

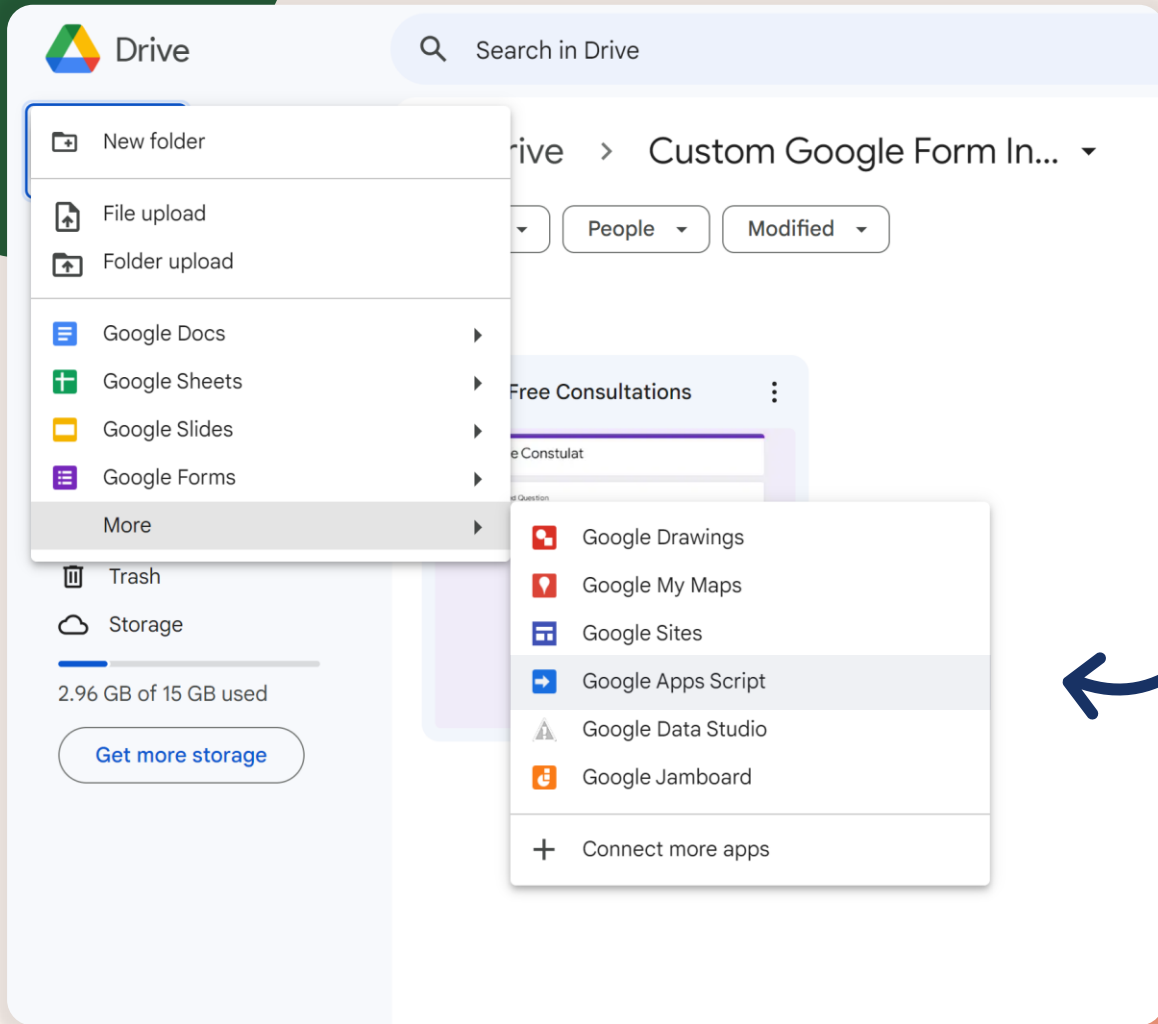


# **Install your custom Google Form.**

# Install process.

<b>1.</b>	<b>Initialize Google Apps Script</b>	<b>3</b>
<b>1.1</b>	<b>Create Your Google Apps Script</b>	<b>3</b>
<b>1.2</b>	<b>Insert Apps Script</b>	<b>4</b>
<b>1.3</b>	<b>Deploy Your Apps Script</b>	<b>5</b>
<b>2.</b>	<b>Update HTML Document</b>	<b>7</b>
<b>3.</b>	<b>Launch Your Form</b>	<b>9</b>
<b>4.</b>	<b>Insert a Google Calendar</b>	<b>10</b>
<b>4.1</b>	<b>Update Your Google Apps Script</b>	<b>10</b>
<b>4.2</b>	<b>Add a Calendar to Your Form</b>	<b>13</b>
<b>4.3</b>	<b>View Your New Form</b>	<b>15</b>


NEW



### Step 1.1

## Create your Google Apps Script.

Head over to your Google Drive. Navigate to where your Google Form is located. In that location add a Google Apps Script file.




Name	Status	Date modified	Type
css	✓	10/12/2023 4:00 PM	File folder
js	✓	10/16/2023 1:33 PM	File folder
php	✓	10/12/2023 4:00 PM	File folder
index.html	✓	10/16/2023 1:11 PM	Chrome HTML Do...

## Step 1.2

# Insert App Script Code.

Once the Google Apps Script file is created head over to the application's folder. Navigate to the JS folder. Open the google-apps-script.js file. Copy the code and paste it into the Google Apps Script. After the code has been pasted, delete the google-apps-script.js file.

Name	Status	Date modified	Type
google-apps-script.js	✓	7/24/2023 12:37 PM	JavaScript Source ...
google-form script is	✓	10/12/2023 4:02 PM	JavaScript Source



Files

Code.gs

Libraries

Services

1. GLOBAL VARIABLES

// 1.1 stores the Form Service script (will be set in the doPost function)

let form;

// 1.2 stores the form's item (will be set in the doPost function)

let form\_items;

// 1.3 stores the form's item types

const item\_types = {

text: "TEXT",

multipleChoice: "MULTIPLE\_CHOICE",

paragraphText: "PARAGRAPH\_TEXT",

checkbox: "CHECKBOX",

list: "LIST",

scale: "SCALE",

grid: "GRID",

checkboxGrid: "CHECKBOX\_GRID",

date: "DATE",

time: "TIME",

fileUpload: "FILE\_UPLOAD",

image: "IMAGE",

sectionHeader: "SECTION\_HEADER",

pageBreak: "PAGE\_BREAK",

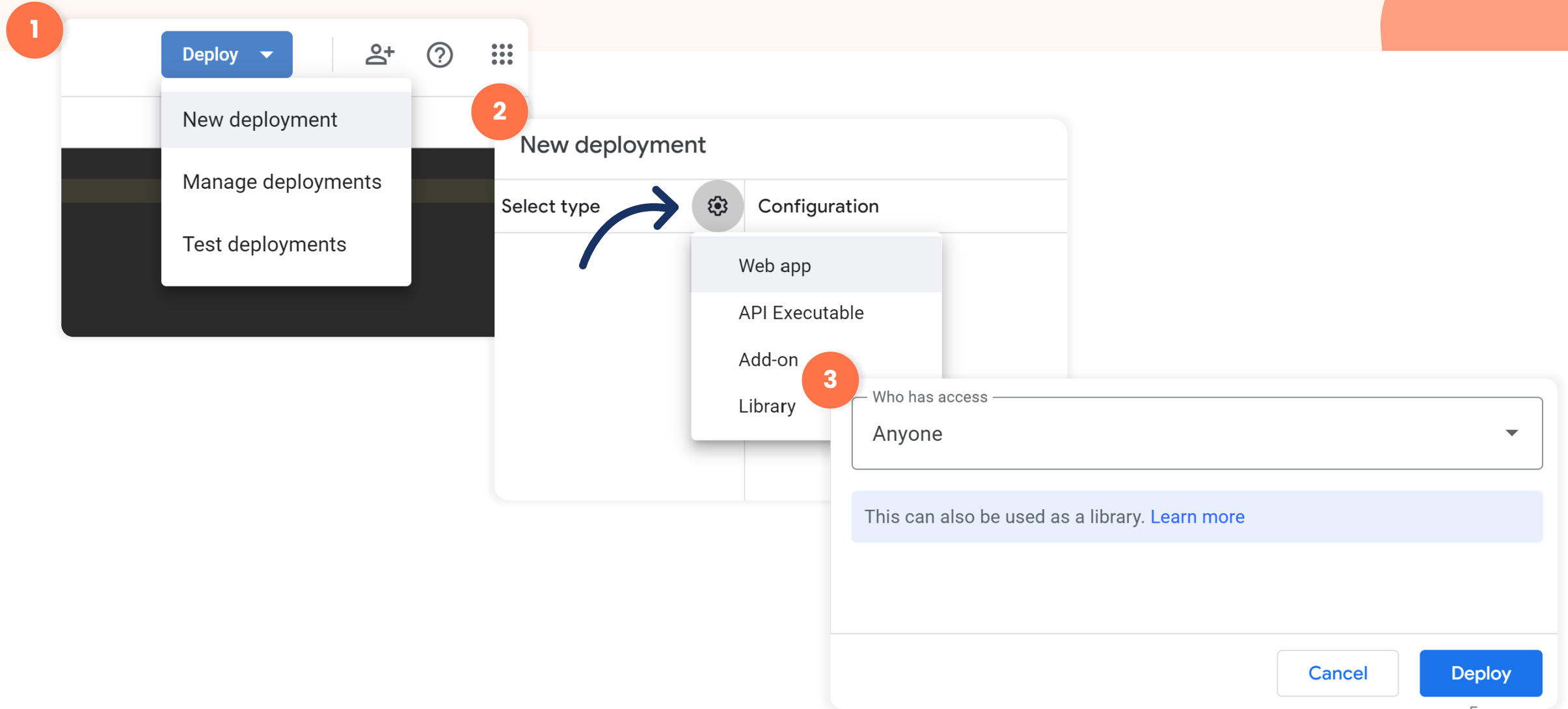
video: "VIDEO",

};

### Step 1.3

## Deploy your Google Apps Script.

In the Google Apps Script, select the deploy button. Create a new deployment. Set it as a web app. For who has access, set it as anyone. When done, deploy the script.



## Step 1.3 Continued

# Deploy your Google Apps Script.

Authorize the script using your Google account. After that copy the Web App URL. When done close the Google Apps Script.

1

### New deployment

The Web App requires you to authorize access to your data.

Authorize access



2

Sign in with Google

Untitled project wants to access your Google Account

irenegibsona7@gmail.com

allow Untitled project to:

View and manage your forms in Google Drive

Make sure you trust Untitled project

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

Learn how Google helps you share data safely

See Untitled project's Privacy Policy and Terms of Service.

Cancel

Allow



3

Web app


URL

<https://script.google.com/macros/s/AMTycbwWpDg8tMwChZTh-g8U7YtZJ2w4t>

Copy



1



Name	Status	Date modified	Type
css	✓	10/12/2023 4:00 PM	File folder
js	✓	10/16/2023 1:33 PM	File folder
php	✓	10/12/2023 4:00 PM	File folder
index.html	✓	10/16/2023 1:11 PM	Chrome HTML Do...


2

```
<body>

<script>

  new GoogleForm({
    form: {
      id: "ENTER FORM ID",

```



docs.google.com/forms/d/1IEa6C7NUG2DfdLAss8zY6BVO-eefj-uTd6puFnJF\_FE/edit

```
<script>

  new GoogleForm({
    form: {
      id: "ENTER FORM ID",
      webAppURL: "ENTER WEP APP URL"
    },
    redirect: "ENTER REDIRE",
    recaptcha: {
      sitekey: 'ENTER SITEI',
      secretKey: 'ENTER SEI'
    }
  });
</script>
```

3

```
<script>

  new GoogleForm({
    form: {
      id: "ENTER FORM ID",
      webAppURL: "ENTER WEP APP URL"
    }
  });
</script>
```

## Step 2

# Update the HTML Document.

In the application's folder open the index.html file. Locate the Google form script. Insert the Web App URL. Insert your form's ID. To retrieve your form's ID, open the form. Copy its ID from the URL.

The redirect and reCAPTCHA attributes are optional. The redirect attribute is a URL. After the form is submitted, the redirect will send the user to a confirmation message. To add Google reCAPTCHA insert a site and secret key. If you do not wish to add reCAPTCHA or a redirect remove the attributes.

## Step 2 Continued

# Update the HTML Document.

Below are some examples on how to use the redirect and reCAPTCHA attribute.

```
<script>

new GoogleForm({
  form: {
    id: "ENTER FORM ID",
    webAppURL: "ENTER WEB APP URL"
  },
  redirect: "/pages/thank-you.html",
  recaptcha: {
    sitekey: 'ENTER STIEKEY',
    secretKey: 'ENTER SECRET KEY'
  }
});

</script>
```

Adding a shorthand redirect URL

```
<script>

new GoogleForm({
  form: {
    id: "ENTER FORM ID",
    webAppURL: "ENTER WEB APP URL"
  },
  redirect: "https://mywebsite.com/pages/thank-you.html",
  recaptcha: {
    sitekey: 'ENTER STIEKEY',
    secretKey: 'ENTER SECRET KEY'
  }
});

</script>
```

Adding a full redirect URL

```
<script>

new GoogleForm({
  form: {
    id: "ENTER FORM ID",
    webAppURL: "ENTER WEB APP URL"
  },
  redirect: "ENTER REDIRECT URL",
  recaptcha: {
    sitekey: '6Lfn7vMhAAAAAICH6aYXTiqq8PHmvtFg93bKdV3z',
    secretKey: '6Lfn7vMhAAAAAGZd3bqlvkoH9hd5fjIV9_g70y0U'
  }
});

</script>
```

Adding Google reCAPTCHA.





### Step 3

# Launch your form.

Your form is ready to go live. Upload the index.html document, CSS folder, JS folder, and PHP folder to your server.

Contact Us

---

First Name

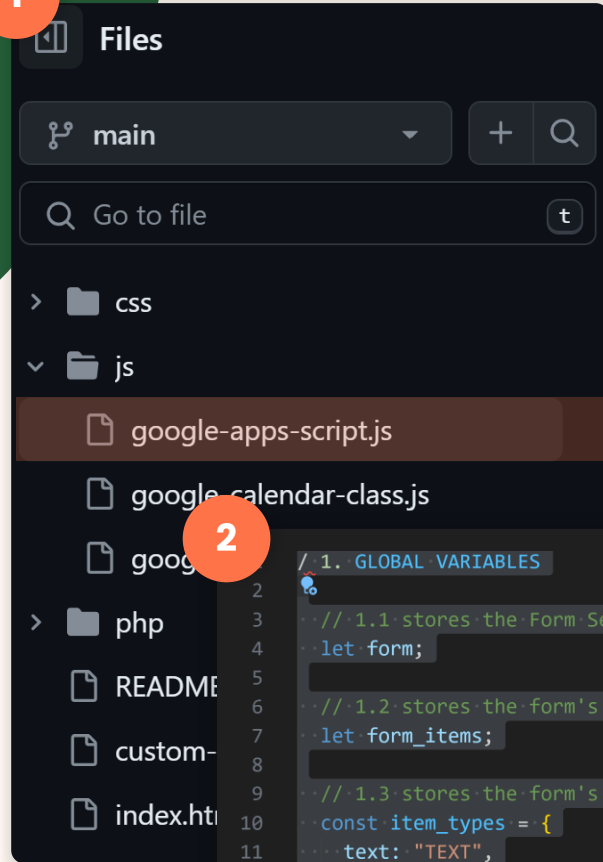
Last Name

Email

Message

Submit

1



2

```
// 1. GLOBAL VARIABLES
// 1.1 stores the Form Service script (will be set in the doPost function)
let form;
// 1.2 stores the form's item (will be set in the doPost function)
let form_items;
// 1.3 stores the form's item types
const item_types = {
  text: "TEXT",
  multipleChoice: "MULTIPLE_CHOICE",
  paragraphText: "PARAGRAPH_TEXT",
  checkbox: "CHECKBOX",
  list: "LIST",
  scale: "SCALE",
  grid: "GRID",
  checkboxGrid: "CHECKBOX_GRID",
  date: "DATE",
  time: "TIME",
  fileUpload: "FILE_UPLOAD",
  image: "IMAGE",
  sectionHeader: "SECTION_HEADER",
}
```

## Step 4.1

# Insert a Google Calendar.

Head over to GitHub and download the google-apps-script.js file. Open the file and copy the file's contents.

## Step 4.1 Continued

# Update your Google Apps Script.

Open your Google Apps Script and replace the code with the code you copied. When done redeploy your apps script.

The image shows a sequence of four numbered steps for updating a Google Apps Script and managing its deployments:

- 1** The Google Apps Script editor interface. The left sidebar shows 'Files', 'Code.gs', 'Libraries', and 'Services'. The main editor area contains the following code:

```
1 // 1. GLOBAL VARIABLES
2
3 // 1.1 stores the Form Service script (will be set in the doPost function)
4 let form;
5
6 // 1.2 stores the form's item (will be set in the doPost function)
7 let form_items;
8
9 // 1.3 stores the form's item types
10 const item_types = {
11   text: "TEXT",
12   multipleChoice: "MULTIPLE_CHOICE",
13   paragraphText: "PARAGRAPH_TEXT",
14   checkbox: "CHECKBOX",
15   list: "LIST",
```
- 2** A close-up of the 'Deploy' button in the top right corner of the editor.
- 3** A close-up of the 'Deploy' dropdown menu, which includes the option 'Manage deployments' (highlighted with an orange arrow).
- 4** The 'Manage deployments' dialog box. It has a table with two columns: 'Active' and 'Configuration'. The 'Active' column contains the text 'Active'. The 'Configuration' column contains a question mark icon, a pencil icon (highlighted with a blue arrow), and a download icon.

## Step 4.1 Continued

# Update your Google Apps Script.

5

Configuration

Version

Version 1 on [blurred text]

New version

Version 1 on [blurred text]

Deployment ID

6

Cancel

Deploy

## Step 4.2

# Add a calendar to your form.

Open the HTML document where your form lives. Add a calendar object to the class. The object can take the following properties: ID, from, to, duration, and label. The ID is a Google Calendar ID.

From, is the start time. To, is the end time. The duration is how long the event will last. Lastly, the label is the calendar's title on the form.

```
calendar: {  
  id: "ENTER CALENDAR ID",  
  from: 9, // ENTER AN INTEGER BETWEEN 0 AND 23  
  to: 17, // ENTER AN INTEGER BETWEEN 0 AND 23  
  duration: 45, // ENTER AN INTEGER BETWEEN 0 AND 60  
  label: "ENTER A TITLE"  
}
```

## Step 4.3 Continued

# Retrieving your Google Calendar ID.

In your web browser, head over to your Google Calendar. Open your calendar settings and select the calendar you want to use and copy its ID.

**1** Click the gear icon in the top left of the Google Calendar interface.

**2** In the 'Settings for my calendars' panel, click the calendar you want to use.

**3** In the 'Settings for [calendar name]' panel, click 'Integrate calendar'.

**4** Copy the 'Calendar ID' from the 'Integrate calendar' page.

Calendar ID  
482240a3b4455ab9a4a5a962685478340c118a04a8b1a0a2cc7c208a4a4a4a200@group.calendar.g  
oogle.com



Step 4.3

# View your new form.

Launch your form and test your new form. The calendar feature can be used for booking an appointment.

Contact Us

Set an Appointment

April 2024

< >

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Open Slots

☐ 2:50 pm - 3:35 pm

☐ 3:40 pm - 4:25 pm

☐ 4:30 pm - 5:15 pm

Load More



**Thank you.**