



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

**Last updated 14 March 2018**

**Copyright © 2018 Sona Systems, Ltd., All Rights Reserved**

## Table of Contents

Introduction .....	7
Intended Audience .....	7
About This Documentation .....	7
Obtaining Technical Support .....	7
Release Schedule .....	7
Improvements and Suggestions .....	8
API Introduction .....	8
Access and URL .....	8
Authentication .....	8
Authentication Errors .....	8
URL GET .....	9
SOAP .....	9
Making an HTTP POST Request .....	9
Performance Considerations .....	10
Testing Tools .....	10
Security Concerns .....	10
API Functions .....	10
GetSignupsForTimeslot .....	10
Description .....	10
Input Parameters .....	10
Output Parameters .....	11
Error Codes .....	11
Sample Input .....	11
Sample Output .....	11
GetStudyScheduleList .....	12
Description .....	12
Input Parameters .....	12
Output Parameters .....	12
Error Codes .....	12
Sample Input .....	13
Sample Output .....	13
GetStudyList .....	13
Description .....	13
Input Parameters .....	14
Output Parameters .....	14
Error Codes .....	14
Sample Input .....	14
Sample Output .....	14
GetResearcherIDByExperimentID .....	16
Description .....	16
Input Parameters .....	16

Output Parameters .....	16
Error Codes .....	16
Sample Input .....	16
Sample Output .....	16
GetPrescreenSurveyDataByPersonID .....	16
Description .....	16
Input Parameters .....	17
Output Parameters .....	17
Error Codes .....	17
Sample Input .....	17
Sample Output .....	17
GetPrescreenSurveyDataByDateRange .....	18
Description .....	18
Input Parameters .....	18
Output Parameters .....	19
Error Codes .....	19
Sample Input .....	19
Sample Output .....	19
GetStudyCreditAssignmentMismatch .....	21
Description .....	21
Input Parameters .....	21
Output Parameters .....	21
Error Codes .....	21
Sample Input .....	21
Sample Output .....	21
FixStudyCreditAssignmentMismatch .....	22
Description .....	22
Input Parameters .....	23
Output Parameters .....	23
Error Codes .....	23
Sample Input .....	23
Sample Output .....	23
GetPersonInfoByPersonID .....	23
Description .....	23
Input Parameters .....	23
Output Parameters .....	24
Error Codes .....	24
Sample Input .....	24
Sample Output .....	24
GetPersonInfoByUserID .....	25
Description .....	25
Input Parameters .....	25
Output Parameters .....	25
Error Codes .....	25
Sample Input .....	25
Sample Output .....	26

GetCreditBySignupID.....	26
Description.....	26
Input Parameters .....	26
Output Parameters.....	27
Error Codes .....	27
Sample Input .....	27
Sample Output .....	27
GetCreditsByPersonID .....	28
Description.....	28
Input Parameters .....	28
Output Parameters.....	28
Error Codes .....	28
Sample Input .....	28
Sample Output .....	28
GetCreditsByTimeslotID .....	29
Description.....	29
Input Parameters .....	29
Output Parameters.....	30
Error Codes .....	30
Sample Input .....	30
Sample Output .....	30
GetTimeslotsByExperimentID .....	31
Description.....	31
Input Parameters .....	31
Output Parameters.....	32
Error Codes .....	32
Sample Input .....	32
Sample Output .....	32
GetTimeslotsByExperimentIDDateRange.....	33
Description.....	33
Input Parameters .....	33
Output Parameters.....	33
Error Codes .....	33
Sample Input .....	34
Sample Output .....	34
GetParticipantStats.....	34
Description.....	34
Input Parameters .....	34
Output Parameters.....	35
Error Codes .....	35
Sample Input .....	35
Sample Output .....	35
GetStudyStats.....	35
Description.....	35
Input Parameters .....	35
Output Parameters.....	36

Error Codes .....	36
Sample Input .....	36
Sample Output .....	36
GetStudyStatsByDateRange .....	36
Description .....	36
Input Parameters .....	36
Output Parameters .....	37
Error Codes .....	37
Sample Input .....	37
Sample Output .....	37
GetInstructorByClassID .....	38
Description .....	38
Input Parameters .....	38
Output Parameters .....	38
Error Codes .....	38
Sample Input .....	38
Sample Output .....	38
GetPersonListByLoginDate .....	39
Description .....	39
Input Parameters .....	39
Output Parameters .....	39
Error Codes .....	39
Sample Input .....	40
Sample Output .....	40
GetPersonListByCreationDate .....	41
Description .....	41
Input Parameters .....	41
Output Parameters .....	42
Error Codes .....	42
Sample Input .....	42
Sample Output .....	42
GetStudyInfoByExperimentID .....	44
Description .....	44
Input Parameters .....	44
Output Parameters .....	44
Error Codes .....	44
Sample Input .....	44
Sample Output .....	44
ActivateUser .....	45
Description .....	45
Input Parameters .....	45
Output Parameters .....	46
Error Codes .....	46
Sample Input .....	46
Sample Output .....	46
DeactivateUser .....	46

Description .....	46
Input Parameters .....	46
Output Parameters .....	46
Error Codes .....	46
Sample Input .....	47
Sample Output .....	47
SetUserAdminComments .....	47
Description .....	47
Input Parameters .....	47
Output Parameters .....	47
Error Codes .....	48
Sample Input .....	48
Sample Output .....	48
WebStudyCreditGrant .....	48
API Data Classes .....	48
APICreditItem .....	48
APISignup .....	49
APIStudySchedule .....	49
APIStudyInfo .....	50
APIPrescreenSurveyData .....	51
APIPrescreenSurveyResponseItem .....	52
APIPerson .....	52
APIParticipantStats .....	53
APIStudyStats .....	54

## 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 (unlike our main software), but instead 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

### Access and URL

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

If you go to that URL (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](http://www.sona-systems.com) so you may need to modify this to your site's own URL.

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

### Authentication

All API functions require a username and password. Currently, you may specify any valid username and password for your site, which is set up as an administrator account and is allowed to login to the system in general. You can view and set these accounts up in Users | Add/Edit/Search on your main site.

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	login not allowed (account disabled)
-4	user not found



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

## URL 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.

## 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 HTTP. 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 HTTP GET call.

## Making an HTTP POST Request

Listed below is a sample HTTP POST request to the API for the GetPersonInfoByUserID function, assuming your site name is yoursite.sona-systems.com. Some items to note are highlighted in yellow in the example:

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

HTTP/1.1 100 Continue

```
<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">http://www.sona-
systems.com/SonaAPI/GetPersonInfoByUserID_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>
<GetPersonInfoByUserID_Post
xmlns="http://www.sona-systems.com/">
<username>admin</username> <password>XXX</password>
<user_id>YYY</user_id>
</GetPersonInfoByUserID_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.

## Performance Considerations

Currently, the system does not employ any throttling mechanisms to limit the number of API calls within a period of time. Regardless, you should design your programs to use the API judiciously and avoid unnecessary calls.

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.

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.

## Testing Tools

There are a number of tools out there for testing your API calls to determine 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

If you use the URL GET method to do API calls, then the username and password are part of the URL. There is the chance that proxy servers within your network will keep a log or cache of this call since it is part of the URL. Additionally, our own servers keep a log of the parameters used in any GET request, as part of regular website logging. For this reason, you may prefer to use SOAP calls to access the API, as then the credentials are not included in the URL and are less likely to be logged or cached.

## API Functions

### GetSignupsForTimeslot

#### 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 GetSignupsForTimeslot API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.

password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
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/GetSignUpsForTimeslot?username=adminusername&password=adminpassword&timeslot\\_id=75](https://myschool.sona-systems.com/services/SonaAPI.svc/GetSignUpsForTimeslot?username=adminusername&password=adminpassword&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:student_id_num>09341404</a:student_id_num>
        <a:survey_code>222</a:survey_code>
        <a:user_id>mikel</a:user_id>
      </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:student_id_num>09341405</a:student_id_num>
        <a:survey_code>729</a:survey_code>
        <a:user_id>jessej</a:user_id>
      </a:APISignUp>
    </a:Result>
  </a:GetSignUpsForTimeslotResult>
</a:GetSignUpsForTimeslotResponse>
```

```

        </a:Result>
    </GetSignUpsForTimeslotResult>
</GetSignUpsForTimeslotResponse>

```

## GetStudyScheduleList

### 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 GetStudyScheduleList API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
location_id	Possible values are as follows: -2 = all locations -1 = all locations with no resource ids 0 = all locations with resource ids > 0 = specific resource id.  A resource_id exists if the location was chosen from the pulldown list of locations set in Location Listings.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
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/GetStudyScheduleList?username=adminusername&password=adminpassword&location\\_id=-2&start\\_date=2012-01-03&end\\_date=2012-01-10&lab\\_only=1](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyScheduleList?username=adminusername&password=adminpassword&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: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:APIStudySchedule>
    </a:Result>
  </GetStudyScheduleListResult>
</GetStudyScheduleListResponse>
```

## GetStudyList

### 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 GetResearcherIDByExperimentID API function.

## Input Parameters

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

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
active	Possible values are as follows: -1 = all 0 = inactive studies only 1 = active studies only
approved	Possible values are as follows: -1 = all 0 = un-approved studies only 1 = approved studies only
web_flag	Possible values are as follows: -1 = all 0 = lab studies only 1 = web studies only (internal online and external web)
survey_flag	Possible values are as follows: -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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
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/GetStudyList?username=adminusername&password=adminpassword&active=-1&approved=-1&web\\_flag=-1&survey\\_flag=-1](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyList?username=adminusername&password=adminpassword&active=-1&approved=-1&web_flag=-1&survey_flag=-1)

## Sample Output

```
<GetStudyListResponse>  
  <GetStudyListResult>
```

```

<a:Errors i:nil="true"/>
<a:Result>
  <a:APIStudyInfo>
    <a:active>1</a:active>
    <a:auto_credit_enable>1</a:auto_credit_enable>
    <a:cancel_limit>24</a:cancel_limit>
    <a:course_restrictions/>
    <a:creation_date>2013-05-01T08:00:21</a:creation_date>
    <a:credits>1</a:credits>
    <a:double_signup_flag>0</a:double_signup_flag>
    <a:duration>30</a:duration>
    <a:exp_signup_email>0</a:exp_signup_email>
    <a:experiment_id>144</a:experiment_id>
    <a:experiment_url>http://www.google.com<a:experiment_url/>
    <a:experimenter_timeslot>1</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>1234567890</a:irb_code>
    <a:irb_expiration_date>2014-12-
12T00:00:00</a:irb_expiration_date>
    <a:is_neg_req>0</a:is_neg_req>
    <a:last_activity>2013-05-03T08:04:43</a:last_activity>
    <a:last_timeslot>2013-05-05T09:30:00</a:last_timeslot>
    <a:long_desc>Test Study long description</a:long_desc>
    <a:multipart_exact>0</a:multipart_exact>
    <a:multipart_max_days>20</a:multipart_max_days>
    <a:multipart_min_days>1</a:multipart_min_days>
    <a:multipart_number>1</a:multipart_number>
    <a:total_parts>1</a:total_parts>
    <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/>
    <a:restrictions>None</a:restrictions>
    <a:short_desc>Test Study short description</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>Test Study</a:study_name>
    <a:survey_anon_flag>0</a:survey_anon_flag>
    <a:survey_flag>0</a:survey_flag>
    <a:total_credits>3</a:total_credits>
    <a:type>C</a:type>
    <a:usage_limit>1000</a:usage_limit>
    <a:valid>1</a:valid>
    <a:web_flag>0</a:web_flag>
  </a:APIStudyInfo>
</a:Result>
</GetStudyListResult>
</GetStudyListResponse>

```

## GetResearcherIDByExperimentID

### 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 GetResearcherIDByExperimentID API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
experiment_id	The experiment_id of the study. Pass in 0 for the prescreen.

### Output Parameters

Will return the person\_ids 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	invalid experiment_id
Error Code	-6	no researchers found for experiment_id or invalid experiment_id

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetResearcherIDByExperimentID?username=adminusername&password=adminpassword&experiment\\_id=123](https://myschool.sona-systems.com/services/SonaAPI.svc/GetResearcherIDByExperimentID?username=adminusername&password=adminpassword&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>
```

## GetPrescreenSurveyDataByPersonID

### 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 GetPrescreenSurveyDataByPersonID API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-1	no prescreen nor survey found for the specified experiment_id
Error Code	-2	no response data or participant found for the specified participant_id

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetPrescreenSurveyDataByPersonID?username=adminusername&password=adminpassword&experiment\\_id=0&person\\_id=9444](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPrescreenSurveyDataByPersonID?username=adminusername&password=adminpassword&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:responses>
    </a:Result>
  </a:Errors i:nil="true"/>
</a:Errors i:nil="true"/>
```

```

    <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>

```

## GetPrescreenSurveyDataByDateRange

### 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 GetPrescreenSurveyDataByDateRange API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-1	no prescreen nor survey found for the specified experiment_id
Error Code	-2	no response data or participant found for the specified participant_id
Error Code	-3	invalid start date
Error Code	-4	invalid end date
Error Code	-5	start date is greater than end date
Error Code	-6	no data found for selected date range
Error Code	-7	experiment_id is not found/invalid

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetPrescreenSurveyDataByDateRange?username=adminuser&password=adminpassword&experiment\\_id=0&start\\_date=2013-01-01&end\\_date=2013-12-31](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPrescreenSurveyDataByDateRange?username=adminuser&password=adminpassword&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: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>

```

```
</GetPrescreenSurveyDataByDateRangeResult>
</GetPrescreenSurveyDataByDateRangeResponse>
```

## GetStudyCreditAssignmentMismatch

### Description

This function returns a list of credits (signups) 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 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 GetStudyCreditAssignmentMismatch API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.

### Output Parameters

Will return APICreditItem 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	no course restriction mismatches found

### Sample Input

<https://myschool.sona-systems.com/services/SonaAPI.svc/>

[GetStudyCreditAssignmentMismatch?username=adminusername&password=adminpassword](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyCreditAssignmentMismatch?username=adminusername&password=adminpassword)

### Sample Output

```
<GetStudyCreditAssignmentMismatchResponse>
  <GetStudyCreditAssignmentMismatchResult>
    <a:Errors i:nil="true"/>
    <a:Result>
      <a:APICreditItem>
        <a:class_id>3</a:class_id>
        <a:comments>1.00 Credits archived record 22B</a:comments>
```

```

    <a:credit_id>453</a:credit_id>
    <a:credit_type>G</a:credit_type>
    <a:date_granted>2012-06-26T15:00:08</a:date_granted>
    <a:email>3p@yourschool.edu</a:email>
    <a:experiment_id>150</a:experiment_id>
    <a:first_name>Ivan</a:first_name>
    <a:grantor_id>0</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>9379</a:person_id>
    <a:show_credit>0</a:show_credit>
    <a:signup_id>380</a:signup_id>
    <a:student_id_num/>
    <a:user_id>smith</a:user_id>
  </a:APICreditItem>
  <a:APICreditItem>
    <a:class_id>3</a:class_id>
    <a:comments>1.00 Creditpoints archived record 22B</a:comments>
    <a:credit_id>460</a:credit_id>
    <a:credit_type>G</a:credit_type>
    <a:date_granted>2012-06-26T15:00:08</a:date_granted>
    <a:email>4p@yourschool.edu</a:email>
    <a:experiment_id>150</a:experiment_id>
    <a:first_name>Gina</a:first_name>
    <a:grantor_id>0</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>9380</a:person_id>
    <a:show_credit>0</a:show_credit>
    <a:signup_id>387</a:signup_id>
    <a:student_id_num/>
    <a:user_id>gjones</a:user_id>
  </a:APICreditItem>
</a:Result>
</GetStudyCreditAssignmentMismatchResult>
</GetStudyCreditAssignmentMismatchResponse>

```

## FixStudyCreditAssignmentMismatch

### 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 FixStudyCreditAssignmentMismatch API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.

### 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/  
FixStudyCreditAssignmentMismatch?username=adminusername&password=adminpassword](https://myschool.sona-systems.com/services/SonaAPI.svc/FixStudyCreditAssignmentMismatch?username=adminusername&password=adminpassword)

### Sample Output

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

## GetPersonInfoByPersonID

### 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 GetPersonInfoByPersonID API function.

Parameter	Description / Possible Values
-----------	-------------------------------

username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	invalid person_id

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByPersonID?username=adminusername&password=adminpassword&person\\_id=4321](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByPersonID?username=adminusername&password=adminpassword&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>4.5</a:credits_earned>
        <a:credits_needed>4</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <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: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:user_id>jake</a:user_id>
    </a:APIPerson>
</a:Result>
</GetPersonInfoByPersonIDResult>
</GetPersonInfoByPersonIDResponse>

```

## GetPersonInfoByUserID

### 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 GetPersonInfoByUserID API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	invalid user_id

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByUserID?username=adminusername&password=adminpassword&user\\_id=userid](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByUserID?username=adminusername&password=adminpassword&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>4.5</a:credits_earned>
        <a:credits_needed>4</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <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: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:user_id>jake</a:user_id>
      </a:APIPerson>
    </a:Result>
  </GetPersonInfoByUserIDResult>
</GetPersonInfoByUserIDResponse>
```

## GetCreditBySignupID

### 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 GetCreditBySignupID API function.

Parameter	Description / Possible Values
-----------	-------------------------------

username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	sign-up (signup_id) not found
Error Code	-6	credit record not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetCreditBySignupID?username=adminusername&password=adminpassword&signup\\_id=100](https://myschool.sona-systems.com/services/SonaAPI.svc/GetCreditBySignupID?username=adminusername&password=adminpassword&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</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>
```

## GetCreditsByPersonID

### 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 GetCreditsByPersonID API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	person (person_id) not found
Error Code	-6	credit record not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetCreditsByPersonID?username=adminusername&password=adminpassword&person\\_id=11498](https://myschool.sona-systems.com/services/SonaAPI.svc/GetCreditsByPersonID?username=adminusername&password=adminpassword&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: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>0.5</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: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>0.5</a:show_credit>
        <a:signup_id>1554</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>1555</a:credit_id>
        <a:credit_type>G</a:credit_type>
        <a:date_granted>2013-11-16T00:00:00</a:date_granted>
        <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>0.5</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>

```

## GetCreditsByTimeslotID

### Description

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

### Input Parameters

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

Parameter	Description / Possible Values
-----------	-------------------------------

username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
Error Code	-5	timeslot (timeslot_id) not found
Error Code	-6	no sign-ups for timeslot found

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByPersonID?username=adminusername&password=adminpassword&person\\_id=4321](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonInfoByPersonID?username=adminusername&password=adminpassword&person_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: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: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>0.5</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: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>0.5</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>

```

## GetTimeslotsByExperimentID

### Description

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

### Input Parameters

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

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
experiment_id	The experiment_id of the study
fill_status	fill_status values: <ul style="list-style-type: none"> <li>- A (all – all timeslots, regardless of if there are signups)</li> <li>- F (full only – every space in the timeslot is taken)</li> <li>- E (empty only – no spaces in the timeslot are taken)</li> <li>- S (with signups – at least one space in the timeslot is taken)</li> </ul>

## 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetTimeslotsByExperimentID?username=adminusername&password=adminpassword&experiment\\_id=123&fill\\_status=A](https://myschool.sona-systems.com/services/SonaAPI.svc/GetTimeslotsByExperimentID?username=adminusername&password=adminpassword&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: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: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: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:APIStudySchedule>
</a:Result>
</GetTimeslotsByExperimentIDResult>
</GetTimeslotsByExperimentIDResponse>

```

## GetTimeslotsByExperimentIDDateRange

### 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 GetTimeslotsByExperimentIDDateRange API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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: <ul style="list-style-type: none"> <li>- A (all – all timeslots, regardless of if there are signups)</li> <li>- F (full only – every space in the timeslot is taken)</li> <li>- E (empty only – no spaces in the timeslot are taken)</li> <li>- S (with signups – at least one space in the timeslot is taken)</li> </ul>

### 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Authentication failed	-4	user not found
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/>

[GetTimeslotsByExperimentIDDateRange?username=adminusername&password=adminpassword&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: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:APIStudySchedule>
    </a:Result>
  </GetTimeslotsByExperimentIDDateRangeResult>
</GetTimeslotsByExperimentIDDateRangeResponse>
```

## GetParticipantStats

### 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 GetParticipantStats API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
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/GetParticipantStats?username=adminusername&password=adminpassword&class\\_id=4](https://myschool.sona-systems.com/services/SonaAPI.svc/GetParticipantStats?username=adminusername&password=adminpassword&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>
```

## GetStudyStats

### 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 GetStudyStats API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid experiment_id	[entered value]	not a valid experiment_id

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyStats?username=adminusername&password=adminpassword&experiment\\_id=14](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyStats?username=adminusername&password=adminpassword&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</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>
```

## GetStudyStatsByDateRange

### 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 GetStudyStatsByDateRange API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.

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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
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/GetStudyStatsByDateRange?username=adminusername&password=adminpassword&experiment\\_id=4&start\\_date=2016-01-01&end\\_date=2016-12-31](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyStatsByDateRange?username=adminusername&password=adminpassword&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</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>
```

## GetInstructorByClassID

### Description

This API retrieves the user information for an Instructor assigned to a particular class.

### Input Parameters

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

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid class_id	[entered value]	not a valid class_id
Invalid person_id	[entered value]	not a valid person_id (no instructor assigned to the course)

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetInstructorByClassID?username=adminusername&password=adminpassword&class\\_id=2](https://myschool.sona-systems.com/services/SonaAPI.svc/GetInstructorByClassID?username=adminusername&password=adminpassword&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</a:credits_needed>
        <a:daily_reminder>0</a:daily_reminder>
        <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>

```

## GetPersonListByLoginDate

### 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 GetPersonListByLoginDate API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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

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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid person_type	-4	blank or non-numeric person_type
Invalid login_allowed	-5	blank or non-numeric login_allowed
Invalid login_date	-6	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/GetPersonListByLoginDate?username=adminusername&password=adminpassword&person\\_type=X&login\\_allowed=1&login\\_date=2017-08-30](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonListByLoginDate?username=adminusername&password=adminpassword&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>4.5</a:credits_earned>
        <a:credits_needed>4</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <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: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: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>4.5</a:credits_earned>
        <a:credits_needed>4</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <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: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:user_id>miles</a:user_id>
    </a:APIPerson>
</a:Result>
</GetPersonListByLoginDateResult>
</GetPersonListByLoginDateResponse>

```

## GetPersonListByCreationDate

### 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 account creation date.

### Input Parameters

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

Parameter	Description / Possible Values
username	Use Sona admin User ID.

password	Use Sona admin password.
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	User's account creation date. 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid person_type	-4	blank or non-numeric person_type
Invalid login_allowed	-5	blank or non-numeric login_allowed
Invalid creation_date	-6	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/GetPersonListByCreationDate?username=adminusername&password=adminpassword&person\\_type=S&login\\_allowed=1&creation\\_date=2017-01-01](https://myschool.sona-systems.com/services/SonaAPI.svc/GetPersonListByCreationDate?username=adminusername&password=adminpassword&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>4.5</a:credits_earned>
        <a:credits_needed>4</a:credits_needed>
        <a:daily_reminder>1</a:daily_reminder>
        <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: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: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>4.5</a:credits_earned>
    <a:credits_needed>4</a:credits_needed>
    <a:daily_reminder>1</a:daily_reminder>
    <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: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:user_id>henry</a:user_id>
  </a:APIPerson>
</a:Result>
</GetPersonListByCreationDateResult>
</GetPersonListByCreationDateResponse>

```

## GetStudyInfoByExperimentID

### 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 GetStudyInfoByExperimentID API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
experiment_id	The experiment_id of the study. In the case of a multi-part study, each part has a different experiment_id, 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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid experiment_id	-4	experiment_id entered is invalid
Experiment_id not found	-5	experiment_id entered is not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyInfoByExperimentID?username=adminusername&password=adminpassword&experiment\\_id=144](https://myschool.sona-systems.com/services/SonaAPI.svc/GetStudyInfoByExperimentID?username=adminusername&password=adminpassword&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</a:age_maximum>
        <a:age_minimum>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</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</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>

```

## ActivateUser

### 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 ActivateUser API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.

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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid person_id	-4	person_id entered is invalid
Person_id not found	-5	person_id entered is not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/ActivateUser?username=adminusername&password=adminpassword&person\\_id=15](https://myschool.sona-systems.com/services/SonaAPI.svc/ActivateUser?username=adminusername&password=adminpassword&person_id=15)

### Sample Output

```
<ActivateUserResponse>
  <ActivateUserResult>
    <a:Errors i:nil="true"/>
      <a:Result>1</a:Result>
    </ActivateUserResult>
  </ActivateUserResponse>
```

## DeactivateUser

### 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 DeactivateUser API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid person_id	-4	person_id entered is invalid
Person_id not found	-5	person_id entered is not found

### Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/DeactivateUser?username=adminusername&password=adminpassword&person\\_id=15](https://myschool.sona-systems.com/services/SonaAPI.svc/DeactivateUser?username=adminusername&password=adminpassword&person_id=15)

### Sample Output

```
<DeactivateUserResponse>
  <DeactivateUserResult>
    <a:Errors i:nil="true"/>
    <a:Result>1</a:Result>
  </DeactivateUserResult>
</DeactivateUserResponse>
```

## SetUserAdminComments

### 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 SetUserAdminComments API function.

Parameter	Description / Possible Values
username	Use Sona admin User ID.
password	Use Sona admin password.
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

Upon success, will return an integer value of:

- 1 indicating that the comment has been added or the previous comment overwritten.
- 2 indicating that the comment has been added or the previous comment overwritten, but some characters were stripped off due to length limits.
- 3 indicating that the comment has been appended.
- 4 indicating that the comment has been appended, but some characters were stripped off due to length limits.
- 5 indicating that the comment has been deleted.

## Error Codes

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

Error Type	Error Code	Description
Authentication failed	0	unspecified error
Authentication failed	-1	bad username/password
Authentication failed	-2	bad user type (not an admin)
Authentication failed	-3	login not allowed (account disabled)
Invalid person_id	-4	person_id entered is invalid
Person_id not found	-5	person_id entered is not found
Invalid action	-6	action entered is invalid

## Sample Input

[https://myschool.sona-systems.com/services/SonaAPI.svc/SetUserAdminComments?username=adminusername&password=adminpassword&person\\_id=15&action=O&comments=place comment here](https://myschool.sona-systems.com/services/SonaAPI.svc/SetUserAdminComments?username=adminusername&password=adminpassword&person_id=15&action=O&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

### 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
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   N = 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

### 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	<p>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:</p> <ul style="list-style-type: none"> <li>• Study has multiple researchers.</li> <li>• Study is configured to allow researchers at timeslot-level.</li> <li>• A particular researcher is assigned to this timeslot.</li> </ul>
---------------	-------	--

### APIStudyInfo

This contains the 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	
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	Int32	Minimum age to sign up for the study. 0 = no age limits.
age_maximum	Int32	Maximum age to sign up for the study. Ignore if age_minimum=0.

### APIPrescreenSurveyData

This class contains the 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 user acknowledged HIPAA 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
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

### 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)