# gocanvas

GoCanvas – Web Services Overview
Version 2.0.19

# Contents

# GoCanvas Web Services API Overview

The GoCanvas web services API provides programmatic access to data hosted within GoCanvas using a secure, simple and powerful application programming interface (API).  The GoCanvas web services API is centered around the REST style web services interface.

The goal of the GoCanvas web service API is to allow GoCanvas customers to automate the integration of data submitted to the GoCanvas servers back into other systems.

# Authentication

There are two methods to authenticate to the GoCanvas API.  By default, each API call accepts the username and password of an authorized user as required parameters.  The second option is to set up an API Key at the account or department level and pass a token inside the header of your request with a username as a parameter.

## Configuring API Token

**First Step:** Enable API Key Feature & Create API Key

1.  Go to the Account > Account Settings page.
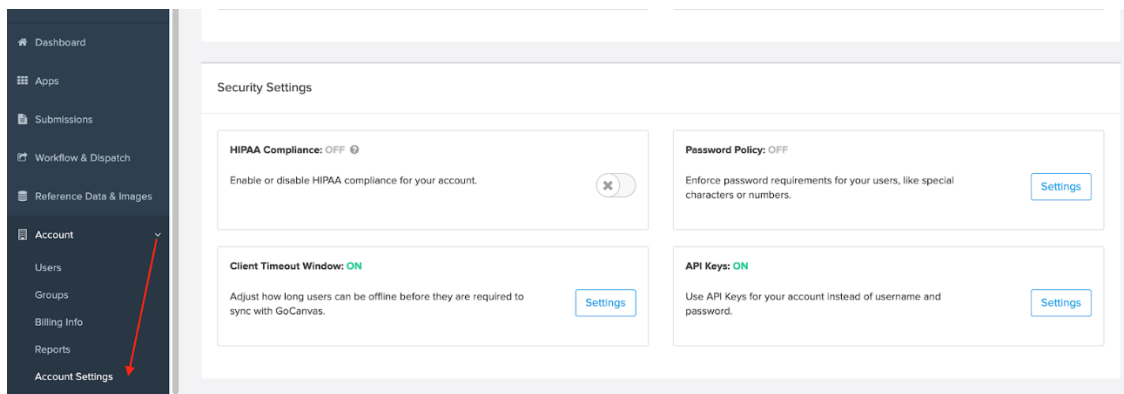


2.  Go to the Security Settings section at the bottom, and you will see "API Keys".
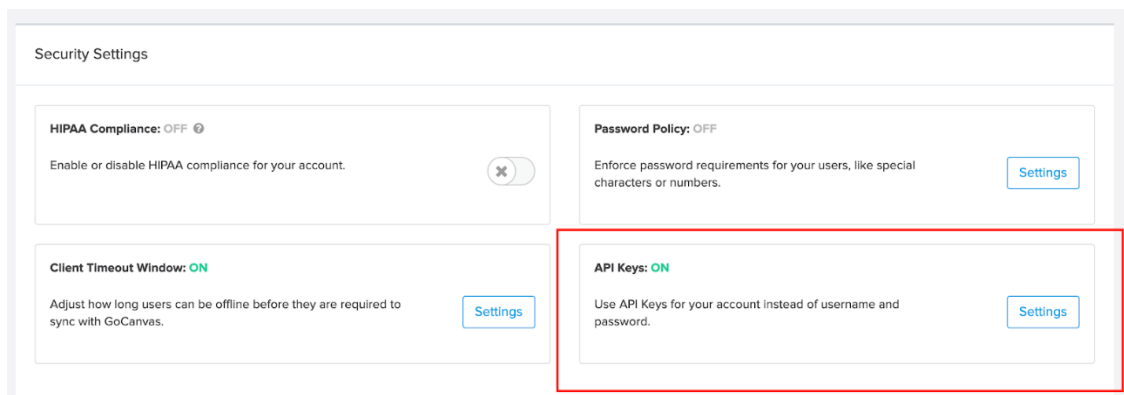
3. Click on the Settings button, you will be redirected to the page where you can create new API keys, and where your keys will be displayed.
4. First, click "Enable API Keys" to enable this feature.
5. Then, click "Create API Key" to create a new API key.

| Name | API Key | Action |
|------|---------|--------|
| test-key | NapDjG******************************************************* | 🗑 |

☑ Enable API Keys                                                     Create API Key

Cancel    Save
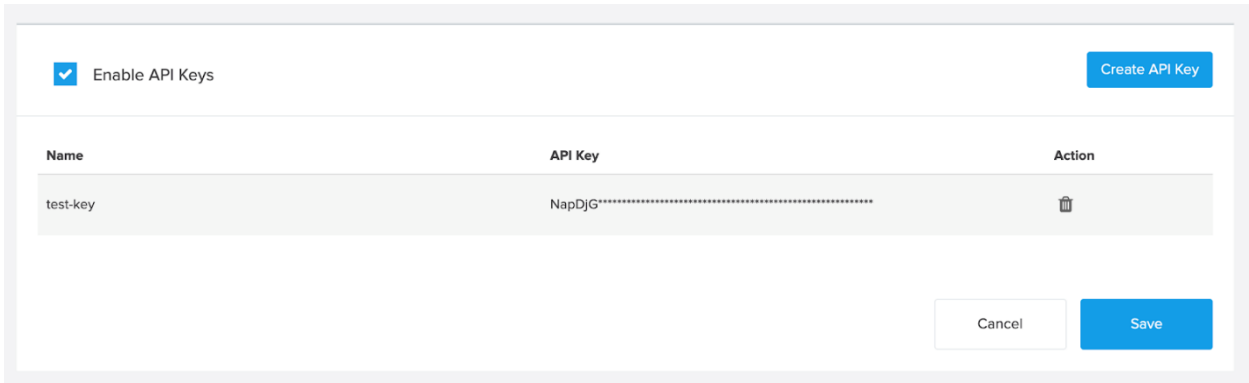
6. By clicking the "Create API Key" button, you will be redirected to the page to create a key.
7. Once you enter a key name and click the "Save" button, you will see the key and you must copy it in order to use it in your API calls.

## API Keys
Account Settings / API Keys

Use API Keys for your account instead of username and password. Learn More

**Key Name**

Cancel    Save

API Keys

Account Settings / API Keys

Use API Keys for your account instead of username and password. Learn More

API Key Created

For security reasons, we will not show this to you again, so please copy this key and save it somewhere safe!

pVoWD9F7mWuyqJ2RHxLUkTB49NMnleFqUnO7qMEw=
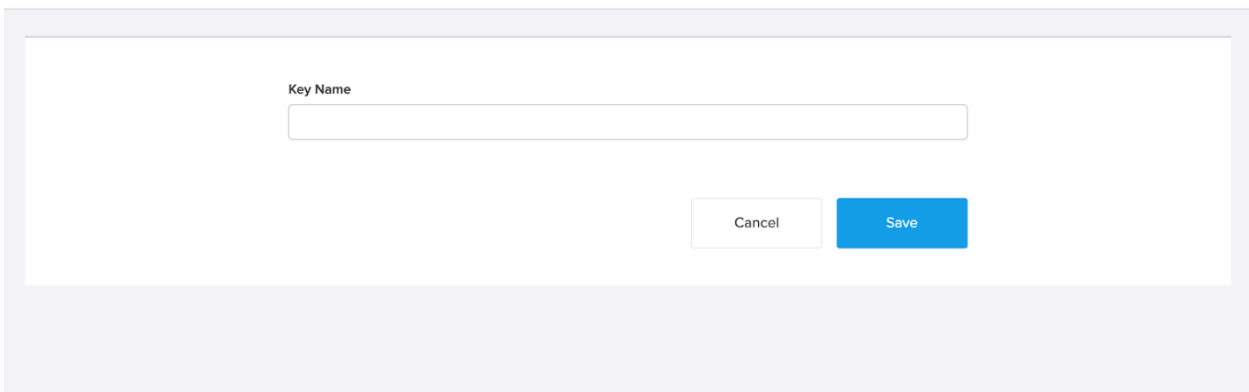
Done
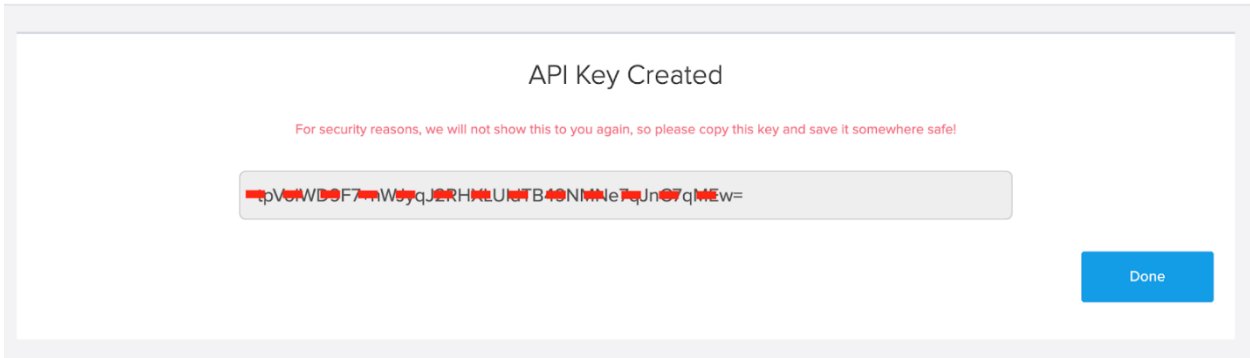
8. If Department Functionality is enabled for your company, you will have the option to choose the Department the key has access to.   If you choose "All" the token will have access to All Department data.

API Keys

Account Settings / API Keys

Use API Keys for your account instead of username and password. Learn More

Key Name

Select Department

All
✓ First Department

Cancel          Save
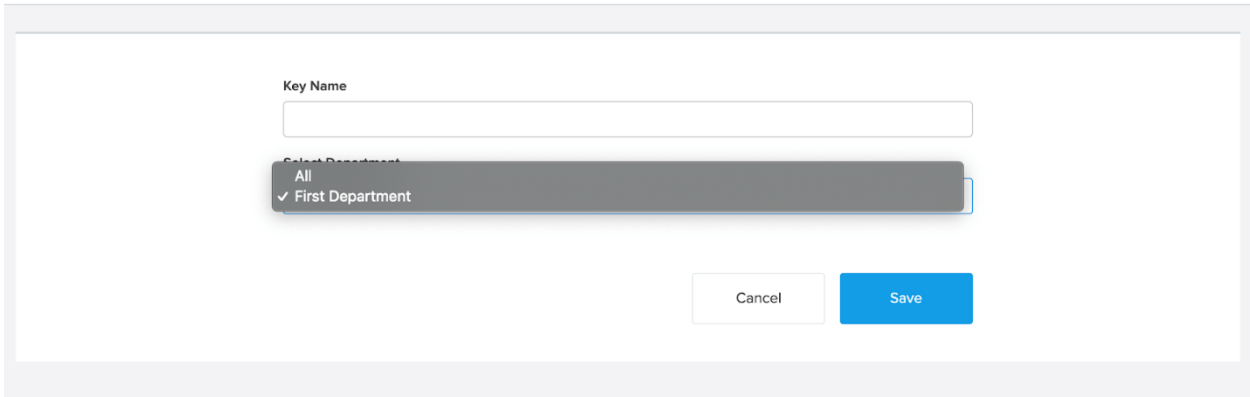
**Second Step:** Once you have enabled the API Keys feature and have created an API Key, you can use API calls by passing the token inside the header of your request.  The username parameter is mandatory, otherwise "Permission Denied for the requested resource" error will be returned by API calls.  When the API Keys feature is enabled, the API will no longer be able to accessible with username and password.

- Curl Request Format:
  - curl -H "Authorization: Bearer NapDjG**********************************"
    localhost:3000/apiv2/forms.xml -F 'username=test_user@gocanvas.com'

- HTTP Request Format:
  - Request Headers:
    Authorization:"Bearer NapDjG**********************************"

  - Request Body:
    username:" test_user@gocanvas.com"

# Forms API

## Description

The Forms API provides the capability to retrieve all or a subset of forms associated with the specified login.

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

Forms API URL: https://www.gocanvas.com/apiv2/forms.xml

## Parameters

The Forms API is restricted to authenticated users and requires a username and password to access. All other parameters are optional. When no parameters are specified, a list of all forms applicable for that user is returned. If Departments is enabled, and if a department isn't specified, it will return the forms from the default department, otherwise you can pass in the name of the department to pull the forms for that department.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| name | String (case insensitive) | No | Return all forms with the given name. Two forms can share the same name, so this may return more than one form. |
| search_name | String (case insensitive) | No | Return all forms whose name contains the search_name. |
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as. You must have permission to the other users account via the Associate functionality. |
| department | String (case insensitive) | No | Name of the department. Returns all forms that belong to the department. If left blank, the forms for the user's default department will be returned. If set to "all" forms for all departments will be returned. |

## Results

The result of the forms web service invocation is a collection of forms.

Form data and description:

| Name | Data Type | Description |
|------|-----------|-------------|
| Name | String | The name of the form |
| Id | Integer | Unique identifier of this form. This identifier is not guaranteed to be consistent between releases of GoCanvas, therefore this identifier should not be stored and used over long periods of time. This identifier can be used to interrogate submissions in a way that is guaranteed to be unique. |
| OriginatingLibraryTemplateId | Integer | Identifier of template from public library from which the form was derived. The primary purpose of this indicator is to guarantee the format of a form by knowing it the format of the originating template and the fact that the derived form can't be modified to break that format. When the form does not have an originating template, the field is empty. |
| GUID | String | Identifier that is unique to this lineage of forms. Different versions of the same logical form will have the same GUID. This can be used to interrogate submissions associated with a logical form that is persistent forever and guaranteed to be unique to the logical form. |
| Status | String | The possible values for this field are 'new', 'published', 'pending' and 'retired' |
| Version | Integer | The version of this form |

## Examples

REST API call:

https://www.gocanvas.com/apiv2/forms.xml?username=test@test.com&password=test_password

Resulting XML:

<?xml version="1.0" encoding="utf-8"?>

```
<CanvasResult>
  <Forms>
    <Form Id="1" OriginatingLibraryTemplateId="" GUID="f6167220-a09d-11df-8e32-
00242bb0cd70">
      <Name>test form</Name>
      <Status>published</Status>
      <Version>1</Version>
    </Form>
    <Form Id="2">
      <Name>second form</Name>
      <Status>new</Status>
      <Version>1</Version>
    </Form>
  </Forms>
</CanvasResult>
```

## Submissions API

### Description

The Submissions API provides the capability to return all or a subset of submissions associated with the specified login.

### URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

Submissions API URL: https://www.gocanvas.com/apiv2/submissions.xml

If you needed to request an individual submission, you can use either the id, response id or the submission_guid attributes. The format of the response will match exactly the one returned by the server when accessing multiple submissions. Please note that when you specify a id, response id or submission_guid to match a single submission, all other request parameters are ignored.

### Parameters

The Submissions API is restricted to authenticated users and requires a username and password to access.  At least one parameter must be specified to limit the result to either a specific form or subset of forms via the form_name, search_form_name or form_id parameters.  If a 'begin_date' is specified, a 'end_date' must also be specified.  If a 'begin_second' is specified, a 'end_second' must also be specified.  Results will be limited to 100 submissions per request.  The 'page' parameter specifies which group of results to return.  It will default to 1 if it is not specified.  The total number of pages is returned in the result.  If Departments is enabled, and if a department isn't specified, it will return the

submissions from the default department, otherwise you can pass in the name of the department to pull the submissions in the department. You can also include workflows with your submissions as well with the 'include_workflow' parameter.

| Name | Data Type | Required | Description |
|---|---|---|---|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| id | Integer | No | Returns a specific submission based on the ID |
| submission_guid | String (case insensitive) | No | Returns a specific submission based on the submission_guid |
| response id | String (case insensitive) | No | Returns a specific submission base on response ID |
| form_name | String (case insensitive) | No | Return all submissions associated with a form with the given name.  Two forms can share the same name, so this may return submissions associated with different forms. |
| search_form_name | String (case insensitive) | No | Return all submissions associated with the forms whose name contain the search_form_name. |
| form_id | Integer | No | Returns all submissions associated with the form which has the specified form_id as a unique identifier |
| form_guid | String (case sensitive) | No | Returns all submissions associated with the forms that have this guid.  All versions of a logical form have the same guid. |
| begin_date | Date (MM/DD/YYYY) | No | Return all submissions submitted on or after this date |
| end_date | Date (MM/DD/YYYY) | No | Return all submissions submitted on or before this date |
| begin_second | Integer | No | Return all submissions submitted at or after the specified time expressed as the seconds since Unix epoch.  This is intended to allow more fine grained control than begin_date. |

| end_second | Integer | No | Return all submissions submitted at or before the specified time expressed as the seconds since Unix epoch. This is intended to allow more fine grained control than end_date |
|---|---|---|---|
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as.  You must have permission to the other users account via the Associate functionality. |
| page | Integer | No | Specifies the page number of the returned submissions. If not specified, it is defaulted to 1. |
| department | String(Case insensitive) | No | Return all submissions by department name |
| Include_workflow | String(Case Insensitive) | No | Available options are: Submission, SubmissionAndWorkflow, and Workflow. If the value is Submission, it will return just the submissions without workflows. If the value is SubmissionAndWorkflow, it will return submissions that have workflows and submissions that do not have them. If the value is Workflow, it will return only the submissions that have workflows. |

## Results

The results of the submissions web service invocation is a collection of submissions.  Each submission contains a simplified form structure and can contain a collection of sections. Each section can contain a collection of sheets.  Each sheet can contain a collection of responses.  In addition, if a sheet is a list (either advanced or simple) it can contain a collection of responsegroups.  A responsegroup simply contains the data entered in the list and a section containing the addition sheets and responses.

Submission data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| Date | Date (YYY.MM.DD HH:MM:SS) | The date and time stamp of when the submission was uploaded to the GoCanvas Cloud. |
| DeviceDate | Date (YYY.MM.DD HH:MM:SS) | The date and time stamp of when the submission was completed on the device. |
| UserName | String | The username of the submitter |

| FirstName | String | The first name of the submitter |
|---|---|---|
| LastName | String | The last name of the submitter |
| ResponseID | String | The unique identifier of the submission – immutable |
| Id | Integer | Unique identifier of this submission. This identifier is not guaranteed to be consistent between releases of GoCanvas, therefore this identifier should not be stored and used over long periods of time. |
| Department | String | The department where the submission resides in. |

Form data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| Name | String | The name of the form |
| Id | Integer | Unique identifier of this form. This identifier is not guaranteed to be consistent between releases of GoCanvas, therefore this identifier should not be stored and used over long periods of time. |
| Status | String | The possible values for this field are 'new', 'published', 'pending' and 'retired' |
| Version | Integer | The version of this form |

Section data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| name | String | The name of the section |

Sheet data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| Name | String | The name of the sheet |

Response data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| Label | String | The label of the response |
| Value | String | The value of the response |
| Type | String | The type of response. Values are: "Text |

| | | Box","Integer","Decimal","Date","Time","Value List","File Selection","Checkbox","Multi-Line Text","Static Text","Image Capture","Signature","Summary","Calculation" |
|---|---|---|

## Examples

REST API call to pull all submissions for a specific account:

https://www.gocanvas.com/apiv2/submissions.xml?username=test@test.com&password=test_password&form_name=test

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
<TotalPages>1</TotalPages>
<Submissions>
  <Submission Id="1" Department="Test Department">
   <UserName>chris.morrisette@gocanvas.com</UserName>
   <FirstName>Chris</FirstName>
   <LastName>Morrisette</LastName>
   <ResponseID>258aaa53-7762-4e58-bb88-f40490c63556</ResponseID>
   <Date>2009.08.31 20:20:17</Date>
   <DeviceDate>2009.08.31 20:20:17</DeviceDate>
   <Form Id="1">
    <Name>test form</Name>
    <Status>published</Status>
    <Version>1</Version>
   </Form>
   <Sections>
    <Section>
     <Name>Main Info</Name>
     <Sheets>
     <Sheet>
      <Name>Main Info</Name>
      <Responses>
        <Response id="8217">
         <label>First Name</label>
         <value>James</value>
         <type>Text Box</type>
        </Response>
      </Responses>
```

```
      </Sheet>
      <Sheet>
        <Name>Date List</Name>
        <ResponseGroups>
          <ResponseGroup>
            <Response>
              <Label>Date</Label>
              <Value>6/22/09</Value>
              <Type>Date</Type>
            </Response>
            <Section>
              <Name>Date Entry</Name>
              <Sheets>
                <Sheet>
                  <Name>Date</Name>
                  <Responses>
                    <Response>
                      <Label>Date</Label>
                      <Value>06/22/09</Value>
                      <Type>Date</Type>
                    </Response>
                    <Response>
                      <Label>Click your track ball and select
                      "Next" to enter a list of time
                      entries.</Label>
                      <Value></Value>
                      <Type>Static Text</Type>
                    </Response>
                  </Responses>
                </Sheet>
              </Sheets>
            </Section>
          </ResponseGroup>
        </ResponseGroups>
      </Sheet>
    </Sheets>
   </Section>
  </Sections>
 </Submission>
</Submissions>
</CanvasResult>
```

REST API call to pull all submissions for a specific account for the a 24 hours period (note: dates are inclusive and GMT based):

https://www.gocanvas.com/apiv2/submissions.xml?username=test@test.com&password=test_password&form_name=test&begin_date=05/25/2010&end_date=05/25/2010

REST API call to pull all submissions for a specific account for the a 48 hours period (note: dates are inclusive and GMT based):

https://www.gocanvas.com/apiv2/submissions.xml?username=test@test.com&password=test_password&form_name=test&begin_date=05/24/2010&end_date=05/25/2010

REST API call to pull submissions with workflows:

https://www.gocanvas.com/apiv2/submissions.xml?username=test@test.com&password=test_password&form_name=test&include_workflow=SubmissionAndWorkflow

Resulting XML:

```xml
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
<TotalPages>1</TotalPages>
<Submissions>
  <Submission Id="1" Department="Test Department">
   <UserName>chris.morrisette@gocanvas.com</UserName>
   <FirstName>Chris</FirstName>
   <LastName>Morrisette</LastName>
   <ResponseID>258aaa53-7762-4e58-bb88-f40490c63556</ResponseID>
   <Date>2009.08.31 20:20:17</Date>
   <DeviceDate>2009.08.31 20:20:17</DeviceDate>
   <Form Id="1">
    <Name>test form</Name>
    <Status>published</Status>
    <Version>1</Version>
   </Form>
   <Sections>
    <Section>
     <Name>Main Info</Name>
     <Sheets>
     <Sheet>
      <Name>Main Info</Name>
      <Responses>
        <Response id="8217">
         <label>First Name</label>
         <value>James</value>
         <type>Text Box</type>
```

```
        </Response>
      </Responses>
    </Sheet>
    <Sheet>
      <Name>Date List</Name>
      <ResponseGroups>
        <ResponseGroup>
          <Response>
            <Label>Date</Label>
            <Value>6/22/09</Value>
            <Type>Date</Type>
          </Response>
          <Section>
            <Name>Date Entry</Name>
            <Sheets>
              <Sheet>
                <Name>Date</Name>
                <Responses>
                  <Response>
                    <Label>Date</Label>
                    <Value>06/22/09</Value>
                    <Type>Date</Type>
                  </Response>
                  <Response>
                    <Label>Click your track ball and select
                    "Next" to enter a list of time
                    entries.</Label>
                    <Value></Value>
                    <Type>Static Text</Type>
                  </Response>
                </Responses>
              </Sheet>
            </Sheets>
          </Section>
        </ResponseGroup>
      </ResponseGroups>
    </Sheet>
  </Sheets>
</Section>
</Sections>
<Workflow Id="ba8db2a506892eb-1438708947021">
    <User>jefferson, thomas</User>
    <UserEmail>tjefferson@test.com</UserEmail>
    <AppName>test form(version 4) (Workflow version 1)</AppName>
    <Status>Completed</Status>
    <RejectionNote/>
```

```
        <CreationDate>2015.08.04 17:22:41</CreationDate>
        <CompletedDate>2015.08.04 17:23:23</CompletedDate>
      </Workflow>
    </Submission>
  </Submissions>
</CanvasResult>
```

## Delete Submissions API

## Description

This API allows you to delete submissions by sending an XML payload with the submission IDs.

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.  This URL and the necessary XML can only be accessed via a POST to the appropriate URL.

Delete Submissions API URL: https://www.gocanvas.com/apiv2/delete_submissions

## Parameters

The Delete Submissions API is restricted to authenticated users and requires a username and password to access.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |

## Payload

The payload to send must be a well formed XML document consisting of a "Submissions" element, wrapping up one or more "Submission" element. Each of those elements will have an "Id" attribute which will be the id of the Submission.

List data and descriptions:

| Name | Data Type | Description |
|------|-----------|-------------|

| Id | String | The Id of the submission you want to delete |
|----|--------|---------------------------------------------|

Post Payload:

```
<?xml version="1.0" encoding="utf-8"?>
<Submissions>
        <Submission Id="7346" />
        <Submission Id="2999" />
        <Submission Id="1989" />
</Submissions>
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
</CanvasResult>
```

# Images API

## Description

The extra processing and difficulty in encoding a binary image into the submission XML structure lends itself to being externalized into its own web service call to retrieve image data associated with a submission.  The unique identifier of the image can be found within the submission payload as specified within the submission results section.

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

Images API URL: https://www.gocanvas.com/apiv2/images.xml

## Parameters

The Submissions API is restricted to authenticated users and requires a username and password to access.  In addition, the id field is required.  If these three fields aren't specified, an error is returned.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |

| image_id | Integer | Yes | The unique id of the image to return. |
|---|---|---|---|
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as. You must have permission to the other users account via the Associate functionality. |

## Results

The result of this web service invocation will be either a standard JPEG image or an error code (detailed in the 'Error Codes' section).

## Examples

https://www.gocanvas.com/apiv2/images.xml?image_id=1&username=test@gocanvas.com&password=test

# Reference Data API

## Description

Reference Data is the concept of storing datasets within GoCanvas that can be referenced by forms.  The datasets are similar to a database table, in that there are columns and rows, with columns being uniquely named and used as the identifying attribute within the reference data. In addition, each reference data dataset needs to be uniquely named.  Datasets can be divided into subsets corresponding to user groups.

There is currently no limit on the number of datasets that can be uploaded.  A dataset uploaded with the same name as an existing dataset will overwrite and replace the existing dataset unless the "Action" attribute is set to "Append".  The columns of appended or updated datasets must match the columns of the existing records.

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.  This URL and the necessary XML can only be accessed via a POST to the appropriate URL.

Reference Data API URL: https://www.gocanvas.com/apiv2/reference_datas

## Parameters

The Reference Data API is restricted to authenticated users and requires a username and password to access. If these two fields aren't specified, an error is returned.

| Name | Data Type | Required | Description |
|---|---|---|---|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as. You must have permission to the other users account via the Associate functionality. |

## Payload

The payload to send must be a well formed XML document consisting of a "List" element, a "Columns" element containing the columns and a "Rows" element containing the rows of the dataset. If Departments is enabled, and if a department attribute isn't specified, it will upload the reference data to the default department, otherwise you can pass in the department attribute in the List node to upload the reference data in the department.

List data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| Name | String | The attribute name of the dataset to operate on. |
| Action | String | Optional. Values "Update" (default) or "Append". Update will replace the data set. Append will add the data set. Append can only be used if the data has been divided into User Groups. Otherwise it is ignored and an Update will be performed. |
| UserGroup | String | Optional. Use when Action="Update". If included, it specifies the User Group that the data will replace. |
| UserGroup Column | String | Optional. Use when Action="Update". If included, the data set will be divided into User Groups by the column specified. |
| Department | String | Name of the department that you want the reference data uploaded into |

Columns data and descriptions:

| Name | Data Type | Description |
|---|---|---|

| c | String | The name of the column within the dataset |
|---|--------|-------------------------------------------|

"r" data and descriptions:

| **Name** | **Data Type** | **Description** |
|----------|---------------|-----------------|
| v | String | The value for that column in that row, the number of values within the row should be equivalent to the number of columns specified |

## Results

If the request is processed correctly, then an empty CanvasResult element will be returned, with a response code of 200 from the web server.  If an error occurs while processing the request, one of the standard errors detailed in the Error Codes section will be returned.

## Examples

REST API call:

```
https://www.gocanvas.com/apiv2/reference_datas?username=test@test.com&password=test_password
```

Post Payload:

```
<?xml version="1.0" encoding="utf-8"?>
<List Name="test name" Department="Test Department">
  <Columns><c>Column1</c><c>Column2</c><c>Column3</c></Columns>
  <Rows>
    <r><v>Data1Column1</v><v>Data1Column2</v><v>Data1Column3</v></r>
    <r><v>Data2Column1</v><v>Data2Column2</v><v>Data2Column3</v></r>
  </Rows>
</List>
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
</CanvasResult>
```

Post Payload to divide data by into user groups defined in Column1:

```
<?xml version="1.0" encoding="utf-8"?>
<List Name="test name" UserGroupColumn ="Column1>
```

```
<Columns><c>Column1</c><c>Column2</c><c>Column3</c></Columns>
<Rows>
 <r><v>Group1</v><v>Data1Column2</v><v>Data1Column3</v></r>
 <r><v>Group2</v><v>Data2Column2</v><v>Data2Column3</v></r>
</Rows>
</List>
```

Post Payload to replace the data in Group2:

```
<?xml version="1.0" encoding="utf-8"?>
<List Name="test name" Action="Update" UserGroup =" Group2>
 <Columns><c>Column1</c><c>Column2</c><c>Column3</c></Columns>
 <Rows>
  <r><v>Group2</v><v>Data3Column2</v><v>Data3Column3</v></r>
  <r><v>Group2</v><v>Data4Column2</v><v>Data4Column3</v></r>
 </Rows>
</List>
```

Post Payload to append new data to existing or new user groups:

```
<?xml version="1.0" encoding="utf-8"?>
<List Name="test name" Action="Append">
 <Columns><c>Column1</c><c>Column2</c><c>Column3</c></Columns>
 <Rows>
  <r><v>Group2</v><v>Data5Column2</v><v>Data5Column3</v></r>
  <r><v>Group3</v><v>Data6Column2</v><v>Data6Column3</v></r>
 </Rows>
</List>
```

# CSV Meta Data API

## Description

The CSV Meta Data API is used to download information regarding what form versions exist that have submissions.  When downloading a CSV using the GoCanvas API, the exact form version must be specified, because it dictates the format of the CSV that is exported – therefore the CSV Meta Data API will usually be invoked first to determine the exact form version to use to download the CSV.

The procedural way that most users will download CSV data is:

- Invoke the Forms API to determine which current Form ID (and its lineage) to work with.  By determining the Form ID, it will also allow the lineage of Form Ids that came before it to be determined.

- Invoke the GoCanvas Meta Data API, passing in the currently  Form ID (or GUID) to

> determine which specific Form ID from the entire lineage to work with based upon the submission count for a specific date range

– For each specific Form ID within the date range, use the CSV API to download the CSV file

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

CSV Meta Data API URL: https://www.gocanvas.com/apiv2/csv_meta_datas.xml

## Parameters

The CSV Meta Data API requires valid GoCanvas username and password to access content, in addition to a way to specify a specific lineage of forms (either a form ID, form GUID or form name) and a date range.  If Departments is enabled, and if a department isn't specified, it will download the information from the default department, otherwise you can pass in the name of the department to download the information in the department.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| form_name | String (case insensitive) | No | Query submission count and form information based upon form name.  Two forms can share the same name, so this may return submissions associated with different forms. |
| search_form_name | String (case insensitive) | No | Query submission count and form information associated with the forms whose name contain the search_form_name. |
| form_id | Integer | No | Query submission count and form information associated with the form which has the specified form_id as a unique identifier |
| form_guid | String (case sensitive) | No | Query submission count and form information associated with the forms that have this guid.  All versions of a logical form have the same guid. |
| begin_date | Date (MM/DD/Y | No | Query submission count and form information submitted on or after this date |

| | | | |
|---|---|---|---|
| | YYY) | | |
| end_date | Date (MM/DD/YYYY) | No | Query submission count and form information submitted on or before this date |
| begin_second | Integer | No | Query submission count and form information submitted at or after the specified time expressed as the seconds since Unix epoch.  This is intended to allow more fine grained control than begin_date. |
| end_second | Integer | No | Query submission count and form information submitted at or before the specified time expressed as the seconds since Unix epoch. This is intended to allow more fine grained control than end_date |
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as.  You must have permission to the other users account via the Associate functionality. |
| department | String (case insensitive) | No | Name of the department that you want the information downloaded from |

## Results

The results of the CSV meta data API is a collection of forms from the same lineage, with the number of submissions associated with each form.

Form data and description:

| Name | Data Type | Description |
|---|---|---|
| Name | String | The name of the form |
| Id | Integer | Unique identifier of this form.  This identifier is not guaranteed to be consistent between releases of GoCanvas, therefore this identifier should not be stored and used over long periods of time.  This identifier can be used to interrogate submissions in a way that is guaranteed to be unique. |
| OriginatingLibraryTemplateId | Integer | Identifier of template from public library from which the form was derived.  The primary purpose of this indicator is to guarantee the format of a form by knowing it the format of the originating template and the fact that the derived form can't be modified to break that format.  When the form does not have an originating |

| | | template, the field is empty. |
|---|---|---|
| GUID | String | Identifier that is unique to this lineage of forms. Different versions of the same logical form will have the same GUID. This can be used to interrogate submissions associated with a logical form that is persistent forever and guaranteed to be unique to the logical form. |
| Name | String | The name of the form for this specific version. This name can easily be different for different versions. |
| Status | String | The possible values for this field are 'new', 'published', 'pending' and 'retired' |
| Version | Integer | The version of this form |
| SubmissionCount | Integer | The number of submissions associated with this specific form ID from the lineage of forms |

## Examples

## Examples

REST API call to pull all submissions for a specific account:

```
https://www.gocanvas.com/apiv2/csv_meta_datas.xml?username=test@test.com&password=test_pass
word&form_id=1&begin_second=962461752&end_second=1593613752
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
 <CSVMetaData>
  <Form Id="1" OriginatingLibraryTemplateId="" GUID="f6167220-a09d-11df-8e32-
00242bb0cd70">
    <Name>test form</Name>
    <Status>retired</Status>
    <Version>1</Version>
    <SubmissionCount>1</SubmissionCount>
  </Form>
  <Form Id="2" OriginatingLibraryTemplateId="" GUID="f6167220-a09d-11df-8e32-
00242bb0cd70">
    <Name>second form</Name>
    <Status>published</Status>
```

```
    <Version>2</Version>
    <SubmissionCount>5</SubmissionCount>
  </Form>
 </CSVMetaData>
</CanvasResult>
```

## CSV API

### Description

The CSV API is to download a CSV (not wrapped in XML) from a specific form version for a specific date range.

### URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

CSV Meta Data API URL: https://www.gocanvas.com/apiv2/csv.xml

### Parameters

The CSV API requires valid GoCanvas username and password to access content, in addition to a specific form ID and a date range. If Departments is enabled, and if a department isn't specified, it will download the information from the default department, otherwise you can pass in the name of the department to download the information in the department.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| form_id | Integer | No | Query submission count and form information associated with the form which has the specified form_id as a unique identifier |
| begin_date | Date (MM/DD/ | No | Query submission count and form information submitted on or after this date |

| | | | |
|---|---|---|---|
| | YYYY) | | |
| end_date | Date (MM/DD/ YYYY) | No | Query submission count and form information submitted on or before this date |
| begin_second | Integer | No | Query submission count and form information submitted at or after the specified time expressed as the seconds since Unix epoch.  This is intended to allow more fine grained control than begin_date. |
| end_second | Integer | No | Query submission count and form information submitted at or before the specified time expressed as the seconds since Unix epoch. This is intended to allow more fine grained control than end_date |
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as.  You must have permission to the other users account via the Associate functionality. |
| department | String(case insensitive) | No | Name of the department that you want the information downloaded from |

## Results

The results of this web service invocation (if it is successful) is a CSV file.  Note, that this result is not encapsulated in any XML structure.

## Examples

REST API call to pull all submissions for a specific account:

```
https://www.gocanvas.com/apiv2/csv.xml?username=test@test.com&password=test_password&form_id=1&begin_second=962461752&end_second=1593613752
```

```
"Start Date","End Date","Export App Name","Username","User Last Name","User First Name","Response ID","Response Date/Time","Device Date/Time","Submission Form Name","Submission Form Version"

2000-07-26 00:00:00","2020-07-26 23:59:00","Form Name","test@test.com","Test","User","B9E3AB25-EAE3-4A55-9136-E7CC921ED01E","07/26/2011 15:04:29","07/26/2011 15:04:09","Form Name"
```

## Dispatch Items API

### Description

Dispatch items implement the concept of prepopulated submissions which may be assigned to any of your company users. There are a couple of restrictions as to what fields can be filled in advance, those should be "simple" fields, like text boxes, check boxes, integer fields, etc, as opposed to more "complex" fields such as images, drawings, signatures, etc.

In order to use this feature, your company must have it enabled. Each form will have to have in enabled from the "App Features" menu option in the "Apps" page. One last requirement is that at least one published version of your app must exist before any items can be added to it.

Editing is specified by the "Action" parameter.  When set to "Update" it will update an existing submission with the "Description" Parameter.  To update the "Description" of a Dispatch, provide the "OriginalDescription" parameter. When set to "Delete", provide the "OriginalDescription" to determine which dispatch to delete. The Dispatch must have the "Unassigned" or "Assigned" status to edit it.  The Dispatch can be any status to delete it.

This API also supports the dispatching of lists. It will only support simple lists and multi-screen single-level lists but not advanced/nested lists.  When a Dispatch List is upload, in the payload, the dispatch entries will belong to <DIListItems> as shown below.

URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.  This URL and the necessary XML can only be accessed via a POST to the appropriate URL.

Dispatch Items API URL: https://www.gocanvas.com/apiv2/dispatch_items

### Parameters

The Dispatch Items API is restricted to authenticated users and requires a username and password to access. If these two fields aren't specified, an error is returned.

| Name | Data Type | Required | Description |
|---|---|---|---|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| act_as_us er | String (case insensitive) | No | E-mail address of the user to act as. You must have permission to the other users account via the Associate functionality. |

## Payload

The payload to send must be a well formed XML document consisting of a "List" element, wrapping up one or more "DI" entries. Each of those DI entries will have an "AppName" mandatory attribute, as well as 3 other optional attributes like "Description", "User", and "ScheduleAt". Lastly, each DI element will also contain at least one "DIEntry" sub-element with its corresponding "Label" and "Value" attributes. The "Department" attribute can be added to specify the department that you want the dispatch to be uploaded to.

List data and descriptions:

| Name | Data Type | Description |
|---|---|---|
| FormName | String | The name of the app you want to create a prepopulated submission for. |
| Action | String | Optional. Values: "Update", "Create"(defaults to "Create"), and "Delete". When set to "Update" it will replace an existing dispatch item with the same Description. When set to "Delete", it will delete the existing dispatch item. |
| Description | String | Optional. A brief description of this item. If none given either the submission numbering or naming settings will be used depending on what's been set for the app. |
| OriginalDescription | String | Optional. Only used if "Action" is set to "Update" or "Delete". If "Update" is provided this will be used to find the Dispatch to update. If not provided "Description" will be used. If "Delete" is provided, this will be used to delete the Dispatch. |
| User | String | Optional. The email address of the user this item will be assigned to. |
| ScheduleAt | String | Optional. Only used if "Action" is set to "Update" or "Create". If "Create" is provided this will be used to to create a Dispatch Item with a ScheduleAt date/time. If "Update" is provided, this will be used to update the dispatch item with the new ScheduleAt date/time. The date format must be in 'DD/MM/YYYY hh:mm' , 'MM/DD/YYYY hh:mm' , or 'YYYY/MM/DD hh:mm' depending on the company date/time format that is currently set. |
| Department | String | Optional. Used to upload dispatches to specified department |
| Duration | Integer | Optional. Used with Dispatch Calendar. The number of minutes the Dispatch is scheduled for. |

| Reminder | Integer | Optional. Used with Dispatch Calendar. The number of minutes before the Dispatch is scheduled that a calendar invite will be scheduled |
| DispatchNa me | String | Optional. Used with Dispatch Calendar. The name of the Dispatch that will appear in the Calendar view |
| SendInvite | String | Optional. Used with Dispatch Calendar. "YES" or "NO" if set to "YES" a calendar invite will be sent to the user. |

Each DIEntry element must consist of:

| Name | Data Type | Description |
|------|-----------|-------------|
| Label | String | Either the Export Label or plain Label field attribute (case-insensitive) as defined in the form builder. |
| Value | String | The value assigned to this Dispatch Item enty. |

## Results

If the request is processed correctly, then an empty CanvasResult element will be returned, with a response code of 200 from the web server. If an error occurs while processing the request, one of the standard errors detailed in the Error Codes section will be returned.

## Examples

REST API call:

https://www.gocanvas.com/apiv2/dispatch_items?username=test@test.com&password=test_password

Post Payload:

```
<?xml version="1.0" encoding="utf-8"?>

<List>

  <DI FormName="Simpsons App" Description="#1 item" User="homer@mailinator.com"
ScheduleAt="01/23/2015 09:15" Department="Test Department">

    <DIEntry Label="first name" Value="Homer"/>

    <DIEntry Label="last name" Value="Simpson"/>

  </DI>

  <DI FormName=" Simpsons App">

    <DIEntry Label="first name" Value="Marge"/>

    <DIEntry Label="age" Value="35"/>

  </DI>

</List>
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
</CanvasResult>
```

Post Payload with Dispatch List:

```xml
<?xml version="1.0" encoding="utf-8"?>
<List>
 <DI FormName="Your App" Description="1st item" User="myemail@email.com">
  <DIEntry Label="First Label" Value="1st value"/>
  <DIEntry Label="Second Label" Value="2rd value"/>
  <DIListItems>
   <DIListItem>
    <DIEntries>
     <DIEntry Label="Name" Value="Tony Stark"/>
     <DIEntry Label="Nick Name" Value="Iron Man"/>
     <DIEntry Label="Occupation" Value="Super Hero"/>
    </DIEntries>
   </DIListItem>
   <DIListItem>
    <DIEntries>
     <DIEntry Label="Name" Value="Bruce Banner"/>
     <DIEntry Label="Nick Name" Value="Hulk"/>
     <DIEntry Label="Occupation" Value="Super Hero"/>
    </DIEntries>
   </DIListItem>
  </DIListItems>
 </DI>
</List>
```

Post Payload for Dispatch Calendar:

```
<?xml version="1.0" encoding="utf-8"?>

<List>

  <DI FormName="Simpsons App" Description="#1 item" User="homer@mailinator.com"
ScheduleAt="01/23/2015 09:15" Department="Test Department" Duration="60"
Reminder="10" DispatchName="Item 1" SendInvite="YES">

    <DIEntry Label="first name" Value="Homer"/>

    <DIEntry Label="last name" Value="Simpson"/>

  </DI>

  <DI FormName=" Simpsons App">

    <DIEntry Label="first name" Value="Marge"/>

    <DIEntry Label="age" Value="35"/>

  </DI>

</List>
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
</CanvasResult>
```

# Dispatch Export API

## Description

The Dispatch Export APi provides the capability to retrieve the dispatch items just like you
would from the dispatch export from the website.
## URL
Since this API call and results could potentially contain sensitive information, it is required to be
accessed over SSL via HTTPS.

Dispatch Export API URL: https://www.gocanvas.com/apiv2/dispatch_export.xml

## Parameters

The Dispatch Export API is restricted to authenticated users and requires a username and password to access. All other parameters are optional. When no parameters are specified, a list of all dispatch items for the user  is returned.

| Name | Data Type | Required | Description |
|---|---|---|---|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| begin_date | Date(MM/DDYYYY) | No | Return all dispatch items created on or after this date |
| end_date | Date(MM/DDYYYY) | No | Return all dispatch items created on or before this date |
| begin_second | Integer | No | Return all dispatch items created at or after the specified time expressed as the seconds since Unix epoch.  This is intended to allow more fine grained control than begin_date. |
| end_second | Integer | No | Return all dispatch items created at or after the specified time expressed as the seconds since Unix epoch.  This is intended to allow more fine grained control than begin_date. |
| Include_submissions | Boolean | No | Returns dispatch items with submissions. Must also use begin/end date or begin/end second with it. |
| form_id | Integer | No | Returns dispatch items by form id and all the versions of the form |
| department | String (non-empty/case sensitive) | No | Returns dispatch item in the department if using dispatch_id, dispatch_name or dispatch_description |
| dispatch_id | Integer | No | Returns dispatch item by dispatch id |
| dispatch_name | String (non-empty/case sensitive) | No | Returns dispatch item by dispatch name |
| dispatch_de | String (non- | No | Returns dispatch item by dispatch |

| scription | empty/case sensitive) | | description | |
|-----------|-----------------------|--|-------------|--|
| | | | | |

## Examples

REST API call to pull dispatch items that the user has access to:

https://www.gocanvas.com/apiv2/dispatch_export.xml?username=test@test.com&password=test_password

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
 <TotalPages>1</TotalPages>
 <CurrentPage>1</CurrentPage>
 <Dispatches>
  <Dispatch Id="1234">
   <Name/>
   <Description>Dispatch test</Description>
   <User>homer.simpson @gocanvas.com</User>
   <Form>TestForm</Form>
   <Status>completed</Status>
   <CreatedAt>2015-12-10 15:40:07 UTC</CreatedAt>
   <CompletedAt>2016-01-13 19:59:29 UTC</CompletedAt>
   <ScheduledAt>2015-12-05 18:02:00 UTC</ScheduledAt>
   <Duration>120.0</Duration>
   <Reminder/>
   <Entries>
    <Entry>
     <Value>Bart</Value>
    </Entry>
   </Entries>
  </Dispatch>
  <Dispatch Id="1235">
   <Name/>
   <Description>
     Best 90's cartoon
   </Description>
   <User>bart.simpson@gocanvas.com</User>
   <Form>SomeForm</Form>
   <Status>received</Status>
```

```
   <CreatedAt>2016-01-21 16:16:31 UTC</CreatedAt>
   <CompletedAt/>
   <ScheduledAt>2016-01-20 03:49:00 UTC</ScheduledAt>
   <Duration>120.0</Duration>
   <Reminder/>
  <Entries>
   <Entry>
    <Value>a1X1A000003M6vLUAS</Value>
   </Entry>
  </Entries>
 </Dispatch>
```

REST API call to pull dispatch items that the user has access to with date range and submissions:

```
https://www.gocanvas.com/apiv2/dispatch_export.xml?
username=test@test.com&password=test_password
&begin_date=01/01/2014&end_date=03/01/2018&include_submissions=true
```

Resulting XML:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
 <TotalPages>1</TotalPages>
 <CurrentPage>1</CurrentPage>
 <Dispatches>
  <Dispatch Id="1234">
   <Name/>
   <Description>Dispatch test</Description>
   <User>homer.simpson @gocanvas.com</User>
   <Form>TestForm</Form>
   <Status>completed</Status>
   <CreatedAt>2015-12-10 15:40:07 UTC</CreatedAt>
   <CompletedAt>2016-01-13 19:59:29 UTC</CompletedAt>
   <ScheduledAt>2015-12-05 18:02:00 UTC</ScheduledAt>
   <Duration>120.0</Duration>
   <Reminder/>
   <Entries>
    <Entry>
     <Value>After LA1 &amp; Using amp; the end</Value>
    </Entry>
   </Entries>
   <Submission Id="35901" Department="First Department">
```

```
<Form Id="605853">
 <Name>test</Name>
 <Status>retired</Status>
 <Version>6</Version>
</Form>
<Date>2016.01.13 19:59:29</Date>
<DeviceDate>2016.01.13 19:59:18</DeviceDate>
<UserName>jeff.li@gocanvas.com</UserName>
<FirstName>jefferson</FirstName>
<LastName>li</LastName>
<ResponseID>ba8db2a506892eb-1452715136070</ResponseID>
<SubmissionStatus>No Status</SubmissionStatus>
<Sections>
 <Section>
  <Name>1</Name>
  <Screens>
   <Screen>
    <Name>1</Name>
     <Responses>
      <Response Guid="83BD5371CEDBA3012467AD57333320369B3140FD">
       <Label>multiline</Label>
       <Value>jeff22asdf</Value>
       <Type>Multi-Line Text</Type>
      </Response>
     </Responses>
    </Screen>
   </Screens>
  </Section>
 </Sections>
</Submission>
</Dispatch>
<Dispatch Id="1235">
 <Name/>
 <Description>
   Best 90's cartoon
 </Description>
 <User>bart.simpson@gocanvas.com</User>
 <Form>SomeForm</Form>
 <Status>received</Status>
 <CreatedAt>2016-01-21 16:16:31 UTC</CreatedAt>
 <CompletedAt/>
 <ScheduledAt>2016-01-20 03:49:00 UTC</ScheduledAt>
 <Duration>120.0</Duration>
 <Reminder/>
 <Entries>
  <Entry>
```

```
      <Value>a1X1A000003M6vLUAS</Value>
    </Entry>
  </Entries>
</Dispatch>
```

# Department API

## Description

The Departments API provides the capability to retrieve all or a subset of departments associated with the specified login.

## URL

Since this API call and results could potentially contain sensitive information, it is required to be accessed over SSL via HTTPS.

Departments API URL: https://www.gocanvas.com/apiv2/departments.xml

## Parameters

The Departments API is restricted to authenticated users and requires a username and password to access. All other parameters are optional. When no parameters are specified, a list of all departments applicable for that user is returned.

| Name | Data Type | Required | Description |
|------|-----------|----------|-------------|
| username | String (non-empty/case insensitive) | Yes | Valid GoCanvas username |
| password | String (non-empty/case sensitive) | Yes | Valid GoCanvas password associated with username |
| search_name | String (case insensitive) | No | Return all departments whose name contains the search_name. |
| act_as_user | String (case insensitive) | No | E-mail address of the user to act as.  You must have permission to the other users account via the Associate functionality. |

## Results

The result of the departments web service invocation is a collection of departments.

Department data and description:

| Name | Data Type | Description |
|------|-----------|-------------|
| Name | String | The name of the department |

## Examples

REST API call:

https://www.gocanvas.com/apiv2/departments.xml?username=test@test.com&password=test_password

Resulting XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<CanvasResult>
  <Departments>
    <Department Id="582">
      <Name> Tech Department</Name>
    </Department>
    <Department Id="3">
      <Name>Design Department</Name>
    </Department>
  </Departments>
</CanvasResult>
```

# Submission Push Notifications

## Description

GoCanvas offers the option to notify a web service that you create any time a submission is uploaded to our service. You can find this in the "Integration Options" section for your GoCanvas application.

To use this service, you'll need to implement a web service that can accept and process posted XML data. When a user uploads a submission, we'll send an HTTP POST request to your service containing metadata about the submission in XML form.

## URL

Since this is an outbound service from GoCanvas to you, not one that you call, there is no URL.

## What GoCanvas Sends to Your Service

For any given submission, GoCanvas will send the submission GUID, the name, ID, and guid of your form, and will optionally include a tag that you define (see the "Setting Up" section below). This will be contained within an XML document as the body of the HTTP POST. Here's an example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<submission-notification>
 <form>
  <id type="integer">1</id>
  <name>FORM_NAME</name>
  <guid>FORM_GUID</guid>
  <tag>Your Tag</tag>
 </form>
 <submission>
  <guid>SUBMISSION_GUID</guid>
 </submission>
</ submission-notification >
```

If you've defined no tag to be sent back to you, the element will be present but empty.

## Implementation Notes

1. GoCanvas does not read or process the body or headers of your web service's response, except for the status code. It is not necessary to include a body in your response.

2. GoCanvas will consider any HTTP status code that is not in the range of 200-205 to be an error, and will queue the notification to be retried later.

3. In the event of a failure, GoCanvas will wait 15 minutes and then try again. On subsequent failures, GoCanvas adds an additional 15 minutes onto the wait time before re-trying. After 10 failures, GoCanvas will give up and email your account admin automatically.

4. GoCanvas does not follow HTTP redirects. Redirects are treated as failures and will be re-tried.

## Setting Up

Once you've created your web service to handle the notifications we'll send you can try it out in the GoCanvas web interface. Visit the "Integration Options" section for the form for which you want to receive notifications, and click the "Activate" button below the GoCanvas logo to the right of the Salesforce Integration  item. (Note that GoCanvas presently only allows one type of integration to be active per form at any given time, so if you already have an integration you'll have to delete it to set this up). Enter the fully-qualified URL of your web service end point including the scheme (which can be either http:// or https://), and optionally enter a tag you'd like us to include with your metadata. The tag

can be any arbitrary text that will help you to identify or categorize submissions by form within your web service.

Once you've provided a URL (and, optionally, a tag), click the "Test" button. GoCanvas will send some sample metadata to the URL you've defined, and display the response for you so that you can see how your web service responded to the posted data. GoCanvas will display the status code, headers, and response body – if any – that we received from your web service. You can use this to determine whether your service is working as intended and make adjustments if you need to.

When you're satisfied that your service is responding as you intended, go ahead and hit the "Save" button to start receiving notifications when submissions are uploaded.

# Error Codes

## Description

Errors can occur when invoking the GoCanvas web services for a number of reasons, such as permission problems, requested data not being found or numerous other scenarios.  When invoking any of the GoCanvas web service APIs, if an error occurs it will be returned in the same format regardless of the API being invoked.

## Results

The format of the error response coming back will be a single "error" node contained within the standard GoCanvas "CanvasResult" node.  If more than one error occurs during execution, only one will be selected and returned.

Error data and descriptions:

| Name | Data Type | Description |
|------|-----------|-------------|
| ErrorCode | Integer | A number representation of the error which occurred |
| Description | String | Adescription of the error which occurred |

## Error Codes

| ErrorCode | Description |
|-----------|-------------|
| 1 | Permission denied to the requested resource |
| 2 | Requested resource not found |
| 10 | Required parameter is missing |
| 11 | Parameter is of an invalid format |

| 30 | Request canceled due to a resource timeout or exceptionally large dataset |
|----|---------------------------------------------------------------------------|
| 40 | Server error, please try again shortly |

## Server Response Codes

An error being successfully detected and handled, will return a server response code of 200 with the payload described above.  In the unlikely event there is an unexpected fault on the server side (hardware, memory or other issues) a server response code of 5xx may be returned.  Implementing software should be able to handle an error code with a server response code of 200, or a more low level server response code of 5xx with an unknown payload.

## Examples

Example of an invalid web service invocation (missing the required "id" parameter)

https://www.gocanvas.com/apiv2/images.xml?username=test@gocanvas.com@password=test

Result:

```
<?xml version="1.0" encoding="utf-8"?>
<CanvasResult>
  <Error>
    <ErrorCode>10</ErrorCode>
    <Description>Required parameter is missing</Description>
  </Error>
</CanvasResult>
```