SonaGetPersonInfoByStudentID.....	64
SonaGetPersonInfoByUserID	67
SonaGetCreditBySignupID	70
SonaGetCreditsByPersonID.....	72
SonaGetCreditsByTimeslotID.....	75
SonaGetCreditsByDateRange.....	78
SonaGetPersonListByClassID	81
SonaGetTimeslotsByExperimentID.....	87
SonaGetTimeslotsByExperimentIDDateRange	90
SonaGetParticipantStats	93
SonaGetStudyStats.....	95
SonaGetStudyStatsByDateRange.....	97
SonaGetInstructorByClassID	99
SonaGetPersonListByLoginDate.....	102
SonaGetPersonListByCreationDate	106
SonaGetStudyInfoByExperimentID.....	110
SonaGetStudyListByResearcher	113
SonaGetStudyListByPI.....	116
SonaActivateUser.....	119
SonaCreateUser	121
SonaDeactivateUser.....	124
SonaApproveUser	126
SonaSetUserAdminComments.....	128
WebStudyCreditGrant.....	130
API Data Classes	131

APICourse	131
APICreditItem.....	132
APISignup	134
APIStudySchedule	135
APIStudyInfo	137
APIPrescreenSurveyData	141
APIPrescreenSurveyResponseItem.....	142
APIPerson	143
APIParticipantStats	146
APIStudyStats	147
Revision History	148

Introduction

The Application Programming Interface (API) for the Experiment Management System is a tool for advanced users to communicate directly with the system via a program or script they have written. Typically, this would be used to query specific information from the database for processing outside the system, or to do batch processing of updates.

The benefit of the API is that this can be done without human interaction. Without the API, the process of getting data from the system would involve logging into the system and going to the relevant page, downloading the data in CSV format, and parsing that. The login process is difficult to automate, so this approach does not work well for repetitive or scheduled tasks.

With the API, you can write your own script to connect directly to the system and query this information. One usage scenario would be if you want to display on a monitor in the main offices a list of all studies running that day. Using the API, you can write a program to query this information and then send it for display on the monitor. Another scenario would be if you want to have a daily or weekly batch process to query timeslot usage and update usage limits based on that information.

Intended Audience

It is important to understand that the API is intended for advanced users only, and requires a certain amount of programming knowledge. If terms like XML, SOAP, and HTTP are unfamiliar to you, then the API is not for you. If the API is too complicated, your best option would be to ask your IT department to write the code for you. Unfortunately, we do not have the resources to provide any training in technical concepts.

About This Documentation

This documentation covers the Application Programming Interface (API) for the Experiment Management System. API features will be released outside of the normal version schedule for our main software, so this document will be updated on a routine basis as new API functions are added.

Obtaining Technical Support

We can provide technical support to advise on the best API functions to use to solve your problem, and also to troubleshoot a problem in the API itself. We are not equipped to debug your code or solve programming problems.

Release Schedule

As releasing new API features does not change the end-user interface or functionality, then we plan to release new API features on a more frequent basis. Because of this, the API will not have

a version number. This documentation will be updated on a regular basis to reflect any updates in the API.

Improvements and Suggestions

Because adding new features to the API does not impact the end-user interface, we are very open to adding new features to the API, especially if they provide a generic function that all customers can benefit from. Please feel free to contact us with any suggestions or questions.

API Introduction

API Access and URL

The API is accessed at the same address as your Sona site. If your Sona site's URL is <https://myschool.sona-systems.com> then the API is located at <https://myschool.sona-systems.com/services/SonaAPI.svc>

You can then access each function either through an HTTPS GET request with the parameters specified in the URL, or as a SOAP request (XML over HTTPS) as a POST.

WSDL

If you go to that URL listed above (changing the "myschool" to your own site of course), you will find a link to the WSDL (Web Services Description Language) data for all functions within the API. Note that the WSDL may point to www.sona-systems.com so you may need to modify this to your site's own URL.

You can find a direct link to the WSDL for your site at <https://myschool.sona-systems.com/services/SonaAPI.svc?singleWsd> (change "myschool" to your site's URL).

Authentication

All API functions require an API key. You must include a valid API key as a parameter in the URL using `api_key=YOUR_API_KEY`. API keys are user-specific and can be generated as follows:

1. Navigate to the Users | Add/Edit/Search section on your main site.
2. Find the user for whom you want to generate an API key.
3. Click on the "Set API Key" button next to the user's information.
4. On the "Set API Key" page, click "Set API Key" to generate a new key for that user.

Each user can have their own API key, allowing for individualized access control and tracking of API usage. Only administrator accounts may use the API.

Authentication Errors

Listed below are the different authentication error codes along with their description.

Error Code	Description
0	unspecified error
-1	bad username/password*
-2	bad user type (not an admin)
-3	access not allowed (account disabled)
-4	user not found
-5	access prevented due to password account lockout
-6	reach API key limit
-7	bad API key

** Username/password access has been deprecated and will be removed by December 31, 2024.*

These error codes apply only if the API key authentication fails. If it works OK, then the return will merely be the typical return value from the API call (and unrelated to authentication).

Username and Password API Authentication Deprecation

To enhance the security of the Sona platform and better support users accessing Sona via Single Sign-On (SSO), we have deprecated username and password authentication for API access. This change will be completed by December 31, 2024.

Moving forward, all API functions will require the use of an API key for authentication. Please update your integrations accordingly to ensure uninterrupted access to Sona services.

Request Requirements

All requests must have a User-Agent header field specified, to properly comply with HTTP standards. The value of this parameter is up to the developer. Some guidance on User-Agents can be found [here](#). This field is required as of 1 August 2023, and our firewall will block requests that omit this, as this is a common pattern from malicious scripts. An example value might be:

```
GET /services/SonaAPI.svc/ws HTTP/1.1
User-Agent: myCustomClient/202309011456
```

Accessing the API using HTTP GET

One method to access the API, particularly if you are just querying data (as opposed to uploading a set of data), is to request a URL with the function name and necessary input parameters, and the system will then return an XML-formatted document with the result. This is an easy way to make most programs use the API, since it involves just loading a URL.

Accessing the API using SOAP

The other method to access the API is through a SOAP (Simple Object Access Protocol) call, which is an XML-formatted request that is sent as a POST over HTTPS. This works well (and is sometimes the only option) if you are using an API function that requires sending a large amount of data, since that's not suitable for an HTTPS GET call.

Making an HTTP POST Request

Listed below is a sample HTTPS POST request to the API for the SonaGetPersonInfoByUserID function, assuming your site name is `yoursite.sona-systems.com`.

Example request:

```
POST /services/SonaAPI.svc/ws HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Host: yoursite.sona-systems.com
User-Agent: myCustomClient/202309011456
Content-Length: 695
Expect: 100-continue
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Example Response:

HTTP/1.1 100 Continue

Message body (update site, credentials, and request specific values – MessageID is optional):

```
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action mustUnderstand="1">
      https://www.sona-systems.com/SonaAPI/SonaGetPersonInfoByUserID_Post
    </a:Action>
    <a:MessageID>urn:uuid:73e142a1-df57-4bfe-9caa-dc9ef790d50c</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To mustUnderstand="1">
      https://yoursite.sona-systems.com/services/SonaAPI.svc/ws
    </a:To>
  </s:Header>
  <s:Body>
    <SonaGetPersonInfoByUserID_Post xmlns="https://www.sona-systems.com/">
      <api_key>your_api_key_here</api_key>
      <user_id>a_user</user_id>
    </SonaGetPersonInfoByUserID_Post>
  </s:Body>
</s:Envelope>
```

The main things to note here are to send the POST request to the URL but with /ws appended to the end, and to append _Post to the function call.

Request Examples for Common Programming Languages

The following examples show how to POST a SOAP request, with the message body above included in a file named data.txt. There are examples in: CURL, Python, Node.js, C#/.NET, Java, and PHP.

CURL

```
curl -v -X POST --http1.1 -H "Content-Type: application/json; charset=utf-8" \
  --data-binary @data.txt -A "myCustomClient/202309011456" \
  https://yoursite.sona-systems.com/services/SonaAPI.svc/SonaActivateUser
```

Python

```
import requests

# Specify the file path and headers
file_path = 'data.txt'
headers = {
    'Content-Type': 'application/json; charset=utf-8',
    'User-Agent': 'myCustomClient/202309011456'
}

# Read the data from the file
with open(file_path, 'r') as file:
    data = file.read()

# Set the URL for the API endpoint
url = 'https://yoursite.sona-systems.com/services/SonaAPI.svc/SonaActivateUser'

# Send the POST request
response = requests.post(url, headers=headers, data=data)

# Output the response
print(response.status_code)
print(response.text)
```

Node.js

```
const fs = require('fs');
const axios = require('axios');

// Read the data from the file
const data = fs.readFileSync('data.txt', 'utf8');

// Set the headers
const headers = {
  'Content-Type': 'application/json; charset=utf-8',
  'User-Agent': 'myCustomClient/202309011456'
};

// Set the URL for the API endpoint
const url = 'https://yoursite.sona-
systems.com/services/SonaAPI.svc/SonaActivateUser';

// Send the POST request
axios.post(url, data, { headers })
  .then(response => {
    console.log(response.status);
    console.log(response.data);
  })
  .catch(error => {
    console.error(error);
  });
```

C#/.NET

```
using System;
using System.IO;
using System.Net.Http;
using System.Threading.Tasks;

class Program
{
    static async Task Main(string[] args)
    {
        string url = "https://yoursite.sona-
systems.com/services/SonaAPI.svc/SonaActivateUser";
        string filePath = "data.txt";

        using HttpClient client = new HttpClient();
        client.DefaultRequestHeaders.Add("User-Agent", "myCustomClient/202309011456");

        string data = await File.ReadAllTextAsync(filePath);
        HttpContent content = new StringContent(data, System.Text.Encoding.UTF8,
"application/json");

        HttpResponseMessage response = await client.PostAsync(url, content);

        string responseBody = await response.Content.ReadAsStringAsync();
        Console.WriteLine(response.StatusCode);
        Console.WriteLine(responseBody);
    }
}
```

Java

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import okhttp3.*;

public class PostRequestExample {

    public static void main(String[] args) throws IOException {

        String url = "https://yoursite.sona-
systems.com/services/SonaAPI.svc/SonaActivateUser";
        String filePath = "data.txt";
        OkHttpClient client = new OkHttpClient();

        String data = new String(Files.readAllBytes(Paths.get(filePath)));
        RequestBody body = RequestBody.create(data, MediaType.get("application/json;
charset=utf-8"));

        Request request = new Request.Builder()
            .url(url)
            .post(body)
            .header("User-Agent", "myCustomClient/202309011456")
            .build();

        try (Response response = client.newCall(request).execute()) {
            System.out.println(response.code());
            System.out.println(response.body().string());
        }
    }
}
```

PHP

```
<?php
$url = 'https://yoursite.sona-systems.com/services/SonaAPI.svc/SonaActivateUser';
$file_path = 'data.txt';

$data = file_get_contents($file_path);

$options = [
    'http' => [
        'header' => [
            "Content-Type: application/json; charset=utf-8",
            "User-Agent: myCustomClient/202309011456"
        ],
        'method' => 'POST',
        'content' => $data,
    ],
];

$context = stream_context_create($options);
$response = file_get_contents($url, false, $context);

if ($response === FALSE) {
    die('Error');
}

echo $http_response_header[0] . "\n";
echo $response;
?>
```

Performance Considerations and Acceptable Use

The system enforces a rate limit of 600 API requests per 5-minute interval. Design your programs to use the API efficiently, minimizing unnecessary calls. If abusive patterns are detected, additional restrictions may be applied.

One example for how to use the API efficiently would be to cache or store locally certain data if it does not change often, instead of calling the API for the same data every time. Also, instead of “polling” the API every few minutes for data, call it only when data is needed (e.g., requested by the user).

Note that calls to the API are to the same site as your main site is running on, so excessive calls can slow down the main site for all your users. We reserve the right to terminate access to the API if it is excessive or adversely affecting site performance.

Testing Tools

There are several tools out there for testing your API calls to determine if they work. One tool available from Microsoft is the WCF Test Client, which is part of the Visual Studio package. More information on the WCF Test Client can be found here: <http://msdn.microsoft.com/en-us/library/bb552364.aspx>

Security Concerns

When using the URL GET method for API calls, the API key is included as part of the URL. Please be aware that this approach has some security considerations:

1. Proxy servers within your network may log or cache these calls, potentially exposing the API key.
2. Our servers maintain logs of parameters used in GET requests as part of standard website logging practices.

For enhanced security, you may prefer to use SOAP calls to access the API. With SOAP, the API key is included in the request body rather than the URL, reducing the likelihood of it being logged or cached by intermediate systems.

Always treat your API key as sensitive information and avoid sharing it unnecessarily. If you suspect your API key has been compromised, generate a new one immediately through the user management interface.

API Functions

SonaConvertUserToSSOByPersonID

Description

This API converts a non-SSO user to a SSO user using the numeric `person_id` of the user as the identifier.

Input Parameters

Below are the following input parameters for use with the SonaConvertUserToSSOByPersonID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The <code>person_id</code> to convert to SSO (e.g. 1234)
sso_id	The SSO id of the user (e.g. <code>someuser@myschool.edu</code>)

Output Parameters

Will return True if converted else errors.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	User not found
Error Code	-9	<code>sso_id</code> already exists
Error Code	-10	Invalid SSO ID
Error Code	-11	Invalid <code>person_id</code>

Sample Input

https://myschool.sona-
systems.com/services/SonaAPI.svc/SonaConvertUserToSSOByPersonID?
api_key=YOUR_API_KEY&person_id=auser&sso_id=auser@myschool.edu

Sample Output

```
<ConvertUserToSSOByPersonIDResponse xmlns="https://www.sona-systems.com/">
  <ConvertUserToSSOByPersonIDResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    <a:Result>true</a:Result>
  </ConvertUserToSSOByPersonIDResult>
</ConvertUserToSSOByPersonIDResponse>
```

SonaConvertUserToSSOByUserID

Description

This API converts a non-SSO user to an SSO user using the `user_id` of the user as the identifier.

Input Parameters

Below are the following input parameters for use with the SonaConvertUserToSSOByUserID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
user_id	The user ID to convert to SSO (e.g. <i>someuser</i>)
sso_id	The SSO ID of the user (e.g. <i>someuser @myschool.edu</i>)

Output Parameters

Will return True if converted else errors.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	sso_id already exists
Error Code	-9	Invalid SSO ID
Error Code	-10	user_id does not exist
Error Code	-11	Invalid user_id

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaConvertUserToSSOByUserID?api_key=YOUR_API_KEY&user_id=auser&sso_id=auser@myschool.edu

Sample Output

```
<ConvertUserToSSOByUserIDResponse xmlns="https://www.sona-systems.com/">
  <ConvertUserToSSOByUserIDResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>true</a:Result>
  </ConvertUserToSSOByUserIDResult>
</ConvertUserToSSOByUserIDResponse>
```

SonaGetCoursesByPersonID

Description

This API retrieves a list of all courses the participant specified by `person_id` is in.

Input Parameters

Below are the following input parameters for use with the SonaGetCoursesByPersonID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The <code>person_id</code> to retrieve courses of

Output Parameters

Will return APICourse classes and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	User is not a participant.
Error Code	-9	Invalid person ID

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetCoursesByPersonID?api_key=YOUR_API_KEY&person_id=1234

Sample Output

```
<GetCoursesByPersonIDResponse xmlns="https://www.sona-systems.com/">
  <GetCoursesByPersonIDResult
    xmlns:a=http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    <a:Result>
      <a:APICourse>
        <a:class_id>5</a:class_id>
        <a:class_name>My Class</a:class_name>
        <a:credits_required>30</a:credits_required>
        <a:instructor_id>9458</a:instructor_id>
        <a:instructor_id2>0</a:instructor_id2>
        <a:selectable>1</a:selectable>
      </a:APICourse>
    </a:Result>
  </GetCoursesByPersonIDResult>
</GetCoursesByPersonIDResponse>
```

SonaGetPersonInfoByName

Description

This API retrieves the person information for all users with specified name(s). At least one name parameter must be specified (first or last name), or both. When searching on only one name parameter, include the other parameter but pass a blank value, instead of omitting the parameter entirely. There may be more than one match. Search is case-insensitive.

Input Parameters

Below are the following input parameters for use with the SonaGetPersonInfoByName API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
first_name	The first name search parameter (<i>first and/or last name required, pass blank value if not being searched</i>)
last_name	The last name search parameter (<i>first and/or last name required, pass blank value if not being searched</i>)

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	No users found with supplied first name and/or last name
Error Code	-9	No first_name or last_name supplied

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetPersonInfoByName?api_key=YOUR_API_KEY&first_name=john&last_name=doe

Sample Output

```
<GetPersonInfoByNameResponse xmlns="https://www.sona-systems.com/">
  <GetPersonInfoByNameResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasytems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments />
        <a:alt_email>testme@yourschool.edu</a:alt_email>
        <a:approved>1</a:approved>
        <a:birthdate>0001-01-01T00:00:00</a:birthdate>
        <a:classes_needed>1</a:classes_needed>
        <a:creation_date>0001-01-01T00:00:00</a:creation_date>
        <a:credits_earned>0</a:credits_earned>
        <a:credits_needed>1.00</a:credits_needed>
        <a:daily_reminder>0</a:daily_reminder>
        <a:email_needed>0</a:email_needed>
        <a:first_name>test</a:first_name>
        <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>0001-01-01T00:00:00</a:last_login>
        <a:last_name>Test</a:last_name>
        <a:limit_date>0001-01-01T00:00:00</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>0</a:login_days>
        <a:login_sent_date>2012-10-08T04:05:24</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>1</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>9416</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_i>0</a:person_type_i>
        <a:person_type_p>0</a:person_type_p>
```



```
<a:person_type_pref>S</a:person_type_pref>
<a:person_type_s>1</a:person_type_s>
<a:phone>12444</a:phone>
<a:pretest>0</a:pretest>
<a:student_id_num>12333</a:student_id_num>
<a:study_invitation>1</a:study_invitation>
<a:usage_limit>0</a:usage_limit>
<a:user_id>testme</a:user_id>
<a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments />
  <a:alt_email>test3323@yourschool.edu</a:alt_email>
  <a:approved>1</a:approved>
  <a:birthdate>0001-01-01T00:00:00</a:birthdate>
  <a:classes_needed>1</a:classes_needed>
  <a:creation_date>0001-01-01T00:00:00</a:creation_date>
  <a:credits_earned>0</a:credits_earned>
  <a:credits_needed>0</a:credits_needed>
  <a:daily_reminder>0</a:daily_reminder>
  <a:email_needed>0</a:email_needed>
  <a:first_name>Test</a:first_name>
  <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>0001-01-01T00:00:00</a:last_login>
  <a:last_name>Test</a:last_name>
  <a:limit_date>0001-01-01T00:00:00</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>0</a:login_days>
  <a:login_sent_date>2013-05-03T10:29:13</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>1</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>9435</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_i>0</a:person_type_i>
  <a:person_type_p>0</a:person_type_p>
  <a:person_type_pref>S</a:person_type_pref>
  <a:person_type_s>1</a:person_type_s>
  <a:phone>301-111-1111</a:phone>
  <a:pretest>0</a:pretest>
```

```
<a:student_id_num>111-23</a:student_id_num>
<a:study_invitation>1</a:study_invitation>
<a:usage_limit>0</a:usage_limit>
<a:user_id>test3323</a:user_id>
<a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
</a:Result>
</GetPersonInfoByNameResult>
</GetPersonInfoByNameResponse>
```

SonaSignUpUserToATimeslot

Description

This API function signs users up for a specific timeslot. The timeslot is specified as a `timeslot_id`. The user is specified using their `person_id`. Note: This function only works for lab studies and external web studies. It does not work for internal online survey studies.

Input Parameters

Below are the following input parameters for use with the SonaSignUpUserToATimeslot API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The <code>person_id</code> of the person to sign up.
timeslot_id	The <code>timeslot_id</code> of the timeslot to sign the person up for.

Output Parameters

Will return APISignup class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid or no <code>timeslot_id</code> supplied
Error Code	-9	No participant found
Error Code	-10	No timeslot found
Error Code	-11	Invalid or no <code>person_id</code> supplied

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaSignUpUserToATimeslot?api_key=YOUR_API_KEY&person_id=11498×lot_id=75

Sample Output

```
<SignUpUserToATimeslotResponse>
  <SignUpUserToATimeslotResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APISignUp>
        <a:anon_id_code>1234</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>mike@myschool.edu</a:email>
        <a:first_name>Mike</a:first_name>
        <a:last_name>Lee</a:last_name>
        <a:phone>8509886</a:phone>
        <a:signup_id>258</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341404</a:student_id_num>
        <a:survey_code>222</a:survey_code>
        <a:user_id>mikel</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
      <a:APISignUp>
        <a:anon_id_code>5678</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>jesse@myschool.edu</a:email>
        <a:first_name>Jesse</a:first_name>
        <a:last_name>James</a:last_name>
        <a:phone>8509887</a:phone>
        <a:signup_id>266</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341405</a:student_id_num>
        <a:survey_code>729</a:survey_code>
        <a:user_id>jessej</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
    </a:Result>
  </SignUpUserToATimeslotResult>
</SignUpUserToATimeslotResponse>
```

SonaGetSignUpsForTimeslot

Description

This API retrieves the list of signup records for a given timeslot. The timeslot is specified as a `timeslot_id`. In the system's interface, when you click on a timeslot, you will see the `timeslot_id` in the URL in the browser address section.

Input Parameters

Below are the following input parameters for use with the SonaGetSignUpsForTimeslot API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
timeslot_id	The numeric reference of the timeslot from which to retrieve the records from.

Output Parameters

Will return APISignup class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid timeslot_id	[entered value]	timeslot_id must be numeric
No signups found for timeslot_id, or invalid timeslot_id	[entered value]	either no records were found for timeslot_id or timeslot_id does not exist

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetSignUpsForTimeslot?  
api_key=YOUR_API_KEY&timeslot_id=75
```

Sample Output

```
<GetSignUpsForTimeslotResponse>
  <GetSignUpsForTimeslotResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APISignUp>
        <a:anon_id_code>1234</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>mike@myschool.edu</a:email>
        <a:first_name>Mike</a:first_name>
        <a:last_name>Lee</a:last_name>
        <a:phone>8509886</a:phone>
        <a:signup_id>258</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341404</a:student_id_num>
        <a:survey_code>222</a:survey_code>
        <a:user_id>mikel</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
      <a:APISignUp>
        <a:anon_id_code>5678</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>jesse@myschool.edu</a:email>
        <a:first_name>Jesse</a:first_name>
        <a:last_name>James</a:last_name>
        <a:phone>8509887</a:phone>
        <a:signup_id>266</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341405</a:student_id_num>
        <a:survey_code>729</a:survey_code>
        <a:user_id>jessej</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
    </a:Result>
  </GetSignUpsForTimeslotResult>
</GetSignUpsForTimeslotResponse>
```

SonaGetSignupsByDateRange

Description

This API retrieves the list of signup records, within a given date range.

Input Parameters

Below are the following input parameters for use with the SonaGetSignupsByDateRange API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
start_date	The start date of the date range to include. Date must be formatted as: YYYY-MM-DD.
end_date	The end date of the date range to include. Date must be formatted as: YYYY-MM-DD.

Output Parameters

Will return APISignup class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid end date
Error Code	-9	Start date is after end date
Error Code	-10	Invalid start date
Error Code	-11	No credit record found for selected date range

Sample Input

`https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetSignupsByDateRange?api_key=YOUR_API_KEY&start_date=2014-01-01&end_date=2014-12-31`

Sample Output

```
<GetSignupsByDateRangeResponse>
  <GetSignupsByDateRangeResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APISignUp>
        <a:anon_id_code>1234</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>mike@myschool.edu</a:email>
        <a:first_name>Mike</a:first_name>
        <a:last_name>Lee</a:last_name>
        <a:phone>8509886</a:phone>
        <a:signup_id>258</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341404</a:student_id_num>
        <a:survey_code>222</a:survey_code>
        <a:user_id>mikel</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
      <a:APISignUp>
        <a:anon_id_code>5678</a:anon_id_code>
        <a:credit_status>G</a:credit_status>
        <a:email>jesse@myschool.edu</a:email>
        <a:first_name>Jesse</a:first_name>
        <a:last_name>James</a:last_name>
        <a:phone>8509887</a:phone>
        <a:signup_id>266</a:signup_id>
        <a:signup_date>2019-08-13T15:00:00</a:signup_date>
        <a:student_id_num>09341405</a:student_id_num>
        <a:survey_code>729</a:survey_code>
        <a:user_id>jessej</a:user_id>
        <a:link_click_date>2019-08-14T16:00:00</a:link_click_date>
      </a:APISignUp>
    </a:Result>
  </GetSignupsByDateRangeResult>
</GetSignupsByDateRangeResponse>
```


SonaGetStudyScheduleList

Description

This API retrieves the list of studies within a given date range. It basically mimics the Study Schedule Report as found in the Generate Reports page of the main site.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyScheduleList API function.

Parameter	Description / Possible Values								
api_key	Use API key generated for Sona admin user								
location_id	<p>Possible values are as follows:</p> <table border="1"><tbody><tr><td>-2</td><td>All locations</td></tr><tr><td>-1</td><td>All locations with no resource ids</td></tr><tr><td>0</td><td>All locations with resources ids</td></tr><tr><td>> 0</td><td>Specific resource id</td></tr></tbody></table> <p>A resource_id exists if the location was chosen from the pulldown list of locations set in Location Listings.</p>	-2	All locations	-1	All locations with no resource ids	0	All locations with resources ids	> 0	Specific resource id
-2	All locations								
-1	All locations with no resource ids								
0	All locations with resources ids								
> 0	Specific resource id								
start_date	Date to use for beginning of listing. Date must be formatted as: YYYY-MM-DD.								
end_date	Date to use for end of listing. Date must be formatted as: YYYY-MM-DD.								
lab_only	Possible values are as follows: 1 = only standard studies 0 = all studies.								

Output Parameters

Will return APIStudySchedule class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid start_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)
Invalid end_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)
Invalid date range (start date is after end date)	[entered value] to [entered value]	start_date must be earlier than end_date
Invalid location_id	[entered value]	location_id must be numeric
Invalid lab_only	[entered value]	lab_only must be numeric

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyScheduleList?
api_key=YOUR_API_KEY&location_id=-2&start_date=2012-01-03&
end_date=2012-01-10&lab_only=1
```

Sample Output

```
<GetStudyScheduleListResponse>
  <GetStudyScheduleListResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>11</a:experiment_id>
        <a:location>LAB 01</a:location>
        <a:study_name>Study Name 1</a:study_name>
        <a:survey_flag>0</a:survey_flag>
        <a:timeslot_date>2011-11-16T15:00:00</a:timeslot_date>
        <a:timeslot_id>67</a:timeslot_id>
        <a:web_flag>0</a:web_flag>
        <a:num_signed_up>5</a:num_signed_up>
        <a:num_students>10</a:num_students>
        <a:researcher_id>9311</a:researcher_id>
        <a:videoconf_flag>1</a:videoconf_flag>
        <a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
      </a:APIStudySchedule>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
```

```
<a:experiment_id>12</a:experiment_id>
<a:location>LAB 02</a:location>
<a:study_name>Study Name 2</a:study_name>
<a:survey_flag>0</a:survey_flag>
<a:timeslot_date>2011-11-16T15:00:00</a:timeslot_date>
<a:timeslot_id>68</a:timeslot_id>
<a:web_flag>0</a:web_flag>
<a:num_signed_up>3</a:num_signed_up>
<a:num_students>3</a:num_students>
<a:researcher_id>9312</a:researcher_id>
<a:videoconf_flag>1</a:videoconf_flag>
<a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
</a:APIStudySchedule>
</a:Result>
</GetStudyScheduleListResult>
</GetStudyScheduleListResponse>
```

SonaGetStudyList

Description

This API retrieves the list of studies in the system based on study type. Note that multipart studies will be listed once for each part of the study, since each part (internally) has a unique `experiment_id`. The `multipart_number` field will list which part of the study it is, and this is 1 for all non-multipart studies.

Some information is not included here, like the list of researchers, pre-requisites, and disqualifiers for a study, because these are stored separately. For a list of researchers for the study, use the `SonaGetResearcherIDByExperimentID` API function.

Input Parameters

Below are the following input parameters for use with the `SonaGetStudyList` API function.

Parameter	Description / Possible Values						
api_key	Use API key generated for Sona admin user						
active	Possible values are as follows: <table border="1"><tbody><tr><td>-1</td><td>All</td></tr><tr><td>0</td><td>Inactive studies only</td></tr><tr><td>1</td><td>Active studies only</td></tr></tbody></table>	-1	All	0	Inactive studies only	1	Active studies only
-1	All						
0	Inactive studies only						
1	Active studies only						
approved	Possible values are as follows: <table border="1"><tbody><tr><td>-1</td><td>All</td></tr><tr><td>0</td><td>Unapproved studies only</td></tr><tr><td>1</td><td>Approved studies only</td></tr></tbody></table>	-1	All	0	Unapproved studies only	1	Approved studies only
-1	All						
0	Unapproved studies only						
1	Approved studies only						
web_flag	Possible values are as follows: <table border="1"><tbody><tr><td>-1</td><td>All</td></tr></tbody></table>	-1	All				
-1	All						

	<table border="1"> <tr> <td>0</td> <td>Lab studies only</td> </tr> <tr> <td>1</td> <td>Web studies only (internal online and external web)</td> </tr> </table>	0	Lab studies only	1	Web studies only (internal online and external web)		
0	Lab studies only						
1	Web studies only (internal online and external web)						
survey_flag	<p>Possible values are as follows:</p> <table border="1"> <tr> <td>-1</td> <td>All</td> </tr> <tr> <td>0</td> <td>All except online surveys</td> </tr> <tr> <td>1</td> <td>Online surveys only</td> </tr> </table>	-1	All	0	All except online surveys	1	Online surveys only
-1	All						
0	All except online surveys						
1	Online surveys only						

Output Parameters

Will return APIStudyInfo class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid active value	[entered value]	active must be numeric
Invalid approved value	[entered value]	approved must be numeric
Invalid web_flag value	[entered value]	web_flag must be numeric
Invalid survey_flag value	[entered value]	survey_flag must be numeric
No records found		no records were found/retrieved

Sample Input

`https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyList?api_key=YOUR_API_KEY&active=-1&approved=-1&web_flag=-1&survey_flag=-1`


```
<a:study_name>Online Internal Study 1</a:study_name>
<a:survey_anon_flag>0</a:survey_anon_flag>
<a:survey_flag>1</a:survey_flag>
<a:total_credits>1.00</a:total_credits>
<a:total_parts>1</a:total_parts>
<a:total_payment />
<a:type>C</a:type>
<a:usage_limit>200</a:usage_limit>
<a:valid>1</a:valid>
<a:web_flag>1</a:web_flag>
</a:APIStudyInfo>
</a:Result>
</GetStudyListResult>
</GetStudyListResponse>
```

SonaGetResearcherIDByExperimentID

Description

This API call retrieves the list of researchers (by person_id) for a study, based on the study's experiment_id.

Input Parameters

Below are the following input parameters for use with the SonaGetResearcherIDByExperimentID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study.

Output Parameters

Will return the person_id of all researchers for the study.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	invalid experiment_id
Error Code	-9	no researchers found for experiment_id or invalid experiment_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetResearcherIDByExperimentID?  
api_key=YOUR_API_KEY&experiment_id=123
```

Sample Output

```
<GetResearcherIDByExperimentIDResponse>  
  <GetResearcherIDByExperimentIDResult>
```



```
<a:Errors i:nil="true" />
<a:Result>
  <a:int>9272</a:int>
  <a:int>9406</a:int>
</a:Result>
</GetResearcherIDByExperimentIDResult>
</GetResearcherIDByExperimentIDResponse>
```

SonaGetPrescreenSurveyDataByPersonID

Description

This API call retrieves the list of prescreen or survey responses of a particular participant.

Input Parameters

Below are the following input parameters for use with the SonaGetPrescreenSurveyDataByPersonID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study. Pass in 0 for the prescreen.
person_id	The person_id of the participant.

Output Parameters

Will return APIPrescreenSurveyData class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	no prescreen nor survey found for the specified experiment_id
Error Code	-9	no pretest_respondents or people record found
Error Code	-10	Invalid experiment_id
Error Code	-11	Invalid person_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetPrescreenSurveyDataByPersonID?  
api_key=YOUR_API_KEY&experiment_id=0&person_id=9444
```

Sample Output

```
<GetPrescreenSurveyDataByPersonIDResponse>  
  <GetPrescreenSurveyDataByPersonIDResult>  
    <a:Errors i:nil="true" />  
    <a:Result>  
      <a:anon_id_code />  
      <a:data_status>S</a:data_status>  
      <a:end_time>2013-05-07T00:26:02</a:end_time>  
      <a:first_name>Jim</a:first_name>  
      <a:last_name>Smith</a:last_name>  
      <a:person_id>9444</a:person_id>  
      <a:pretest_id>10</a:pretest_id>  
      <a:responses>  
        <a:APIPrescreenSurveyResponseItem>  
          <a:choice_id>462</a:choice_id>  
          <a:opt_out_choice>0</a:opt_out_choice>  
          <a:question_id>17</a:question_id>  
          <a:response_id>1020</a:response_id>  
          <a:response_numeric>0</a:response_numeric>  
          <a:response_text>4 sup</a:response_text>  
        </a:APIPrescreenSurveyResponseItem>  
        <a:APIPrescreenSurveyResponseItem>  
          <a:choice_id>132</a:choice_id>  
          <a:opt_out_choice>0</a:opt_out_choice>  
          <a:question_id>27</a:question_id>  
          <a:response_id>1021</a:response_id>  
          <a:response_numeric>0</a:response_numeric>  
          <a:response_text>kinda</a:response_text>  
        </a:APIPrescreenSurveyResponseItem>  
        <a:APIPrescreenSurveyResponseItem>  
          <a:choice_id>147</a:choice_id>  
          <a:opt_out_choice>0</a:opt_out_choice>  
          <a:question_id>32</a:question_id>  
          <a:response_id>1022</a:response_id>  
          <a:response_numeric>0</a:response_numeric>  
          <a:response_text>erg</a:response_text>  
        </a:APIPrescreenSurveyResponseItem>  
        <a:APIPrescreenSurveyResponseItem>  
          <a:choice_id>293</a:choice_id>  
          <a:opt_out_choice>0</a:opt_out_choice>  
          <a:question_id>75</a:question_id>  
          <a:response_id>1023</a:response_id>  
          <a:response_numeric>0</a:response_numeric>  
          <a:response_text>yeah!</a:response_text>
```

```
    </a:APIPrescreenSurveyResponseItem>
  </a:responses>
  <a:start_time>2013-05-07T00:25:36</a:start_time>
  <a:status>C</a:status>
  <a:user_id>jsmith</a:user_id>
</a:Result>
</GetPrescreenSurveyDataByPersonIDResult>
</GetPrescreenSurveyDataByPersonIDResponse>
```

SonaGetPrescreenSurveyDataByDateRange

Description

This API retrieves the list of prescreen or survey responses, for multiple participants, based on a given date range.

Input Parameters

Below are the following input parameters for use with the SonaGetPrescreenSurveyDataByDateRange API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study. Pass in 0 for the prescreen.
start_date	The start date of the date range to include. Date must be formatted as: YYYY-MM-DD. This date is based on when the participant completed the prescreen or online survey.
end_date	The end date of the date range to include. Date must be formatted as: YYYY-MM-DD. This date is based on when the participant completed the prescreen or online survey.

Output Parameters

Will return APIPrescreenSurveyData class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	no prescreen nor survey found for the specified experiment_id
Error Code	-9	no pretest_respondents or people record found

Error Code	-10	invalid start date
Error Code	-11	invalid end date
Error Code	-12	start date is greater than end date
Error Code	-13	no data found for selected date range
Error Code	-14	experiment_id is not found/invalid

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetPrescreenSurveyDataByDateRange?api_key=YOUR_API_KEY&experiment_id=0&start_date=2013-01-01&end_date=2013-12-31

Sample Output

```
<GetPrescreenSurveyDataByDateRangeResponse>
  <GetPrescreenSurveyDataByDateRangeResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPrescreenSurveyData>
        <a:anon_id_code />
        <a:data_status>S</a:data_status>
        <a:end_time>2013-04-24T15:28:58</a:end_time>
        <a:first_name>Frank</a:first_name>
        <a:last_name>Bass</a:last_name>
        <a:person_id>9394</a:person_id>
        <a:pretest_id>10</a:pretest_id>
        <a:responses>
          <a:APIPrescreenSurveyResponseItem>
            <a:choice_id>57</a:choice_id>
            <a:opt_out_choice>0</a:opt_out_choice>
            <a:question_id>17</a:question_id>
            <a:response_id>1005</a:response_id>
            <a:response_numeric>0</a:response_numeric>
            <a:response_text>külm</a:response_text>
          </a:APIPrescreenSurveyResponseItem>
          <a:APIPrescreenSurveyResponseItem>
            <a:choice_id>129</a:choice_id>
            <a:opt_out_choice>1</a:opt_out_choice>
            <a:question_id>27</a:question_id>
        </a:responses>
      </a:APIPrescreenSurveyData>
    </a:Result>
  </a:Errors>
</a:Errors>
```

```

    <a:response_id>1006</a:response_id>
    <a:response_numeric>0</a:response_numeric>
    <a:response_text>[Decline to Answer]</a:response_text>
</a:APIPrescreenSurveyResponseItem>
<a:APIPrescreenSurveyResponseItem>
    <a:choice_id>145</a:choice_id>
    <a:opt_out_choice>1</a:opt_out_choice>
    <a:question_id>32</a:question_id>
    <a:response_id>1007</a:response_id>
    <a:response_numeric>0</a:response_numeric>
    <a:response_text>[Decline to Answer]</a:response_text>
</a:APIPrescreenSurveyResponseItem>
</a:responses>
<a:start_time>2013-04-24T15:28:45</a:start_time>
<a:status>C</a:status>
<a:user_id>bass</a:user_id>
</a:APIPrescreenSurveyData>
<a:APIPrescreenSurveyData>
    <a:anon_id_code />
    <a:data_status>S</a:data_status>
    <a:end_time>2013-04-29T23:31:55</a:end_time>
    <a:first_name>Greg</a:first_name>
    <a:last_name>Smith</a:last_name>
    <a:person_id>9308</a:person_id>
    <a:pretest_id>10</a:pretest_id>
    <a:responses>
        <a:APIPrescreenSurveyResponseItem>
            <a:choice_id>57</a:choice_id>
            <a:opt_out_choice>0</a:opt_out_choice>
            <a:question_id>17</a:question_id>
            <a:response_id>1010</a:response_id>
            <a:response_numeric>0</a:response_numeric>
            <a:response_text>külm</a:response_text>
        </a:APIPrescreenSurveyResponseItem>
        <a:APIPrescreenSurveyResponseItem>
            <a:choice_id>130</a:choice_id>
            <a:opt_out_choice>0</a:opt_out_choice>
            <a:question_id>27</a:question_id>
            <a:response_id>1011</a:response_id>
            <a:response_numeric>1</a:response_numeric>
            <a:response_text>Yes</a:response_text>
        </a:APIPrescreenSurveyResponseItem>
    </a:responses>
</a:APIPrescreenSurveyData>
</a:APIPrescreenSurveyData>

```

```
<a:choice_id>146</a:choice_id>
<a:opt_out_choice>0</a:opt_out_choice>
<a:question_id>32</a:question_id>
<a:response_id>1012</a:response_id>
<a:response_numeric>2</a:response_numeric>
<a:response_text>No</a:response_text>
</a:APIPrescreenSurveyResponseItem>
</a:responses>
<a:start_time>2013-04-29T23:31:38</a:start_time>
<a:status>C</a:status>
<a:user_id>smithg</a:user_id>
</a:APIPrescreenSurveyData>
</a:Result>
</GetPrescreenSurveyDataByDateRangeResult>
</GetPrescreenSurveyDataByDateRangeResponse>
```


SonaGetPrescreenDataByExperimentID

Description

This API retrieves the list of prescreen or survey responses, for multiple participants, based on `experiment_id`.

Input Parameters

Below are the following input parameters for use with the `SonaGetPrescreenDataByExperimentID` API function.

Parameter	Description / Possible Values
<code>api_key</code>	Use API key generated for Sona admin user
<code>experiment_id</code>	The <code>experiment_id</code> of the study. Pass in 0 for the prescreen.

Output Parameters

Will return `APIPrescreenSurveyData` class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	no prescreen nor survey found for the specified <code>experiment_id</code>
Error Code	-9	no response data or participant found for the specified <code>participant_id</code>
Error Code	-10	no data found
Error Code	-11	<code>experiment_id</code> is not found or is invalid
Error Code	-12	no <code>pretest_respondents</code> or people record found

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/  
SonaGetPrescreenDataByExperimentID?  
api_key=YOUR_API_KEY&experiment_id=0
```

Sample Output

```
<GetPrescreenDataByExperimentIDResponse>  
  <GetPrescreenDataByExperimentIDResult>  
    <a:Errors i:nil="true" />  
    <a:Result>  
      <a:APIPrescreenSurveyData>  
        <a:anon_id_code />  
        <a:data_status>S</a:data_status>  
        <a:end_time>2013-04-24T15:28:58</a:end_time>  
        <a:first_name>Frank</a:first_name>  
        <a:last_name>Bass</a:last_name>  
        <a:person_id>9394</a:person_id>  
        <a:pretest_id>10</a:pretest_id>  
        <a:responses>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>57</a:choice_id>  
            <a:opt_out_choice>0</a:opt_out_choice>  
            <a:question_id>17</a:question_id>  
            <a:response_id>1005</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>külm</a:response_text>  
          </a:APIPrescreenSurveyResponseItem>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>129</a:choice_id>  
            <a:opt_out_choice>1</a:opt_out_choice>  
            <a:question_id>27</a:question_id>  
            <a:response_id>1006</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>[Decline to Answer]</a:response_text>  
          </a:APIPrescreenSurveyResponseItem>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>145</a:choice_id>  
            <a:opt_out_choice>1</a:opt_out_choice>  
            <a:question_id>32</a:question_id>  
            <a:response_id>1007</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>[Decline to Answer]</a:response_text>
```

```

    </a:APIPrescreenSurveyResponseItem>
  </a:responses>
  <a:start_time>2013-04-24T15:28:45</a:start_time>
  <a:status>C</a:status>
  <a:user_id>bass</a:user_id>
</a:APIPrescreenSurveyData>
<a:APIPrescreenSurveyData>
  <a:anon_id_code />
  <a:data_status>S</a:data_status>
  <a:end_time>2013-04-29T23:31:55</a:end_time>
  <a:first_name>Greg</a:first_name>
  <a:last_name>Smith</a:last_name>
  <a:person_id>9308</a:person_id>
  <a:pretest_id>10</a:pretest_id>
  <a:responses>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>57</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>17</a:question_id>
      <a:response_id>1010</a:response_id>
      <a:response_numeric>0</a:response_numeric>
      <a:response_text>külm</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>130</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>27</a:question_id>
      <a:response_id>1011</a:response_id>
      <a:response_numeric>1</a:response_numeric>
      <a:response_text>Yes</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>146</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>32</a:question_id>
      <a:response_id>1012</a:response_id>
      <a:response_numeric>2</a:response_numeric>
      <a:response_text>No</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
  </a:responses>
  <a:start_time>2013-04-29T23:31:38</a:start_time>
  <a:status>C</a:status>
  <a:user_id>smithg</a:user_id>

```

```
    </a:APIPrescreenSurveyData>  
  </a:Result>  
</GetPrescreenDataByExperimentIDResult>  
</GetPrescreenDataByExperimentIDResponse>
```

SonaGetPrescreenDataByTimeslotID

Description

This API retrieves the list of prescreen or survey responses, for multiple participants, based on `timeslot_id`.

Input Parameters

Below are the following input parameters for use with the `SonaGetPrescreenDataByTimeslotID` API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
timeslot_id	The <code>timeslot_id</code> of the study. Pass in 0 for the prescreen.

Output Parameters

Will return `APIPrescreenSurveyData` class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	no prescreen nor survey found for the specified <code>experiment_id</code>
Error Code	-9	no response data or participant found for the specified <code>participant_id</code>
Error Code	-10	no data found
Error Code	-11	<code>experiment_id</code> is not found or is invalid
Error Code	-12	no <code>pretest_respondents</code> or people record found

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/  
SonaGetPrescreenDataByTimeslotID?  
api_key=YOUR_API_KEY&experiment_id=0
```

Sample Output

```
<GetPrescreenDataByTimeslotIDResponse>  
  <GetPrescreenDataByTimeslotIDResult>  
    <a:Errors i:nil="true" />  
    <a:Result>  
      <a:APIPrescreenSurveyData>  
        <a:anon_id_code />  
        <a:data_status>S</a:data_status>  
        <a:end_time>2013-04-24T15:28:58</a:end_time>  
        <a:first_name>Frank</a:first_name>  
        <a:last_name>Bass</a:last_name>  
        <a:person_id>9394</a:person_id>  
        <a:pretest_id>10</a:pretest_id>  
        <a:responses>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>57</a:choice_id>  
            <a:opt_out_choice>0</a:opt_out_choice>  
            <a:question_id>17</a:question_id>  
            <a:response_id>1005</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>külm</a:response_text>  
          </a:APIPrescreenSurveyResponseItem>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>129</a:choice_id>  
            <a:opt_out_choice>1</a:opt_out_choice>  
            <a:question_id>27</a:question_id>  
            <a:response_id>1006</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>[Decline to Answer]</a:response_text>  
          </a:APIPrescreenSurveyResponseItem>  
          <a:APIPrescreenSurveyResponseItem>  
            <a:choice_id>145</a:choice_id>  
            <a:opt_out_choice>1</a:opt_out_choice>  
            <a:question_id>32</a:question_id>  
            <a:response_id>1007</a:response_id>  
            <a:response_numeric>0</a:response_numeric>  
            <a:response_text>[Decline to Answer]</a:response_text>
```

```

    </a:APIPrescreenSurveyResponseItem>
  </a:responses>
  <a:start_time>2013-04-24T15:28:45</a:start_time>
  <a:status>C</a:status>
  <a:user_id>bass</a:user_id>
</a:APIPrescreenSurveyData>
<a:APIPrescreenSurveyData>
  <a:anon_id_code />
  <a:data_status>S</a:data_status>
  <a:end_time>2013-04-29T23:31:55</a:end_time>
  <a:first_name>Greg</a:first_name>
  <a:last_name>Smith</a:last_name>
  <a:person_id>9308</a:person_id>
  <a:pretest_id>10</a:pretest_id>
  <a:responses>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>57</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>17</a:question_id>
      <a:response_id>1010</a:response_id>
      <a:response_numeric>0</a:response_numeric>
      <a:response_text>külm</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>130</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>27</a:question_id>
      <a:response_id>1011</a:response_id>
      <a:response_numeric>1</a:response_numeric>
      <a:response_text>Yes</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
    <a:APIPrescreenSurveyResponseItem>
      <a:choice_id>146</a:choice_id>
      <a:opt_out_choice>0</a:opt_out_choice>
      <a:question_id>32</a:question_id>
      <a:response_id>1012</a:response_id>
      <a:response_numeric>2</a:response_numeric>
      <a:response_text>No</a:response_text>
    </a:APIPrescreenSurveyResponseItem>
  </a:responses>
  <a:start_time>2013-04-29T23:31:38</a:start_time>
  <a:status>C</a:status>
  <a:user_id>smithg</a:user_id>

```

```
</a:APIPrescreenSurveyData>  
</a:Result>  
</GetPrescreenDataByTimeslotIDResult>  
</GetPrescreenDataByTimeslotIDResponse>
```


SonaGrantCreditBySignupID

Description

This function updates the credit status for the specified `signup_id`. This allows for granting credit, marking a no-show, and updating comments. Variable credit granting is supported. Only 1 signup can be updated at a time. Internal online survey studies are not supported.

Input Parameters

Below are the following input parameters for use with the SonaGrantCreditBySignupID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
signup_id	Numeric ID of the signup to adjust the credit status for.
credit_type	The new credit status for the sign-up. It should be one of the following single characters: G = credit grant W = awaiting action E = Excused No-Show I = Unexcused No-Show
comments	Comment for credit status (may be empty)
credits	Decimal value of credit to set (e.g. 2.0). Set to -1 for paid studies or to use the study's default credit value if a credit study. If Variable Credit Granting is enabled, then value may be set to a value from 0-2x the study's credit value, following the acceptable values as specified in Credit Increment in System Settings.
skip_credit_email	Whether an email should be sent to the participant about the change in credit status. <i>false</i> will send an email and <i>true</i> will not send the email

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	signup_id not found
Error Code	-9	credit_type invalid
Error Code	-10	credits value invalid
Error Code	-11	credits value increment invalid
Error Code	-12	skip_credit_email value is invalid
Error Code	-13	invalid study type
Error Code	-14	signup_id not valid (non-numeric)

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetPersonInfoByName?
 api_key=YOUR_API_KEY&signup_id=123&credit_type=G&
 comments=Test Comment&credits=1&skip_credit_email=false

Sample Output

```
<GrantCreditBySignupIDResponse xmlns="https://www.sona-systems.com/">
  <GrantCreditBySignupIDResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>true</a:Result>
  </GrantCreditBySignupIDResult>
</GrantCreditBySignupIDResponse>
```

If the output returned is similar to the output above, but the value of Result is false instead of true, this indicates that the information provided was valid, but that the signup was already marked with the same credit status (and comments) so there was no action to be taken.

SonaFixStudyCreditAssignmentMismatch

Description

This function adjusts all cases where there is a credit assignment mismatch. A credit assignment mismatch is a case where a study has course restrictions for a certain set of courses, and the participant (who is in one of those courses) has assigned the credit to another one of their courses that was not in the restriction list. An example would be if a study has course restrictions such that only participants in courses A and B may sign up. A participant who is in courses B and C signs up, and assigns the credit to course C. This situation is entirely normal and usually allowable.

This API call will look for all cases of a mismatch. When it finds one, it will go through that participant's courses in order based on the course name, and reassign the credit to the first course that is within the course restrictions for the study. If there are no courses to assign the credit to (a rare case), it will un-assign the credit from the current course.

The return value indicates the number of cases where a credit was reassigned or unassigned.

This API call was added for a specific customer, and may not generally be of use in most situations.

Input Parameters

Below are the following input parameters for use with the SonaFixStudyCreditAssignmentMismatch API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user

Output Parameters

Will return the number of mismatches fixed.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaFixStudyCreditAssignmentMismatch?api_key=YOUR_API_KEY

Sample Output

```
<FixStudyCreditAssignmentMismatchResponse>  
  <FixStudyCreditAssignmentMismatchResult>  
    <a:Errors i:nil="true" />  
    <a:Result>4</a:Result>  
  </FixStudyCreditAssignmentMismatchResult>  
</FixStudyCreditAssignmentMismatchResponse>
```

SonaGetPersonInfoByPersonID

Description

This function will provide all information from the main user (people) table about a particular user, based on their person_id. It will work for any user type, but which fields are populated depends on the user type.

Input Parameters

Below are the following input parameters for use with the SonaGetPersonInfoByPersonID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the user

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid person_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetPersonInfoByPersonID?  
api_key=YOUR_API_KEY&person_id=4321
```

Sample Output

```
<GetPersonInfoByPersonIDResponse>
  <GetPersonInfoByPersonIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments>Taking course again</a:admin_comments>
        <a:approved>1</a:approved>
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>
        <a:classes_needed>0</a:classes_needed>
        <a:creation_date>2013-01-12T00:00:00</a:creation_date>
        <a:credits_earned>4.50</a:credits_earned>
        <a:credits_needed>4.00</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <a:weekly_reminder>1</a:weekly_reminder>
        <a:study_invitation>0</a:study_invitation>
        <a:alt_email>jake@myschool.edu</a:alt_email>
        <a:email_needed>0</a:email_needed>
        <a:first_name>Jake</a:first_name>
        <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>2013-09-13T09:00:56</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>2013-09-13T09:00:53</a:last_login>
        <a:last_name>Rogers</a:last_name>
        <a:limit_date>2013-03-15T08:35:05</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>16</a:login_days>
        <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>0</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>4321</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_p>0</a:person_type_p>
        <a:person_type_i>0</a:person_type_i>
        <a:person_type_s>1</a:person_type_s>
        <a:person_type_pref>S</a:person_type_pref>
        <a:phone>867-5309</a:phone>
        <a:pretest>1</a:pretest>
```

```
<a:student_id_num />
<a:usage_limit>0</a:usage_limit>
<a:user_id>jake</a:user_id>
</a:APIPerson>
</a:Result>
</GetPersonInfoByPersonIDResult>
</GetPersonInfoByPersonIDResponse>
```

SonaGetPersonInfoByStudentID

Description

This function will provide all information from the main user (people) table about specified participants, based on their student_id_num value (University ID number). It is possible in some configurations that more than one participant has the same ID number, in which case all matching participants will be returned. It will only work for participants or multi-role users with participant user type enabled (person_type_s=1).

Input Parameters

Below are the following input parameters for use with the SonaGetPersonInfoByStudentID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
student_id	The student_id_num of the user.

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	invalid student_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetPersonInfoByStudentID?  
api_key=YOUR_API_KEY&student_id=s54321
```


Sample Output

```
<GetPersonInfoByStudentIDResponse>
  <GetPersonInfoByStudentIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments>This is an old student</a:admin_comments>
        <a:approved>1</a:approved>
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>
        <a:classes_needed>0</a:classes_needed>
        <a:creation_date>2015-01-12T00:00:00</a:creation_date>
        <a:credits_earned>4.50</a:credits_earned>
        <a:credits_needed>4.00</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <a:weekly_reminder>1</a:weekly_reminder>
        <a:study_invitation>0</a:study_invitation>
        <a:alt_email>william@myschool.edu</a:alt_email>
        <a:email_needed>0</a:email_needed>
        <a:first_name>William</a:first_name>
        <a:hipaa_ack_date>2015-09-13T09:00:56</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>2015-09-13T09:00:56</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>2015-09-13T09:00:53</a:last_login>
        <a:last_name>Blake</a:last_name>
        <a:limit_date>2015-03-15T08:35:05</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>26</a:login_days>
        <a:login_sent_date>2015-04-21T07:11:00</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>0</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>12340</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_p>0</a:person_type_p>
        <a:person_type_i>0</a:person_type_i>
        <a:person_type_s>1</a:person_type_s>
        <a:person_type_pref>S</a:person_type_pref>
        <a:phone>888-5399</a:phone>
        <a:pretest>1</a:pretest>
```

```
<a:student_id_num>s54321</a:student_id_num>  
<a:usage_limit>0</a:usage_limit>  
<a:user_id>william</a:user_id>  
</a:APIPerson>  
</a:Result>  
</GetPersonInfoByStudentIDResult>  
</GetPersonInfoByStudentIDResponse>
```

SonaGetPersonInfoByUserID

Description

This function will provide all information from the main user (people) table about a particular user, based on their user_id. It will work for any user type, but which fields are populated depends on the user type.

Input Parameters

Below are the following input parameters for use with the SonaGetPersonInfoByUserID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
user_id	The alphanumeric User ID of the user

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid user_id

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetPersonInfoByUserID?api_key=YOUR_API_KEY&user_id=userid

Sample Output

```
<GetPersonInfoByUserIDResponse>
  <GetPersonInfoByUserIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments>Taking course again</a:admin_comments>
        <a:approved>1</a:approved>
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>
        <a:classes_needed>0</a:classes_needed>
        <a:creation_date>2013-01-12T00:00:00</a:creation_date>
        <a:credits_earned>4.50</a:credits_earned>
        <a:credits_needed>4.00</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <a:weekly_reminder>0</a:weekly_reminder>
        <a:study_invitation>0</a:study_invitation>
        <a:alt_email>jake@myschool.edu</a:alt_email>
        <a:email_needed>0</a:email_needed>
        <a:first_name>Jake</a:first_name>
        <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>2013-09-13T09:00:56</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>2013-09-13T09:00:53</a:last_login>
        <a:last_name>Rogers</a:last_name>
        <a:limit_date>2013-03-15T08:35:05</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>16</a:login_days>
        <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>0</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>4321</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_p>0</a:person_type_p>
        <a:person_type_i>0</a:person_type_i>
        <a:person_type_s>1</a:person_type_s>
        <a:person_type_pref>S</a:person_type_pref>
        <a:phone>867-5309</a:phone>
        <a:pretest>1</a:pretest>
```

```
<a:student_id_num />
  <a:usage_limit>0</a:usage_limit>
  <a:user_id>jake</a:user_id>
</a:APIPerson>
</a:Result>
</GetPersonInfoByUserIDResult>
</GetPersonInfoByUserIDResponse>
```

SonaGetCreditBySignupID

Description

This function will provide all information about a particular study signup, with the specified `signup_id`.

Input Parameters

Below are the following input parameters for use with the SonaGetCreditBySignupID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
signup_id	The numeric identifier of the sign-up.

Output Parameters

Will return a single `APICreditItem` class (here can only be one signup per `signup_id`).

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Sign-up (<code>signup_id</code>) not found
Error Code	-9	Credit record not found
Error Code	-10	Invalid <code>signup_id</code>

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetCreditBySignupID?  
api_key=YOUR_API_KEY&signup_id=100
```

Sample Output

```
<GetCreditBySignupIDResponse>
  <GetCreditBySignupIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:class_id>99</a:class_id>
      <a:comments />
      <a:credit_id>100</a:credit_id>
      <a:credit_type>G</a:credit_type>
      <a:date_granted>2013-09-08T00:00:00</a:date_granted>
      <a:email>mike@yourschool.edu</a:email>
      <a:experiment_id>149</a:experiment_id>
      <a:first_name>Mike</a:first_name>
      <a:grantor_id>10144</a:grantor_id>
      <a:last_name>Donlon</a:last_name>
      <a>manual_credit_flag>0</a>manual_credit_flag>
      <a:penalty_credit>0</a:penalty_credit>
      <a:person_id>11622</a:person_id>
      <a:show_credit>1.00</a:show_credit>
      <a:signup_id>100</a:signup_id>
      <a:student_id_num />
      <a:user_id>1671835</a:user_id>
    </a:Result>
  </GetCreditBySignupIDResult>
</GetCreditBySignupIDResponse>
```

SonaGetCreditsByPersonID

Description

This function will provide information on each credit and signup for the specified participant, including non-study credit grants.

Input Parameters

Below are the following input parameters for use with the SonaGetCreditsByPersonID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the participant.

Output Parameters

Will return an array of APICreditItem class.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Person (person_id) not found
Error Code	-9	Credit record not found
Error Code	-10	Invalid person_id

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetCreditsByPersonID?api_key=YOUR_API_KEY&person_id=11498

Sample Output

```
<GetCreditsByPersonIDResponse>
  <GetCreditsByPersonIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APICreditItem>
        <a:class_id>115</a:class_id>
        <a:comments />
        <a:credit_id>1369</a:credit_id>
        <a:credit_type>G</a:credit_type>
        <a:date_granted>2013-11-10T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>jferry@yourschool.edu</a:email>
        <a:experiment_id>159</a:experiment_id>
        <a:first_name>Jake</a:first_name>
        <a:grantor_id>11818</a:grantor_id>
        <a:last_name>Ferry</a:last_name>
        <a>manual_credit_flag>0</a>manual_credit_flag>
        <a:penalty_credit>0</a:penalty_credit>
        <a:person_id>11498</a:person_id>
        <a:show_credit>0.50</a:show_credit>
        <a:signup_id>1369</a:signup_id>
        <a:student_id_num />
        <a:user_id>jferry</a:user_id>
      </a:APICreditItem>
      <a:APICreditItem>
        <a:class_id>115</a:class_id>
        <a:comments />
        <a:credit_id>1554</a:credit_id>
        <a:credit_type>G</a:credit_type>
        <a:date_granted>2013-11-16T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>jferry@yourschool.edu</a:email>
        <a:experiment_id>153</a:experiment_id>
        <a:first_name>Jake</a:first_name>
        <a:grantor_id>10144</a:grantor_id>
        <a:last_name> Ferry </a:last_name>
        <a>manual_credit_flag>0</a>manual_credit_flag>
        <a:penalty_credit>0</a:penalty_credit>
        <a:person_id>11498</a:person_id>
        <a:show_credit>0.50</a:show_credit>
        <a:signup_id>1554</a:signup_id>
        <a:student_id_num />
      </a:APICreditItem>
    </a:Result>
  </a:Result>
</a:Result>
</GetCreditsByPersonIDResponse>
```

```
<a:user_id>jferry</a:user_id>
</a:APICreditItem>
<a:APICreditItem>
  <a:class_id>115</a:class_id>
  <a:comments />
  <a:credit_id>1555</a:credit_id>
  <a:credit_type>G</a:credit_type>
  <a:date_granted>2013-11-16T00:00:00</a:date_granted>
  <a:occur_date>2012-06-26T15:00:08</a:occur_date>
  <a:email>jferry@yourschool.edu</a:email>
  <a:experiment_id>155</a:experiment_id>
  <a:first_name>Jake</a:first_name>
  <a:grantor_id>10144</a:grantor_id>
  <a:last_name>Ferry</a:last_name>
  <a:manual_credit_flag>0</a:manual_credit_flag>
  <a:penalty_credit>0</a:penalty_credit>
  <a:person_id>11498</a:person_id>
  <a:show_credit>0.50</a:show_credit>
  <a:signup_id>1555</a:signup_id>
  <a:student_id_num />
  <a:user_id>jferry</a:user_id>
</a:APICreditItem>
</a:Result>
</GetCreditsByPersonIDResult>
</GetCreditsByPersonIDResponse>
```

SonaGetCreditsByTimeslotID

Description

This function will provide information on every sign up for the specified timeslot.

Input Parameters

Below are the following input parameters for use with the SonaGetCreditsByTimeslotID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
timeslot_id	The numeric identifier of the time slot.

Output Parameters

Will return an array of APICreditItem class.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	timeslot (timeslot_id) not found
Error Code	-9	no sign-ups for timeslot found
Error Code	-10	Invalid timeslot_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetCreditsByTimeslotID?  
api_key=YOUR_API_KEY&timeslot_id=4321
```

Sample Output

```
<GetCreditsByTimeslotIDResponse>
  <GetCreditsByTimeslotIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APICreditItem>
        <a:class_id>99</a:class_id>
        <a:comments />
        <a:credit_id>1749</a:credit_id>
        <a:credit_type>E</a:credit_type>
        <a:date_granted>2013-11-22T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>jsmith@myschool.edu</a:email>
        <a:experiment_id>152</a:experiment_id>
        <a:first_name>Jay</a:first_name>
        <a:grantor_id>11507</a:grantor_id>
        <a:last_name>Smith</a:last_name>
        <a>manual_credit_flag>0</a>manual_credit_flag>
        <a:penalty_credit>0</a:penalty_credit>
        <a:person_id>11570</a:person_id>
        <a:show_credit>0</a:show_credit>
        <a:signup_id>1749</a:signup_id>
        <a:student_id_num />
        <a:user_id>1666134</a:user_id>
      </a:APICreditItem>
      <a:APICreditItem>
        <a:class_id>97</a:class_id>
        <a:comments />
        <a:credit_id>1897</a:credit_id>
        <a:credit_type>G</a:credit_type>
        <a:date_granted>2013-12-09T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>bjones@yourschool.edu</a:email>
        <a:experiment_id>152</a:experiment_id>
        <a:first_name>Bill</a:first_name>
        <a:grantor_id>11423</a:grantor_id>
        <a:last_name>Jones</a:last_name>
        <a>manual_credit_flag>0</a>manual_credit_flag>
        <a:penalty_credit>0</a:penalty_credit>
        <a:person_id>11672</a:person_id>
        <a:show_credit>0.50</a:show_credit>
        <a:signup_id>1897</a:signup_id>
        <a:student_id_num />
      </a:APICreditItem>
    </a:Result>
  </a:Result>
</a:Result>
</GetCreditsByTimeslotIDResponse>
```

```
<a:user_id>4523445</a:user_id>
</a:APICreditItem>
<a:APICreditItem>
  <a:class_id>101</a:class_id>
  <a:comments />
  <a:credit_id>1948</a:credit_id>
  <a:credit_type>G</a:credit_type>
  <a:date_granted>2013-12-09T00:00:00</a:date_granted>
  <a:occur_date>2012-06-26T15:00:08</a:occur_date>
  <a:email>lgoodman@yourschool.edu</a:email>
  <a:experiment_id>152</a:experiment_id>
  <a:first_name>Lisa</a:first_name>
  <a:grantor_id>11423</a:grantor_id>
  <a:last_name>Goodman</a:last_name>
  <a:manual_credit_flag>0</a:manual_credit_flag>
  <a:penalty_credit>0</a:penalty_credit>
  <a:person_id>11661</a:person_id>
  <a:show_credit>0.50</a:show_credit>
  <a:signup_id>1948</a:signup_id>
  <a:student_id_num />
  <a:user_id>1657823</a:user_id>
</a:APICreditItem>
</a:Result>
</GetCreditsByTimeslotIDResult>
</GetCreditsByTimeslotIDResponse>
```

SonaGetCreditsByDateRange

Description

This function will provide information on each credit within a given date range.

Input Parameters

Below are the following input parameters for use with the SonaGetCreditsByDateRange API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
start_date	The start date of the date range to include. Date must be formatted as: YYYY-MM-DD.
end_date	The end date of the date range to include. Date must be formatted as: YYYY-MM-DD.

Output Parameters

Will return an array of APICreditItem class.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid end date
Error Code	-9	Start date is after end date
Error Code	-10	Invalid start date
Error Code	-11	No credit record found for selected date range

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetCreditsByDateRange?api_key=YOUR_API_KEY&start_date=2014-01-01&end_date=2014-12-31

Sample Output

```
<GetCreditsByDateRangeResponse>
  <GetCreditsByDateRangeResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APICreditItem>
        <a:class_id>99</a:class_id>
        <a:comments />
        <a:credit_id>1749</a:credit_id>
        <a:credit_type>E</a:credit_type>
        <a:date_granted>2013-11-22T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>jsmith@myschool.edu</a:email>
        <a:experiment_id>152</a:experiment_id>
        <a:first_name>Jay</a:first_name>
        <a:grantor_id>11507</a:grantor_id>
        <a:last_name>Smith</a:last_name>
        <a>manual_credit_flag>0</a>manual_credit_flag>
        <a:penalty_credit>0</a:penalty_credit>
        <a:person_id>11570</a:person_id>
        <a:show_credit>0</a:show_credit>
        <a:signup_id>1749</a:signup_id>
        <a:student_id_num />
        <a:user_id>1666134</a:user_id>
      </a:APICreditItem>
      <a:APICreditItem>
        <a:class_id>97</a:class_id>
        <a:comments />
        <a:credit_id>1897</a:credit_id>
        <a:credit_type>G</a:credit_type>
        <a:date_granted>2013-12-09T00:00:00</a:date_granted>
        <a:occur_date>2012-06-26T15:00:08</a:occur_date>
        <a:email>bjones@yourschool.edu</a:email>
        <a:experiment_id>152</a:experiment_id>
        <a:first_name>Bill</a:first_name>
        <a:grantor_id>11423</a:grantor_id>
```

```
<a:last_name>Jones</a:last_name>
<a:manual_credit_flag>0</a:manual_credit_flag>
<a:penalty_credit>0</a:penalty_credit>
<a:person_id>11672</a:person_id>
<a:show_credit>0.50</a:show_credit>
<a:signup_id>1897</a:signup_id>
<a:student_id_num />
<a:user_id>4523445</a:user_id>
</a:APICreditItem>
<a:APICreditItem>
  <a:class_id>101</a:class_id>
  <a:comments />
  <a:credit_id>1948</a:credit_id>
  <a:credit_type>G</a:credit_type>
  <a:date_granted>2013-12-09T00:00:00</a:date_granted>
  <a:occur_date>2012-06-26T15:00:08</a:occur_date>
  <a:email>lgoodman@yourschool.edu</a:email>
  <a:experiment_id>152</a:experiment_id>
  <a:first_name>Lisa</a:first_name>
  <a:grantor_id>11423</a:grantor_id>
  <a:last_name>Goodman</a:last_name>
  <a:manual_credit_flag>0</a:manual_credit_flag>
  <a:penalty_credit>0</a:penalty_credit>
  <a:person_id>11661</a:person_id>
  <a:show_credit>0.50</a:show_credit>
  <a:signup_id>1948</a:signup_id>
  <a:student_id_num />
  <a:user_id>1657823</a:user_id>
</a:APICreditItem>
</a:Result>
</GetCreditsByDateRangeResult>
</GetCreditsByDateRangeResponse>
```


SonaGetPersonListByClassID

Description

This API retrieves a list of all participants in the specified course.

Input Parameters

Below are the following input parameters for use with the SonaGetPersonListByClassID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
class_id	The class_id of the course

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid class_id. No record found.

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetPersonListByClassID?  
api_key=YOUR_API_KEY&class_id=1234
```



```

    <a:pretest>0</a:pretest>
    <a:student_id_num />
    <a:study_invitation>0</a:study_invitation>
    <a:usage_limit>0</a:usage_limit>
    <a:user_id>cspru</a:user_id>
    <a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments />
  <a:alt_email>docrivers@nba.com</a:alt_email>
  <a:approved>1</a:approved>
  <a:birthdate>0001-01-01T00:00:00</a:birthdate>
  <a:classes_needed>0</a:classes_needed>
  <a:creation_date>2019-09-19T06:18:05</a:creation_date>
  <a:credits_earned>0</a:credits_earned>
  <a:credits_needed>4.00</a:credits_needed>
  <a:daily_reminder>0</a:daily_reminder>
  <a:email_needed>0</a:email_needed>
  <a:first_name>Doc</a:first_name>
  <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>0001-01-01T00:00:00</a:last_login>
  <a:last_name>Rivers</a:last_name>
  <a:limit_date>0001-01-01T00:00:00</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>0</a:login_days>
  <a:login_sent_date>0001-01-01T00:00:00</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>0</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>9846</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_i>0</a:person_type_i>
  <a:person_type_p>0</a:person_type_p>
  <a:person_type_pref>S</a:person_type_pref>
  <a:person_type_s>1</a:person_type_s>
  <a:phone />
  <a:pretest>0</a:pretest>

```

```
<a:student_id_num />
<a:study_invitation>0</a:study_invitation>
<a:usage_limit>0</a:usage_limit>
<a:user_id>docrivers@hotmail.com</a:user_id>
<a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments />
  <a:alt_email>eric_s@nba.com</a:alt_email>
  <a:approved>1</a:approved>
  <a:birthdate>0001-01-01T00:00:00</a:birthdate>
  <a:classes_needed>0</a:classes_needed>
  <a:creation_date>2019-09-19T06:18:05</a:creation_date>
  <a:credits_earned>0</a:credits_earned>
  <a:credits_needed>4.00</a:credits_needed>
  <a:daily_reminder>0</a:daily_reminder>
  <a:email_needed>0</a:email_needed>
  <a:first_name>Eric</a:first_name>
  <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>0001-01-01T00:00:00</a:last_login>
  <a:last_name>Spoelstra</a:last_name>
  <a:limit_date>0001-01-01T00:00:00</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>0</a:login_days>
  <a:login_sent_date>0001-01-01T00:00:00</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>0</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>9847</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_i>0</a:person_type_i>
  <a:person_type_p>0</a:person_type_p>
  <a:person_type_pref>S</a:person_type_pref>
  <a:person_type_s>1</a:person_type_s>
  <a:phone />
  <a:pretest>0</a:pretest>
  <a:student_id_num />
```

```

    <a:study_invitation>0</a:study_invitation>
    <a:usage_limit>0</a:usage_limit>
    <a:user_id>eric_s@somemail.com</a:user_id>
    <a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments />
  <a:alt_email>johnsad_smith@hotmail.com</a:alt_email>
  <a:approved>1</a:approved>
  <a:birthdate>0001-01-01T00:00:00</a:birthdate>
  <a:classes_needed>0</a:classes_needed>
  <a:creation_date>2020-01-08T16:33:38</a:creation_date>
  <a:credits_earned>0</a:credits_earned>
  <a:credits_needed>4.00</a:credits_needed>
  <a:daily_reminder>0</a:daily_reminder>
  <a:email_needed>0</a:email_needed>
  <a:first_name>John</a:first_name>
  <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>0001-01-01T00:00:00</a:last_login>
  <a:last_name>Smith</a:last_name>
  <a:limit_date>0001-01-01T00:00:00</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>0</a:login_days>
  <a:login_sent_date>0001-01-01T00:00:00</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>0</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>9862</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_i>0</a:person_type_i>
  <a:person_type_p>0</a:person_type_p>
  <a:person_type_pref>S</a:person_type_pref>
  <a:person_type_s>1</a:person_type_s>
  <a:phone />
  <a:pretest>0</a:pretest>
  <a:student_id_num />
  <a:study_invitation>0</a:study_invitation>

```

```
<a:usage_limit>0</a:usage_limit>
<a:user_id>jssdfmi123@yourschool.edu</a:user_id>
<a:weekly_reminder>0</a:weekly_reminder>
</a:APIPerson>
</a:Result>
</GetPersonListByClassIDResult>
</GetPersonListByClassIDResponse>
```

SonaGetTimeslotsByExperimentID

Description

This API retrieves a list of timeslots for a particular study.

Input Parameters

Below are the following input parameters for use with the SonaGetTimeslotsByExperimentID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study
fill_status	fill_status values: A (all – all timeslots, regardless of if there are signups) F (full only – every space in the timeslot is taken) E (empty only – no spaces in the timeslot are taken) S (with signups – at least one space in the timeslot is taken)

Output Parameters

Will return APIStudySchedule class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid experiment_id

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetTimeslotsByExperimentID?  
api_key=YOUR_API_KEY&experiment_id=123&fill_status=A
```

Sample Output

```
<GetTimeslotsByExperimentIDResponse>
  <GetTimeslotsByExperimentIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>123</a:experiment_id>
        <a:location>Miller Hall</a:location>
        <a:num_signed_up>9</a:num_signed_up>
        <a:num_students>10</a:num_students>
        <a:researcher_id>0</a:researcher_id>
        <a:study_name>Physics Study</a:study_name>
        <a:survey_flag>0</a:survey_flag>
        <a:timeslot_date>2012-06-26T09:00:00</a:timeslot_date>
        <a:timeslot_id>282</a:timeslot_id>
        <a:web_flag>0</a:web_flag>
        <a:videoconf_flag>1</a:videoconf_flag>
        <a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
      </a:APIStudySchedule>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>123</a:experiment_id>
        <a:location>Miller Hall</a:location>
        <a:num_signed_up>1</a:num_signed_up>
        <a:num_students>10</a:num_students>
        <a:researcher_id>0</a:researcher_id>
        <a:study_name>Physics Study</a:study_name>
        <a:survey_flag>0</a:survey_flag>
        <a:timeslot_date>2012-10-05T10:00:00</a:timeslot_date>
        <a:timeslot_id>305</a:timeslot_id>
        <a:web_flag>0</a:web_flag>
        <a:videoconf_flag>1</a:videoconf_flag>
        <a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
      </a:APIStudySchedule>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>123</a:experiment_id>
        <a:location>Miller Hall</a:location>
        <a:num_signed_up>3</a:num_signed_up>
        <a:num_students>10</a:num_students>
        <a:researcher_id>0</a:researcher_id>
        <a:study_name>Physics Study</a:study_name>
      </a:APIStudySchedule>
    </a:Result>
  </a:Result>
</a:Result>
</GetTimeslotsByExperimentIDResponse>
```



```
<a:survey_flag>0</a:survey_flag>
<a:timeslot_date>2014-07-15T10:30:00</a:timeslot_date>
<a:timeslot_id>470</a:timeslot_id>
<a:web_flag>0</a:web_flag>
<a:videoconf_flag>1</a:videoconf_flag>
<a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
</a:APIStudySchedule>
</a:Result>
</GetTimeslotsByExperimentIDResult>
</GetTimeslotsByExperimentIDResponse>
```

SonaGetTimeslotsByExperimentIDDateRange

Description

This API retrieves the list of timeslots, for a particular study, within a given date range.

Input Parameters

Below are the following input parameters for use with the SonaGetTimeslotsByExperimentIDDateRange API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study
start_date	Date to use for beginning of listing. Date must be formatted as: YYYY-MM-DD.
end_date	Date to use for end of listing. Date must be formatted as: YYYY-MM-DD.
fill_status	fill_status values: A (all – all timeslots, regardless of if there are signups) F (full only – every space in the timeslot is taken) E (empty only – no spaces in the timeslot are taken) S (with signups – at least one space in the timeslot is taken)

Output Parameters

Will return APISchedule class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Error Code	-8	Invalid experiment_id
Invalid start_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)

Invalid end_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)
Invalid date range (start date is after end date)	[entered value] to [entered value]	start_date must the same as or earlier than end_date

Sample Input

```
https://myschool.sona-
systems.com/services/SonaAPI.svc/SonaGetTimeslotsByExperimentIDDateRange?
api_key=YOUR_API_KEY&experiment_id=123&start_date=2014-01-01&
end_date=2014-12-31&fill_status=A
```

Sample Output

```
<GetTimeslotsByExperimentIDDateRangeResponse>
  <GetTimeslotsByExperimentIDDateRangeResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>123</a:experiment_id>
        <a:location>Wick Bldg.</a:location>
        <a:num_signed_up>3</a:num_signed_up>
        <a:num_students>10</a:num_students>
        <a:researcher_id>0</a:researcher_id>
        <a:study_name>Revenge Study</a:study_name>
        <a:survey_flag>0</a:survey_flag>
        <a:timeslot_date>2014-07-15T10:30:00</a:timeslot_date>
        <a:timeslot_id>470</a:timeslot_id>
        <a:web_flag>0</a:web_flag>
        <a:videoconf_flag>1</a:videoconf_flag>
        <a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
      </a:APIStudySchedule>
      <a:APIStudySchedule>
        <a:duration>30</a:duration>
        <a:experiment_id>123</a:experiment_id>
        <a:location>Wick Bldg.</a:location>
        <a:num_signed_up>2</a:num_signed_up>
        <a:num_students>3</a:num_students>
        <a:researcher_id>0</a:researcher_id>
        <a:study_name>Revenge Study</a:study_name>
        <a:survey_flag>0</a:survey_flag>
```

```
<a:timeslot_date>2014-10-19T11:00:00</a:timeslot_date>
<a:timeslot_id>477</a:timeslot_id>
<a:web_flag>0</a:web_flag>
<a:videoconf_flag>1</a:videoconf_flag>
<a:videoconf_url>https://www.zoom.com/123</a:videoconf_url>
</a:APIStudySchedule>
</a:Result>
</GetTimeslotsByExperimentIDDateRangeResult>
</GetTimeslotsByExperimentIDDateRangeResponse>
```

SonaGetParticipantStats

Description

This API retrieves basic credit and usage information (summarized) for a particular course, or all courses in the system.

Input Parameters

Below are the following input parameters for use with the SonaGetParticipantStats API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
class_id	The class_id of the course. 0 = all courses.

Output Parameters

Will return a single APIParticipantStats result.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid class_id	[entered value]	not a valid class_id
Invalid class_id. No record found	[entered value]	class_id entered is an integer but no record was found

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetParticipantStats?api_key=YOUR_API_KEY&class_id=4

Sample Output

```
<GetParticipantStatsResponse>
  <GetParticipantStatsResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:class_id>4</a:class_id>
      <a:class_name>PSYCH101</a:class_name>
      <a:complete_total>3</a:complete_total>
      <a:course_credits_earned_total>0</a:course_credits_earned_total>
      <a:incomplete_total>1</a:incomplete_total>
      <a:login_total>2</a:login_total>
      <a:no_login_total>2</a:no_login_total>
      <a:no_signups_total>3</a:no_signups_total>
      <a:overall_credits_earned_total>4</a:overall_credits_earned_total>
      <a:overall_credits_needed_total>14</a:overall_credits_needed_total>
    </a:Result>
  </GetParticipantStatsResult>
</GetParticipantStatsResponse>
```

SonaGetStudyStats

Description

This API retrieves basic usage information for a particular study. It's similar to the Timeslot Usage Summary in the web interface.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyStats API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study. For multi-part studies, specify the experiment_id of the first part; results will be based on all parts of the study.

Output Parameters

Will return a single APIStudyStats result.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid experiment_id	[entered value]	not a valid experiment_id

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyStats?  
api_key=YOUR_API_KEY&experiment_id=14
```

Sample Output

```
<GetStudyStatsResponse>
  <GetStudyStatsResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:awaiting_action_count>3</a:awaiting_action_count>
      <a:credits_granted_total>7.00</a:credits_granted_total>
      <a:experiment_id>14</a:experiment_id>
      <a:noshow_count>1</a:noshow_count>
      <a:noshow_excused_count>2</a:noshow_excused_count>
      <a:noshow_unexcused_count>7</a:noshow_unexcused_count>
      <a:open_spaces_future>10</a:open_spaces_future>
      <a:open_spaces_past>0</a:open_spaces_past>
      <a:participated_count>5</a:participated_count>
      <a:study_name>Behavioral Study 1</a:study_name>
      <a:timeslot_count>3</a:timeslot_count>
    </a:Result>
  </GetStudyStatsResult>
</GetStudyStatsResponse>
```


SonaGetStudyStatsByDateRange

Description

This API retrieves basic usage information for a particular study, within a given date range. It's similar to the Timeslot Usage Summary in the web interface.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyStatsByDateRange API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The experiment_id of the study. For multi-part studies, specify the experiment_id of the first part; results will be based on all parts of the study.
start_date	Date to use for beginning of listing. Date must be formatted as: YYYY-MM-DD.
end_date	Date to use for end of listing. Date must be formatted as: YYYY-MM-DD.

Output Parameters

Will return a single APIStudyStats result.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid experiment_id	[entered value]	not a valid experiment_id
Invalid start_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)

Invalid end_date	[entered value]	not a valid date or date is not properly formatted (must be YYYY-MM-DD)
Invalid date range (start date is after end date)	[entered value] to [entered value]	start_date must the same as or earlier than end_date

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyStatsByDateRange?api_key=YOUR_API_KEY&experiment_id=4&start_date=2016-01-01&end_date=2016-12-31

Sample Output

```
<GetStudyStatsByDateRangeResponse>
  <GetStudyStatsByDateRangeResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:awaiting_action_count>0</a:awaiting_action_count>
      <a:credits_granted_total>7.00</a:credits_granted_total>
      <a:experiment_id>4</a:experiment_id>
      <a:noshow_count>0</a:noshow_count>
      <a:noshow_excused_count>0</a:noshow_excused_count>
      <a:noshow_unexcused_count>0</a:noshow_unexcused_count>
      <a:open_spaces_future>0</a:open_spaces_future>
      <a:open_spaces_past>10</a:open_spaces_past>
      <a:participated_count>5</a:participated_count>
      <a:study_name>A Study On Dreams</a:study_name>
      <a:timeslot_count>3</a:timeslot_count>
    </a:Result>
  </GetStudyStatsByDateRangeResult>
</GetStudyStatsByDateRangeResponse>
```

SonaGetInstructorByClassID

Description

This API retrieves the user information for the Instructors assigned to a particular course. A course may have up to two instructors.

Input Parameters

Below are the following input parameters for use with the SonaGetInstructorByClassID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
class_id	The class_id of the course.

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid class_id	-8	not a valid class_id
Invalid person_id	-9	not a valid person_id (no instructors assigned to the course)

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetInstructorByClassID?api_key=YOUR_API_KEY&class_id=2

Sample Output

```
<GetInstructorByClassIDResponse>
  <GetInstructorByClassIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments />
        <a:alt_email>vsmith@yourschool.edu</a:alt_email>
        <a:approved>1</a:approved>
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>
        <a:classes_needed>0</a:classes_needed>
        <a:creation_date>0001-01-01T00:00:00</a:creation_date>
        <a:credits_earned>0</a:credits_earned>
        <a:credits_needed>3.00</a:credits_needed>
        <a:daily_reminder>0</a:daily_reminder>
        <a:weekly_reminder>1</a:weekly_reminder>
        <a:study_invitation>1</a:study_invitation>
        <a:email_needed>0</a:email_needed>
        <a:first_name>Victor</a:first_name>
        <a:hipaa_ack_date>0001-01-01T00:00:00</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>2016-11-08T11:52:52</a:last_login>
        <a:last_name>Smith</a:last_name>
        <a:limit_date>0001-01-01T00:00:00</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>48</a:login_days>
        <a:login_sent_date>0001-01-01T00:00:00</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>0</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>9256</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_i>1</a:person_type_i>
        <a:person_type_p>0</a:person_type_p>
        <a:person_type_pref>I</a:person_type_pref>
        <a:person_type_s>0</a:person_type_s>
        <a:phone>202-244-3993</a:phone>
        <a:pretest>0</a:pretest>
```

```
<a:student_id_num>12345</a:student_id_num>
<a:usage_limit>0</a:usage_limit>
<a:user_id>ints</a:user_id>
</a:APIPerson>
</a:Result>
</GetInstructorByClassIDResult>
</GetInstructorByClassIDResponse>
```

SonaGetPersonListByLoginDate

Description

This function returns a list of users from the people table based on their user type, if they're allowed to login (or not) and their last recorded login date.

Input Parameters

Below are the following input parameters for use with the SonaGetPersonListByLoginDate API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_type	Possible values: A = admin E = researcher P = PI I = instructor S = student (participant) X = all
login_allowed	Possible values: 0 = login not allowed 1 = login allowed
login_date	User's last login date. Date must be formatted as: YYYY-MM-DD.

Output Parameters

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid person_type	-8	blank or non-numeric person_type
Invalid login_allowed	-9	blank or non-numeric login_allowed
Invalid login_date	-10	not a valid date or date is not properly formatted (must be YYYY-MM-DD)

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetPersonListByLoginDate?api_key=YOUR_API_KEY&person_type=X&login_allowed=1&login_date=2017-08-30

Sample Output

```
<GetPersonListByLoginDateResponse>
  <GetPersonListByLoginDateResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIPerson>
        <a:admin_comments>comments go here</a:admin_comments>
        <a:approved>1</a:approved>
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>
        <a:classes_needed>0</a:classes_needed>
        <a:creation_date>2013-01-12T00:00:00</a:creation_date>
        <a:credits_earned>4.50</a:credits_earned>
        <a:credits_needed>4.00</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <a:weekly_reminder>1</a:weekly_reminder>
        <a:study_invitation>1</a:study_invitation>
        <a:alt_email>jake@myschool.edu</a:alt_email>
        <a:email_needed>0</a:email_needed>
        <a:first_name>Jake</a:first_name>
        <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>
        <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
        <a:language_pref>EN</a:language_pref>
        <a:last_login>2017-08-30T09:00:53</a:last_login>
        <a:last_name>Rogers</a:last_name>
        <a:limit_date>2013-03-15T08:35:05</a:limit_date>
        <a:limit_type />
        <a:limited>0</a:limited>
        <a:login_allowed>1</a:login_allowed>
        <a:login_days>16</a:login_days>
        <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>
        <a:noshow_count>0</a:noshow_count>
        <a:office />
        <a:password_change_needed>0</a:password_change_needed>
        <a:password_failed_count>0</a:password_failed_count>
        <a:person_id>4321</a:person_id>
        <a:person_type_a>0</a:person_type_a>
        <a:person_type_e>0</a:person_type_e>
        <a:person_type_p>0</a:person_type_p>
        <a:person_type_i>0</a:person_type_i>
```

```
<a:person_type_s>1</a:person_type_s>
<a:person_type_pref>S</a:person_type_pref>
<a:phone>867-5309</a:phone>
<a:pretest>1</a:pretest>
<a:student_id_num />
<a:usage_limit>500</a:usage_limit>
<a:user_id>jake</a:user_id>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments>comments go here</a:admin_comments>
  <a:approved>1</a:approved>
  <a:birthdate>1975-07-20T00:00:00</a:birthdate>
  <a:classes_needed>0</a:classes_needed>
  <a:creation_date>2013-01-12T00:00:00</a:creation_date>
  <a:credits_earned>4.50</a:credits_earned>
  <a:credits_needed>4.00</a:credits_needed>
  <a:daily_reminder>1</a:daily_reminder>
  <a:weekly_reminder>1</a:weekly_reminder>
  <a:study_invitation>1</a:study_invitation>
  <a:alt_email>miles@myschool.edu</a:alt_email>
  <a:email_needed>0</a:email_needed>
  <a:first_name>Miles</a:first_name>
  <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>0001-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>2017-08-30T09:00:53</a:last_login>
  <a:last_name>Teller</a:last_name>
  <a:limit_date>2013-03-15T08:35:05</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>16</a:login_days>
  <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>0</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>1234</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_p>0</a:person_type_p>
  <a:person_type_i>0</a:person_type_i>
  <a:person_type_s>1</a:person_type_s>
  <a:person_type_pref>S</a:person_type_pref>
```



```
<a:phone>867-5309</a:phone>
<a:pretest>1</a:pretest>
<a:student_id_num />
<a:usage_limit>500</a:usage_limit>
<a:user_id>miles</a:user_id>
</a:APIPerson>
</a:Result>
</GetPersonListByLoginDateResult>
</GetPersonListByLoginDateResponse>
```

SonaGetPersonListByCreationDate

Description

This function returns a list of users from the people table whose creation date is on or after the specified date. It includes their user type and if they're allowed to login (or not).

Input Parameters

Below are the following input parameters for use with the SonaGetPersonListByCreationDate API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_type	Possible values: A = admin E = researcher P = PI I = instructor S = student (participant) X = all
login_allowed	Possible values: 0 = login not allowed 1 = login allowed
creation_date	The date the users' accounts were created on or after. Date must be formatted as: YYYY-MM-DD.

Output Parameters

Will return APIPerson class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid person_type	-8	blank or non-numeric person_type
Invalid login_allowed	-9	blank or non-numeric login_allowed
Invalid creation_date	-10	not a valid date or date is not properly formatted (must be YYYY-MM-DD)

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetPersonListByCreationDate?  
api_key=YOUR_API_KEY&person_type=S&login_allowed=1&creation_date=2017-01-01
```

Sample Output

```
<GetPersonListByCreationDateResponse>  
  <GetPersonListByCreationDateResult>  
    <a:Errors i:nil="true" />  
    <a:Result>  
      <a:APIPerson>  
        <a:admin_comments>good student</a:admin_comments>  
        <a:approved>1</a:approved>  
        <a:birthdate>1975-07-20T00:00:00</a:birthdate>  
        <a:classes_needed>0</a:classes_needed>  
        <a:creation_date>2017-01-01T00:00:00</a:creation_date>  
        <a:credits_earned>4.50</a:credits_earned>  
        <a:credits_needed>4.00</a:credits_needed>  
        <a:daily_reminder>1</a:daily_reminder>  
        <a:weekly_reminder>1</a:weekly_reminder>  
        <a:study_invitation>1</a:study_invitation>  
        <a:alt_email>william@myschool.edu</a:alt_email>  
        <a:email_needed>0</a:email_needed>  
        <a:first_name>William</a:first_name>  
        <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>  
        <a:hipaa_ack_date_nonp>2017-01-01T00:00:00</a:hipaa_ack_date_nonp>  
        <a:language_pref>EN</a:language_pref>  
        <a:last_login>2013-09-13T09:00:53</a:last_login>  
        <a:last_name>Bliss</a:last_name>  
        <a:limit_date>2013-03-15T08:35:05</a:limit_date>  
        <a:limit_type />  
        <a:limited>0</a:limited>  
        <a:login_allowed>1</a:login_allowed>  
        <a:login_days>16</a:login_days>  
        <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>  
        <a:noshow_count>0</a:noshow_count>  
        <a:office />  
        <a:password_change_needed>0</a:password_change_needed>  
        <a:password_failed_count>0</a:password_failed_count>  
        <a:person_id>1122</a:person_id>  
        <a:person_type_a>0</a:person_type_a>  
        <a:person_type_e>0</a:person_type_e>
```

```
<a:person_type_p>0</a:person_type_p>
<a:person_type_i>0</a:person_type_i>
<a:person_type_s>1</a:person_type_s>
<a:person_type_pref>S</a:person_type_pref>
<a:phone>867-5309</a:phone>
<a:pretest>1</a:pretest>
<a:student_id_num />
<a:usage_limit>500</a:usage_limit>
<a:user_id>william</a:user_id>
</a:APIPerson>
<a:APIPerson>
  <a:admin_comments>average student</a:admin_comments>
  <a:approved>1</a:approved>
  <a:birthdate>1975-07-20T00:00:00</a:birthdate>
  <a:classes_needed>0</a:classes_needed>
  <a:creation_date>2017-01-01T00:00:00</a:creation_date>
  <a:credits_earned>4.50</a:credits_earned>
  <a:credits_needed>4.00</a:credits_needed>
  <a:daily_reminder>1</a:daily_reminder>
  <a:weekly_reminder>1</a:weekly_reminder>
  <a:study_invitation>1</a:study_invitation>
  <a:alt_email>henry@myschool.edu</a:alt_email>
  <a:email_needed>0</a:email_needed>
  <a:first_name>Henry</a:first_name>
  <a:hipaa_ack_date>2013-09-13T09:00:56</a:hipaa_ack_date>
  <a:hipaa_ack_date_nonp>2017-01-01T00:00:00</a:hipaa_ack_date_nonp>
  <a:language_pref>EN</a:language_pref>
  <a:last_login>2013-09-13T09:00:53</a:last_login>
  <a:last_name>McCormick</a:last_name>
  <a:limit_date>2013-03-15T08:35:05</a:limit_date>
  <a:limit_type />
  <a:limited>0</a:limited>
  <a:login_allowed>1</a:login_allowed>
  <a:login_days>16</a:login_days>
  <a:login_sent_date>2013-04-21T07:11:00</a:login_sent_date>
  <a:noshow_count>0</a:noshow_count>
  <a:office />
  <a:password_change_needed>0</a:password_change_needed>
  <a:password_failed_count>0</a:password_failed_count>
  <a:person_id>3344</a:person_id>
  <a:person_type_a>0</a:person_type_a>
  <a:person_type_e>0</a:person_type_e>
  <a:person_type_p>0</a:person_type_p>
```

```
<a:person_type_i>0</a:person_type_i>
<a:person_type_s>1</a:person_type_s>
<a:person_type_pref>S</a:person_type_pref>
<a:phone>867-5309</a:phone>
<a:pretest>1</a:pretest>
<a:student_id_num />
<a:usage_limit>500</a:usage_limit>
<a:user_id>henry</a:user_id>
</a:APIPerson>
</a:Result>
</GetPersonListByCreationDateResult>
</GetPersonListByCreationDateResponse>
```

SonaGetStudyInfoByExperimentID

Description

This function returns all information pertaining to a study based on its `experiment_id`.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyInfoByExperimentID API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
experiment_id	The <code>experiment_id</code> of the study. In the case of a multi-part study, each part has a different <code>experiment_id</code> , but most data fields are the same for each part.

Output Parameters

Will return `APIStudyInfo` class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid experiment_id	-8	<code>experiment_id</code> entered is invalid
Experiment_id not found	-9	<code>experiment_id</code> entered is not found

Sample Input

```
https://myschool.sona-  
systems.com/services/SonaAPI.svc/SonaGetStudyInfoByExperimentID?  
api_key=YOUR_API_KEY&experiment_id=144
```

Sample Output

```
<GetStudyInfoByExperimentIDResponse>
  <GetStudyInfoByExperimentIDResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudyInfo>
        <a:active>1</a:active>
        <a:age_maximum>99.0</a:age_maximum>
        <a:age_minimum>0.0</a:age_minimum>
        <a:auto_credit_enable>1</a:auto_credit_enable>
        <a:cancel_limit>24</a:cancel_limit>
        <a:course_restrictions />
        <a:creation_date>2011-01-01T00:00:00</a:creation_date>
        <a:credits>2.00</a:credits>
        <a:double_signup_flag>0</a:double_signup_flag>
        <a:duration>30</a:duration>
        <a:exp_signup_email>2</a:exp_signup_email>
        <a:experiment_id>144</a:experiment_id>
        <a:experiment_url />
        <a:experimenter_timeslot>0</a:experimenter_timeslot>
        <a:has_multipart>1</a:has_multipart>
        <a:has_prereqs>0</a:has_prereqs>
        <a:has_rests>0</a:has_rests>
        <a:irb_code>12345678901</a:irb_code>
        <a:irb_expiration_date>2018-06-17T00:00:00</a:irb_expiration_date>
        <a:is_neg_req>0</a:is_neg_req>
        <a:last_activity>2017-09-04T12:25:41</a:last_activity>
        <a:last_timeslot>2017-08-30T10:30:00</a:last_timeslot>
        <a:long_desc>This is a study</a:long_desc>
        <a:multipart_exact>0</a:multipart_exact>
        <a:multipart_max_days>-1</a:multipart_max_days>
        <a:multipart_min_days>-1</a:multipart_min_days>
        <a:multipart_number>1</a:multipart_number>
        <a:neg_req_any>1</a:neg_req_any>
        <a:parent_experiment_id>0</a:parent_experiment_id>
        <a:payment />
        <a:pi_id>9263</a:pi_id>
        <a:preparation>None</a:preparation>
        <a:prereq_any>0</a:prereq_any>
        <a:previously_approved>1</a:previously_approved>
        <a:private_comments>comments go here</a:private_comments>
        <a:restrictions>None</a:restrictions>
        <a:short_desc>This is a study</a:short_desc>
```

```
<a:show_participants>0</a:show_participants>
<a:signup_limit>24</a:signup_limit>
<a:signup_password />
<a:study_alternative>0</a:study_alternative>
<a:study_name>This is a study on Studies</a:study_name>
<a:survey_anon_flag>0</a:survey_anon_flag>
<a:survey_flag>0</a:survey_flag>
<a:total_credits>8.00</a:total_credits>
<a:total_parts>4</a:total_parts>
<a:total_payment />
<a:type>C</a:type>
<a:usage_limit>10</a:usage_limit>
<a:valid>1</a:valid>
<a:web_flag>0</a:web_flag>
</a:APIStudyInfo>
</a:Result>
</GetStudyInfoByExperimentIDResult>
</GetStudyInfoByExperimentIDResponse>
```


SonaGetStudyListByResearcher

Description

This function returns a detailed list of studies managed by a particular researcher.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyListByResearcher API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
active	-1=all; 0=not active only; 1=active only
approved	-1=all; 0=not approved only; 1=approved only
web_flag	-1=all; 0=lab studies only; 1=web studies only (internal online and external web)
survey_flag	-1=all; 0=all except online surveys; 1=online surveys only
researcher_username	researcher's username

Output Parameters

Will return APIStudyInfo class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid researcher_username	-8	username entered does not exist (as a researcher)

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyListByResearcher?
api_key=YOUR_API_KEY&active=1&approved=1&web_flag=1&survey_flag=1&
researcher_username=john_researcher
```

Sample Output

```
<GetStudyListByResearcherResponse>
  <GetStudyListByResearcherResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudyInfo>
        <a:active>1</a:active>
        <a:age_maximum>99.0</a:age_maximum>
        <a:age_minimum>0.0</a:age_minimum>
        <a:auto_credit_enable>1</a:auto_credit_enable>
        <a:cancel_limit>24</a:cancel_limit>
        <a:course_restrictions />
        <a:creation_date>2011-01-01T00:00:00</a:creation_date>
        <a:credits>2.00</a:credits>
        <a:double_signup_flag>0</a:double_signup_flag>
        <a:duration>30</a:duration>
        <a:exp_signup_email>2</a:exp_signup_email>
        <a:experiment_id>144</a:experiment_id>
        <a:experiment_url />
        <a:experimenter_timeslot>0</a:experimenter_timeslot>
        <a:has_multipart>1</a:has_multipart>
        <a:has_prereqs>0</a:has_prereqs>
        <a:has_rests>0</a:has_rests>
        <a:irb_code>12345678901</a:irb_code>
        <a:irb_expiration_date>2018-06-17T00:00:00</a:irb_expiration_date>
        <a:is_neg_req>0</a:is_neg_req>
        <a:last_activity>2017-09-04T12:25:41</a:last_activity>
        <a:last_timeslot>2017-08-30T10:30:00</a:last_timeslot>
        <a:long_desc>This is a study</a:long_desc>
        <a:multipart_exact>0</a:multipart_exact>
        <a:multipart_max_days>-1</a:multipart_max_days>
        <a:multipart_min_days>-1</a:multipart_min_days>
        <a:multipart_number>1</a:multipart_number>
        <a:neg_req_any>1</a:neg_req_any>
        <a:parent_experiment_id>0</a:parent_experiment_id>
```

```
<a:payment />
<a:pi_id>9263</a:pi_id>
<a:preparation>None</a:preparation>
<a:prereq_any>0</a:prereq_any>
<a:previously_approved>1</a:previously_approved>
<a:private_comments>comments go here</a:private_comments>
<a:restrictions>None</a:restrictions>
<a:short_desc>This is a study</a:short_desc>
<a:show_participants>0</a:show_participants>
<a:signup_limit>24</a:signup_limit>
<a:signup_password />
<a:study_alternative>0</a:study_alternative>
<a:study_name>The Study Name Goes Here</a:study_name>
<a:survey_anon_flag>0</a:survey_anon_flag>
<a:survey_flag>0</a:survey_flag>
<a:total_credits>8.00</a:total_credits>
<a:total_parts>4</a:total_parts>
<a:total_payment />
<a:type>C</a:type>
<a:usage_limit>10</a:usage_limit>
<a:valid>1</a:valid>
<a:web_flag>0</a:web_flag>
</a:APIStudyInfo>
</a:Result>
</GetStudyListByResearcherResult>
</GetStudyListByResearcherResponse>
```

SonaGetStudyListByPI

Description

This function returns a detailed list of studies managed by a particular PI.

Input Parameters

Below are the following input parameters for use with the SonaGetStudyListByPI API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
active	-1=all; 0=not active only; 1=active only
approved	-1=all; 0=not approved only; 1=approved only
web_flag	-1=all; 0=lab studies only; 1=web studies only (internal online and external web)
survey_flag	-1=all; 0=all except online surveys; 1=online surveys only
pi_username	principal investigator's username

Output Parameters

Will return APISStudyInfo class and display data in an array.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid pi_username	-8	username entered does not exist (as a PI)

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaGetStudyListByPI?
api_key=YOUR_API_KEY&active=1&approved=1&web_flag=1&survey_flag=1&pi_username=john_pi

Sample Output

```
<GetStudyListByPIResponse>
  <GetStudyListByPIResult>
    <a:Errors i:nil="true" />
    <a:Result>
      <a:APIStudyInfo>
        <a:active>1</a:active>
        <a:age_maximum>99.0</a:age_maximum>
        <a:age_minimum>0.0</a:age_minimum>
        <a:auto_credit_enable>1</a:auto_credit_enable>
        <a:cancel_limit>24</a:cancel_limit>
        <a:course_restrictions />
        <a:creation_date>2011-01-01T00:00:00</a:creation_date>
        <a:credits>2.00</a:credits>
        <a:double_signup_flag>0</a:double_signup_flag>
        <a:duration>30</a:duration>
        <a:exp_signup_email>2</a:exp_signup_email>
        <a:experiment_id>144</a:experiment_id>
        <a:experiment_url />
        <a:experimenter_timeslot>0</a:experimenter_timeslot>
        <a:has_multipart>1</a:has_multipart>
        <a:has_prereqs>0</a:has_prereqs>
        <a:has_rests>0</a:has_rests>
        <a:irb_code>12345678901</a:irb_code>
        <a:irb_expiration_date>2018-06-17T00:00:00</a:irb_expiration_date>
        <a:is_neg_req>0</a:is_neg_req>
        <a:last_activity>2017-09-04T12:25:41</a:last_activity>
        <a:last_timeslot>2017-08-30T10:30:00</a:last_timeslot>
        <a:long_desc>This is a study</a:long_desc>
        <a:multipart_exact>0</a:multipart_exact>
        <a:multipart_max_days>-1</a:multipart_max_days>
        <a:multipart_min_days>-1</a:multipart_min_days>
        <a:multipart_number>1</a:multipart_number>
        <a:neg_req_any>1</a:neg_req_any>
        <a:parent_experiment_id>0</a:parent_experiment_id>
        <a:payment />
        <a:pi_id>9263</a:pi_id>
        <a:preparation>None</a:preparation>
```

```
<a:prereq_any>0</a:prereq_any>
<a:previously_approved>1</a:previously_approved>
<a:private_comments>comments go here</a:private_comments>
<a:restrictions>None</a:restrictions>
<a:short_desc>This is a study</a:short_desc>
<a:show_participants>0</a:show_participants>
<a:signup_limit>24</a:signup_limit>
<a:signup_password />
<a:study_alternative>0</a:study_alternative>
<a:study_name>The Study Name Goes Here</a:study_name>
<a:survey_anon_flag>0</a:survey_anon_flag>
<a:survey_flag>0</a:survey_flag>
<a:total_credits>8.00</a:total_credits>
<a:total_parts>4</a:total_parts>
<a:total_payment />
<a:type>C</a:type>
<a:usage_limit>10</a:usage_limit>
<a:valid>1</a:valid>
<a:web_flag>0</a:web_flag>
</a:APIStudyInfo>
</a:Result>
</GetStudyListByPIResult>
</GetStudyListByPIResponse>
```

SonaActivateUser

Description

This function activates a user's inactive account. If the user's account is already active, the function's numeric output will indicate as such. This is the `login_allowed` field in the user's account.

Input Parameters

Below are the following input parameters for use with the SonaActivateUser API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the user.

Output Parameters

Will return an integer value of 1 indicating a successful activation, or 2 indicating that the user's account is already activated.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid person_id	-8	person_id entered is invalid
Person_id not found	-9	person_id entered is not found

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaActivateUser?api_key=YOUR_API_KEY&person_id=15

Sample Output

```
<ActivateUserResponse xmlns="https://www.sona-systems.com/">
  <ActivateUserResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasytems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>1</a:Result>
  </ActivateUserResult>
</ActivateUserResponse>
```


SonaCreateUser

Description

This function allows for the creation of a new user in the system by providing necessary details such as name, user ID, and person type.

Input Parameters

Below are the following input parameters for use with the SonaCreateUser API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
first_name	User's first name
last_name	User's last name
user_id	Unique identifier for the user. If an SSO user (<code>sso_user=1</code>), then this is the user's SSO ID.
person_type	Possible values: E = researcher P = PI I = instructor S = participant
sso_user	Specifies if the user is an SSO user: 1 = yes 0 = no
password	User's password. Leave blank for auto-assigned password. Not applicable for SSO users. Optional.
alt_email	User's alternate email address (or main email address, if @ suffix is not enabled in System Settings, or SSO user). Optional for non-SSO users when @ suffix is enabled.
birthdate	User's date of birth. YYYY-MM-DD format. Only applies to participants, only if date of birth features is enabled. Optional.
phone	User's phone number. Optional.
student_id_num	User's student ID/university ID number. Optional.
office	User's office location. Only applies to non-participant users. Optional.
admin_comments	Comments seen by administrators only. Optional.

course_list	An optional comma-separated list of numeric course identifiers relevant only when person_type = "S" (Participant). If no valid courses are provided, the system flags the user as needing to choose courses on next login. This field is ignored for non-participants.
login_allowed	Is user allowed to log in? 0/1 value. Default 1.
send_login_info	Send login information to user by email. 1/0 value.
language_pref	User's preferred language (2-char or 5-char abbreviation): EN = English, ZH = Simplified Chinese, JA = Japanese, KO = Korean, FR = French, PT = Portuguese, DE = German, etc. The default language for the system will be used if not specified. Only languages enabled on the site are allowed. Optional.

Output Parameters

Will return an integer value of the unique person_id of the newly created user, if successful.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid input parameter	-15	Input validation failures such as missing fields, invalid formats, or conflicts like duplicate IDs or unsupported configurations. Correct the input to resolve the issue before retrying.
Error adding user	-16	An unexpected issue occurred while processing your request, such as a system or database error. Please check the error message for details and contact support if the problem persists

Sample Input

```
https://myschool.sona-systems.com/services/SonaAPI.svc/SonaCreateUser?
api_key=YOUR_API_KEY&
first_name=John&last_name=Doe&user_id=jdoe123&person_type=S&sso_user=0&
password=password&alt_email=jdoe@example.com&birthdate=1970-01-01&
phone=123-456-7890&student_id_num=123456789&office=&
admin_comments=New%20student%20enrollment&course_list=123,456&
login_allowed=1&send_login_info=0&language_pref=EN
```

Sample Output

```
<SonaCreateUserResponse xmlns="https://www.sona-systems.com/">
  <SonaCreateUserResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>42</a:Result>
  </SonaCreateUserResult>
</SonaCreateUserResponse>
```

SonaDeactivateUser

Description

This function deactivates a user's active account. If the user's account is already deactivated, the function's numeric output will indicate as such. This is the login_allowed field in the user's account.

Input Parameters

Below are the following input parameters for use with the SonaDeactivateUser API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the user.

Output Parameters

Will return an integer value of 1 indicating a successful deactivation, or 2 indicating that the user's account is already deactivated.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid person_id	-8	person_id entered is invalid
Person_id not found	-9	person_id entered is not found

Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/SonaDeactivateUser?
api_key=YOUR_API_KEY&person_id=15](https://myschool.sona-systems.com/services/SonaAPI.svc/SonaDeactivateUser?api_key=YOUR_API_KEY&person_id=15)

Sample Output

```
<DeactivateUserResponse xmlns="https://www.sona-systems.com/">
  <DeactivateUserResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:Errors i:nil="true"
      xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
    <a:Result>1</a:Result>
  </DeactivateUserResult>
</DeactivateUserResponse>
```

SonaApproveUser

Description

This function approves a participant's user account. If the user's account is already approved, the function's numeric output will indicate as such. This is the approved field in the user's account, and only applies if Automatic Account Approval is turned off in System Settings. Unlike approving a user from the Approve Users page, this function does *not* send the user an email that their account has been approved.

Input Parameters

Below are the following input parameters for use with the SonaApproveUser API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the user.

Output Parameters

Will return an integer value of 1 indicating a successful approval, or 2 indicating that the user's account is already approved.

Error Codes

Below are the possible error messages encountered and their respective descriptions.

Error Type	Error Code	Description
Authentication Failed	-1 through -7	Standard authentication errors
Invalid person_id	-8	person_id entered is invalid
Person_id not found	-9	person_id entered is not found

Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/SonaApproveUser?
api_key=YOUR_API_KEY&person_id=18](https://myschool.sona-systems.com/services/SonaAPI.svc/SonaApproveUser?api_key=YOUR_API_KEY&person_id=18)

Sample Output

```
<ApproveUserResponse xmlns="https://www.sona-systems.com/">
  <ApproveUserResult
    xmlns:a="http://schemas.datacontract.org/2004/07/emsdotnet.sonasystems"
    xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
      <a:Errors i:nil="true"
        xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
      <a:Result>2</a:Result>
    </ApproveUserResult>
  </ApproveUserResponse>
```

SonaSetUserAdminComments

Description

This function appends, overwrites or deletes a user's Admin Comments data (admin_comments field) (500-character maximum length limit). If the user currently has an admin_comments record, deleting it will revert the record back to NULL state. A numeric value will be returned in the process to indicate the result.

Input Parameters

Below are the following input parameters for use with the SonaSetUserAdminComments API function.

Parameter	Description / Possible Values
api_key	Use API key generated for Sona admin user
person_id	The numeric identifier of the user.
action	The action to be taken. D = delete A = append O = overwrite/add
comments	The text comment to be added or appended. Ignored if action=D

Output Parameters

The function returns integer values to indicate the outcome of the operation. Positive integers represent success, while negative integers represent errors. The following table lists all the success and error codes:

Type	Return Value	Description
Authentication Failed	-1 through -7	Standard authentication errors
Success Code	1	The comment has been added or the previous comment overwritten.
Success Code	2	The comment has been added or the previous comment overwritten, but some characters were stripped off due to length limits.
Success Code	3	The comment has been appended.

Success Code	4	The comment has been appended, but some characters were stripped off due to length limits.
Success Code	5	The comment has been deleted.
Error Code	-8	An invalid person_id was used.
Error Code	-9	No matching user was found.
Error Code	-10	An invalid action.

Sample Input

https://myschool.sona-systems.com/services/SonaAPI.svc/SonaSetUserAdminComments?api_key=YOUR_API_KEY&person_id=15&action=0&comments=place comment here

Sample Output

```
<SetUserAdminCommentsResponse>
  <SetUserAdminCommentsResult>
    <a:Errors i:nil="true" />
    <a:Result>1</a:Result>
  </SetUserAdminCommentsResult>
</SetUserAdminCommentsResponse>
```

WebStudyCreditGrant

This is explained in detail on the “Server-Side Completion URL Responses” section (page 51) of the Sona documentation. That section contains descriptions, input parameters, error codes and samples pertaining to this API function.

API Data Classes

APICourse

This class is used to output a course. The data derived from this class are as follows:

Data Field	Data Type	Description
class_id	Int32	the numeric identifier of the course
class_name	String	the course's name
instructor_id	Int32	The instructor's person ID. 0 means no instructor selected
instructor_id2	Int32	A second instructor's person ID, if applicable. 0 means no instructor selected
credits_required	Decimal	the unique numeric identifier of the credit record
selectable	Int32	possible values: 0 = participants cannot select this course 1 = participants can select this course

APICreditItem

This class is used to output an array of data containing taken from the credits table. The data derived from this class are as follows:

Data Field	Data Type	Description
signup_id	Int32	the numeric identifier of the sign-up
first_name	String	the participant's first name
last_name	String	the participant's last name
student_id_num	String	the participant's student ID number (if available)
credit_id	Int32	the unique numeric identifier of the credit record
user_id	String	the participant's user ID
email	String	the participant's email address
person_id	Int32	the numeric identifier of the participant
date_granted	Date	the date when the credit was granted
occur_date	Date	the date and time when the study is to occur
show_credit	Decimal	the credit value granted to the participant
penalty_credit	Decimal	the credit value penalized from the participant
credit_type	String	possible values: G = credit granted W = awaiting action E = excused no-show I = unexcused no-show
manual_credit_flag	Int32	possible values: 0 = system-granted 1 = admin-granted
comments	String	comments pertaining to the credit grant
experiment_id	Int32	the numeric identifier of the study

grantor_id	Int32	the numeric identifier of the researcher who granted credit
class_id	Int32	the numeric identifier of the course

APISignup

This class is used to output an array of data containing a list of signup records for a particular timeslot. The data derived from this class are as follows:

Data Field	Data Type	Description
signup_id	Int32	the numeric identifier of the sign-up
first_name	String	the participant's first name
last_name	String	the participant's last name
student_id_num	String	the participant's student ID number (if available)
phone	String	the participant's phone number (if available)
user_id	String	the participant's user ID
email	String	the participant's email address
anon_id_code	String	will apply only when Anonymous ID codes are enabled, otherwise blank
survey_code	String	will apply only when survey_code is used for an external web study, otherwise blank
credit_status	String	possible values: G = credit granted W = awaiting action N = no-show
signup_date	Date	the date the participant signed up for the study
link_click_date	Date	the date when the participant clicked the study link. Applies to external web studies and videoconference timeslots only.
occur_date	Date	the date and time when the study is to occur

APIStudySchedule

This class is used to output an array of data typically found in the Study Schedule Report. The data retrieved from this class are as follows:

Data Field	Data Type	Description
experiment_id	Int32	the numeric identifier of the study
timeslot_id	Int32	the numeric identifier of the timeslot
study_name	String	the name of the study
location	String	the location of the study
timeslot_date	Date	the date and time when the study is to occur
duration	Int32	the duration (in minutes) of the study. Add this to timeslot_date to find end time for the timeslot.
web_flag	Int32	0 = standard (lab) study 1 = internal online survey or external web study
survey_flag	Int32	0 = standard (lab) or external web study 1 = internal online survey
num_signed_up	Int32	the total number of participants signed up for this study.
num_students	Int32	the total number of participants allowed for this timeslot. This is basically the number set when the timeslot is created (its capacity), and is not reduced when participants sign up for the timeslot.
researcher_id	Int32	the numeric identifier for the researcher assigned to this particular timeslot, if applicable. If not applicable, then this value will be 0. This will be set to a non-zero value when all of the following are true: Study has multiple researchers. Study is configured to allow researchers at timeslot-level.

		A particular researcher is assigned to this timeslot.
videoconf_flag	Int32	0 = not a videoconference (default) 1 = videoconference
videoconf_url	String	the URL of the videoconference

APIStudyInfo

This contains general information about a particular study. The data derived from this class are as follows:

Data Field	Data Type	Description
experiment_id	Int32	the numeric identifier of the study
study_name	String	the name of the study
short_desc	String	the short description of the study
long_desc	String	the long description of the study
private_comments	String	admin-only comments
restrictions	String	
pi_id	Int32	the numeric identifier of the PI
duration	Int32	duration (in mins.) of the study
preparation	String	
credits	Decimal	the credits for this part of the study (0 if a paid study)
type	String	C = credit study M = paid study
valid	Int32	1 = valid 2 = invalid
previously_approved	Int32	indicates whether the study was previously approved
active	Int32	1 = active 2 = inactive
exp_signup_email	Int32	
has_prereqs	Int32	
has_rests	Int32	

is_neg_req	Int32	
prereq_any	Int32	
neg_req_any	Int32	
course_restrictions	String	
last_timeslot	Date	the last timeslot for the study
double_signup_flag	Int32	1 = duplicate signups allowed 2 = no duplicate signups
signup_password	String	the invitation code
web_flag	Int32	0 = standard (lab) study 1 = internal online survey or external web study
survey_flag	Int32	0 = standard (lab) or external web study 1 = internal online survey
survey_anon_flag	Int32	
experiment_url	String	the URL for the external study
irb_code	String	the IRB code
irb_expiration_date	Date	the IRB expiration date
has_multipart	Int32	1 = multipart study 0 = non-multipart study
multipart_exact	Int32	0 = can occur at any time compared to previous part 1 = must occur at same time as previous part
multipart_number	Int32	represents which part, of a multipart study, the study belongs to
parent_experiment_id	Int32	null if first part of single-part study, otherwise experiment_id of part 1

multipart_min_days	Int32	the minimum number of days of separation between this part of the study and the previous part (applies to multipart studies only)
multipart_max_days	Int32	the maximum number of days of separation between this part of the study and the previous part (applies to multipart studies only)
total_parts	Int32	the number of parts the study has
total_credits	Decimal	total credits for the entire study. Will be the same as the credits field, except in cases of multi-part studies. 0 for paid studies.
usage_limit	Int32	timeslot usage limit (quota) for study, in hours
signup_limit	Int32	signup deadline (in hours)
cancel_limit	Int32	cancellation deadline (in hours)
auto_credit_enable	Int32	1 = auto crediting enabled 0 = disabled
experimenter_timeslot	Int32	1 = researchers can be assigned at timeslot-level.
show_participants	Int32	(not implemented)
study_alternative	Int32	1 = research alternative 0 = not a research alternative
creation_date	Date	date when study was created. May be empty for older studies.
last_activity	Date	date when the study last underwent any changes
payment	String	the payment for this part of the study (empty if a credit study)

total_payment	String	total payment for the entire study. Will be the same as the payment field, except in cases of multi-part studies. Empty for credit studies.
age_minimum	Decimal	Minimum age to sign up for the study. 0 = no age limits.
age_maximum	Decimal	Maximum age to sign up for the study. Ignore if age_minimum=0.

APIPrescreenSurveyData

This class contains general information about one participant's set of responses for a prescreen or online survey. The data derived from this class are as follows:

Data Field	Data Type	Description
pretest_id	Int32	the numeric identifier of the prescreen
person_id	Int32	the numeric identifier of the participant
start_time	Date	the start time of the prescreen/study
end_time	Date	the end time of the prescreen/study
status	String	completion status. Possible values: C = completed W = withdrew I = in progress
data_status	String	status of response data. Possible values: D = data deleted S = stored on system
first_name	String	the first name of the participant
last_name	String	the last name of the participant
user_id	String	the participant's user ID
anon_id_code	String	applies only when Anonymous ID codes are enabled, otherwise blank
responses	Array	the participant's responses to this prescreen/survey, an array of APIPrescreenSurveyResponseItem items

APIPrescreenSurveyResponseItem

This class represents one response by one participant for one question. If the question allows multiple responses, then there will be an entry for each response provided. The data derived from this class are as follows:

Data Field	Data Type	Description
response_id	Int32	the unique identifier for the response
question_id	Int32	the question this is a response for
choice_id	Int32	the choice chosen
response_text	String	the text representation of the response
response_numeric	Decimal	the numeric representation of the response (applies to numeric questions only)
opt_out_choice	Int32	1 = response was to opt out of the question 0 = opted in

APIPerson

This class is used to output an array of data containing taken from the people table. The data derived from this class are as follows:

Data Field	Data Type	Description
person_id	Int32	the person_id of the user
first_name	String	the user's first name
last_name	String	the user's last name
user_id	String	the user's user ID
student_id_num	String	the user's student ID number (if available)
phone	String	the user's phone number (if available)
office	String	the user's office location (if available)
alt_email	String	the user's alternate email address (or main email address, if @ Suffix is not enabled in System Settings)
person_type_a	Int32	possible values: 1 = admin 0 = not an admin
person_type_e	Int32	possible values: 1 = researcher 0 = not a researcher
person_type_p	Int32	possible values: 1 = PI 0 = not a PI
person_type_i	Int32	possible values: 1 = instructor 0 = not an instructor
person_type_s	Int32	possible values: 1 = student (participant) 0 = not a student (participant)
person_type_pref	String	possible values: A = admin E = researcher P = PI I = instructor S = student (participant)

credits_earned	Decimal	the total credits earned by the participant
credits_needed	Decimal	the total credits needed (requirements) by the participant
pretest	Int32	Whether the participant has completed the prescreen. possible values: 0 = prescreen not taken 1 = prescreen completed 2 = opted out 3 = currently taking prescreen
login_allowed	Int32	possible values: 0 = login not allowed 1 = login allowed
approved	Int32	possible values: 0 = account not approved 1 = account approved
admin_comments	String	comments seen by administrators only
hipaa_ack_date	Date	the date when the participant acknowledged privacy policy
hipaa_ack_date_nonp	Date	the date when the non-participant user role acknowledged privacy policy
login_sent_date	Date	the date when the user's login info was last sent
daily_reminder	Int32	possible values: 0 = no daily reminders 1 = receives daily reminders
weekly_reminder	Int32	possible values: 0 = no weekly reminders 1 = receives weekly reminders
study_invitation	Int32	possible values: 0 = no study invitations 1 = receives study invitations
classes_needed	Int32	possible values: 0 = already in at least one course 1 = needs to join a course
email_needed	Int32	possible values: 0 = email not needed 1 = email needed (currently not implemented)

password_change_needed	Int32	possible values: 0 = password change not required 1 = password change required on next login
password_failed_count	Int32	the number of failed password login attempts
last_login	Date	the user's last login date
creation_date	Date	the date when the user's account was created (if available)
login_days	Int32	the number of unique days the user has logged into the system
noshow_count	Int32	the number of unexcused no-shows accrued
limited	Int32	possible values: 0 = not limited to research alternatives 1 = limited to research alternatives
limit_date	Date	the date when the participant's account was limited
limit_type	String	possible values: M = manual (administrator set the limit) A = automatic (limit was set based on no-show accrual)
language_pref	String	the user's preferred language (2-char abbreviation): EN = English, ZH = Simplified Chinese, JA = Japanese, KO = Korean, FR = French, PT = Portuguese, DE = German
birthdate	Date	the date when the user was born
usage_limit	Int32	timeslot usage limit (quota) for study, in hours

APIParticipantStats

This class is used to display participant statistics (mostly based on Credit Completion Report). The data derived from this class are as follows:

Data Field	Data Type	Description
class_id	Int32	the class_id of the course (0 if for all courses together)
class_name	String	the name of the course (blank if no course specified)
complete_total	Int32	total participants where their overall credits_earned >= overall credits_needed
incomplete_total	Int32	total participants where overall credits_earned < overall credits_needed
no_signups_total	Int32	total participants with no study signups at all
no_login_total	Int32	total participants who have never logged in
login_total	Int32	total who have logged in
overall_credits_earned_total	Decimal	sum of overall credits_earned
overall_credits_needed_total	Decimal	sum of overall credits_needed
course_credits_earned_total	Decimal	sum of credits earned for course, net of no-show penalties [only if course specified]

APIStudyStats

This class is used to display study statistics (mostly based on Timeslot Usage Summary). The data derived from this class are as follows:

Data Field	Data Type	Description
experiment_id	Int32	the experiment_id of the study
study_name	String	the name of the study
timeslot_count	Int32	total number of timeslots (number of timeslot_id's)
noshow_unexcused_count	Int32	total number of unexcused no-shows
noshow_excused_count	Int32	total number of excused no-shows
noshow_count	Int32	total number of no-shows (sum of noshow_unexcused_count and noshow_excused_count)
awaiting_action_count	Int32	total number of sign-ups in awaiting action status
participated_count	Int32	total number of sign-ups with participation marked (credit granted)
credits_granted_total	Decimal	total number of credits granted (credit studies only and does not include no-show penalties)
open_spaces_future	Int32	total number of open spaces in timeslots in the future (within date range, when specified)
open_spaces_past	Int32	total number of open spaces in timeslots in the past (within date range, when specified)

Revision History

Date of Change	Summary of Change
15 February 2022	<ul style="list-style-type: none">• Added revision history• Added new API functions: <i>GetPersonListByClassID</i>, <i>GetCoursesByPersonID</i>, <i>GetPersonInfoByStudentID</i>, <i>GetPersonInfoByName</i>, <i>GrantCreditBySignupID</i>, <i>ConvertUserToSSOByPersonID</i>, <i>ConvertUserToSSOByUserID</i>• Added example of POSTing with Curl• Formatting updates
25 April 2022	<ul style="list-style-type: none">• Formatting updates
29 January 2023	<ul style="list-style-type: none">• Added direct WSDL reference• Updated sample output for Decimal fields (credit value fields)• Updated any reference to HTTP calls to HTTPS
9 March 2023	<ul style="list-style-type: none">• Added notes about requirement of User-Agent header
6 July 2023	<ul style="list-style-type: none">• Fixed typos in <i>GetCreditsByTimeslotID</i> example
10 August 2024	<ul style="list-style-type: none">• Updated to use <i>api_key</i> for authentication instead of username/password (which has been deprecated)• Added Sona prefix to all function names• Added new API functions <i>SonaGetCreditsByDateRange</i> and <i>SonaGetSignupsByDateRange</i>• Updated <i>APICreditItem</i> and <i>APISignUp</i> to include new field <i>occur_date</i>
3 October 2024	<ul style="list-style-type: none">• Added code examples for various languages.
28 November 2024	<ul style="list-style-type: none">• Added new API function <i>SonaCreateUser</i>

5 January 2025	Updated revision history to add in new API functions that were previously added: <ul data-bbox="568 294 1136 483" style="list-style-type: none">• <i>SonaGetPrescreenDataByExperimentID</i>• <i>SonaGetPrescreenDataByTimeslotID</i>• <i>SonaSignUpUserToATimeslot</i>• <i>SonaGrantCreditBySignupID</i>
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------