There is a simple way to save data directly into Spreadsheets, Google Sheets or Dropbox files from webpage designed to collect information as googleform.

Since most of our work goes well with excel sheets, Let’s see how to do it using our tool [API Spreadsheets](https://www.apispreadsheets.com/).

**NOTE:**This does require you signing up for our tool [API Spreadsheets](https://www.apispreadsheets.com/). They have a very generous free tier that should be more than enough for our needs for now. However, they offer professional membership for $10/month and teams(~10 people) membership for $25… just something to be aware off.

**Step 1: Set up a contact form in an HTML file**

*Okay, if you don’t know any HTML, then this might be a steep learning curve.* Let’s create a form in HTML. It’s a basic contact form and doesn’t contain any styling.

The form will look like this

Open up a blank file in the text editor you use for programming, copy and paste the code below and save the file with a [.html] extension. We saved ours as ContactForm\_preconsult.html

We will walk through the HTML part now and the Javascript part later.

The contact form has 4 fields.

Pay attention to the nameattributeof the <input> tags. These will be the column headers of our spreadsheet we save the data in and they MUST match up.

We are going to name the four input tags as follows. The lines refer to the line of code:

1. **full\_name, line 29**

* This will be a text field with the full name of the person

2. **email, line 33**

* This will be a text field with the email of the person

3. **age, lines 37–38**

* This will be a radio selection field with two age options. **18–35**and **35+**. We will denote them in the value attributeof the radio fields

4. **message, line 42**

* This will be a textarea that will contain the message the person types

All four <input> tags will be inside a <form> tag (line 26)that we will give an id attributeof **myForm.**Do not give it any other idas the data submission to spreadsheet function that we will write later in the Javascript is dependent on it.

Finally, our main component will a <button> (line 47)that will be submitting the data. In its onclick attributewe are going to specify the function **SubForm()**.

Again, do not give the function another name as we will be writing this exact function later in Javascript to handle submitting the data.

# ****Step 2: Prepare your spreadsheet that the data will be saved to****

This part is straightforward.

**2*.*Write column headers equal to the*name(s)*of the <input> tags**

Each time someone submits the data, it will save in the appropriate column. *This step is super important so make sure these headers are EXACTLY the same as the names of the input tags above.*

1. **Save this spreadsheet with any name you want**

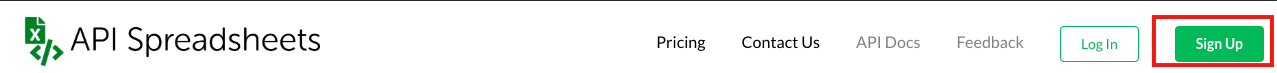
**A screenshot of a computer

Description automatically generated**

**Step 3: Get your API URL from API Spreadsheets**

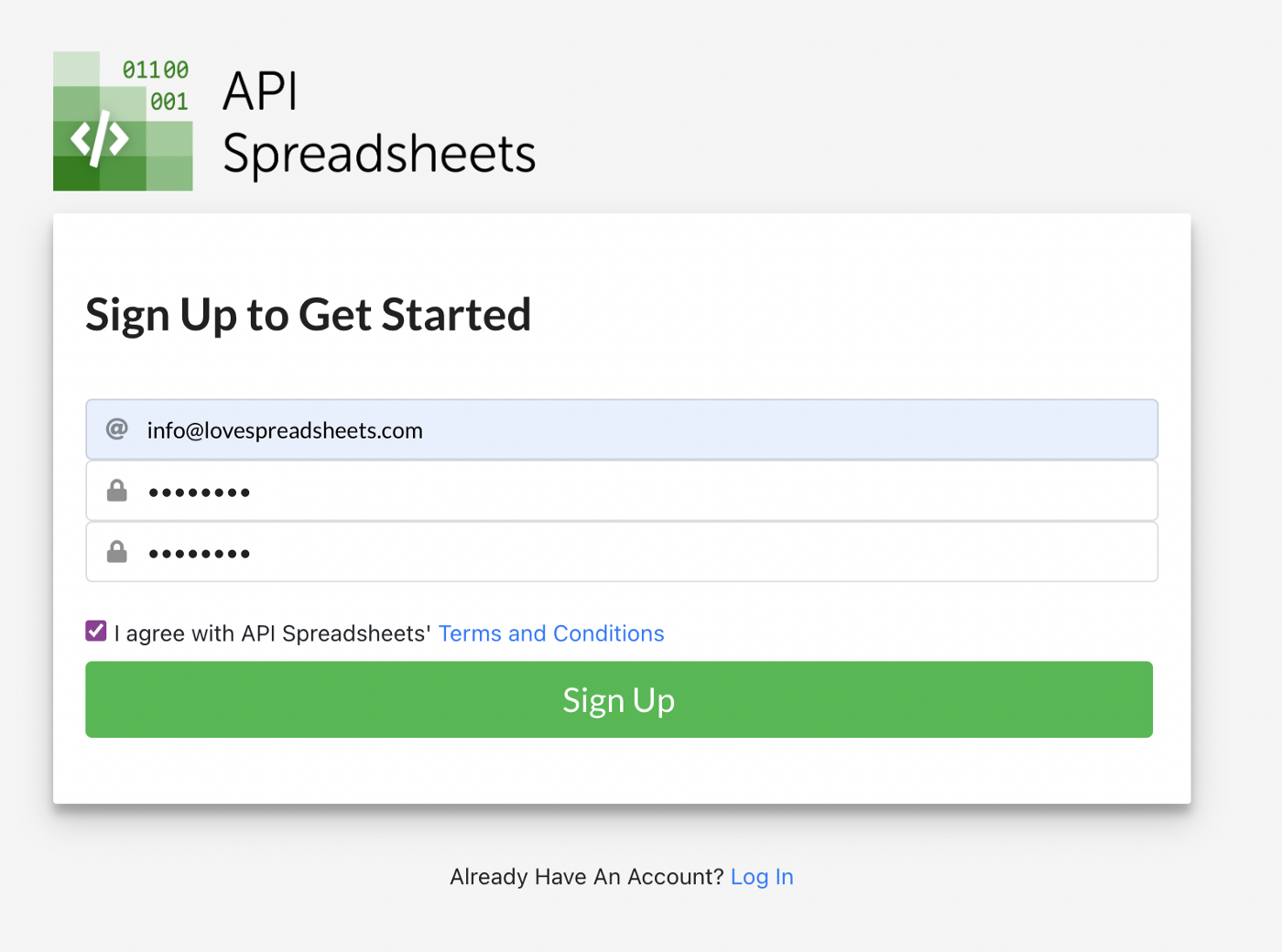
1. **Go to**[**www.apispreadsheets.com**](http://www.apispreadsheets.com)**and sign up for an account**

* Click on Sign up in the Navbar



Sign Up Button on the Toolbar

* Sign up with your email and password



2. **Upload the file you created in Step 2**

* Click on Upload or Drop Files and select your file from Step 2

Graphical user interface, application, website

Description automatically generated

3. **Copy the Create API URL for your file**

* Copy the API URL for your file and save it somewhere handy. We are going to be using this in the Javascript below to submit the data.

Graphical user interface, application, Teams

Description automatically generated

4*. [Optional] If your API is****Private****you will also need to copy the Access and Secret Key and save them somewhere handy*

Graphical user interface, text, application

Description automatically generated

**Step 4: Configure the Javascript to submit data from the form**

*We are going to use AJAX to submit the form. Again, if you are not familiar with Javascript, jQuery or AJAX, it might be a steep learning curve. I am here to help!*

We are going to include the full HTML code here again. The Javascript part is within the <head> tags (*lines 5–23)*.

1. **Add jQuery from a CDN *(lines 5–8)***

We need to ensure the FULL jQuery library is included in our HTML to use AJAX.

<script src="https://code.jquery.com/jquery-3.4.1.js" integrity="sha256-WpOohJOqMqqyKL9FccASB9O0KwACQJpFTUBLTYOVvVU=" crossorigin="anonymous"></script>

2. **Write the SubForm() function between two script tags (*lines 9–23)***

The **SubForm()**function is below. You will replace the **URL**below with the URL you obtained in **Step 3.**The rest of the function will remain the same. We will go through this function in more detail at the end of this post but for now all you need to know is if the data was successfully saved an alert will pop up saying **Form Data Submitted :)**otherwise an alert will pop up saying **There was an error :(**.

<script>  
function SubForm (){  
 $.ajax({  
 url:'https://api.apispreadsheets.com/data/410/',  
 type:'post',  
 data:$("#myForm").serializeArray(),  
 success: function(){  
 alert("Form Data Submitted :)")  
 },  
 error: function(){  
 alert("There was an error :(")  
 }  
 });  
}  
</script>

That’s it! We have done everything needed to save data from our web form to our spreadsheet.

Now let’s test it out and see how to view our data.

**Testing**

1. **Save your HTML file and open it in a browser to see the contact form**

A screenshot of a computer

Description automatically generated

It should look mostly like this depending on the browser

2. **Fill out your form and click *Submit***

An alert window should pop up saying **Form Data Submitted :)**

Graphical user interface, application, Word

Description automatically generated

Yayy! It worked!

If the alert window says **There was an error :(**then feel free to email us.

3. **View your data in your spreadsheet!!**

* Go to [www.apispreadsheet.com](http://www.apispreadsheet.com)
* *Log in if necessary*
* Click on the **Files**tab



* Find your spreadsheet and click on the **Download** **File button**to download it