

I'm not robot  reCAPTCHA

**Continue**

## Google form dropdown from sheet

Published in: Google Apps Script - Google Sheets - Google FormsA school tries to create a Google Form where students can register their data. The form would have a drop-down of countries, the names of class teachers as a multiple choice question, and a checkbox style question where students can select one or more subjects. It's easy to create such a form inside Google Forms - here's a sample form - but there are two questions: There's too much data to write. For example, the country alone has a list of 250 countries and it would take some work for the editor form to manually write each choice on the issue. The query selections in the form can be changed over time. In the previous example, some teachers may move out, new teachers can join, and the drop-down list in the form must be updated manually each time there is a change in staff. Auto-Fill questions in Google Forms with Google SheetsAs with everything else in the world of Google Forms and Google Sheets, we can easily automate the process of adding query selection in Google Forms in bulk using, you guessed it, Google Apps Script. The idea is simple. We will have a Google Sheet that will be the data source and have all the answer choices for different questions in Google Form. The app will read the data from this Google Sheet and automatically fill out the selections in the form with one click. You can even create a time trigger that runs hourly, daily, or monthly to dynamically update the form using the most up-to-date data in the spreadsheet. Add options to dropdown lists & multiple choice queriesCreate a Google spreadsheet and add the query titles to the first row of the sheet, one per column. Next, type down any options or selections that should be available per query. Here's what the spreadsheet structure would look like: The important thing to note here is that your column headings in the spreadsheet should match the exact form field labels in Google Shape. Google Script can add answers to multiple choice questions with a single answer, drop-down list, and multi-option check box. Mass add query selection with Google ScriptOpen Google Sheet which has query selection, go to the tools menu and select Script Editor. Replace the default code in the Script Editor with Google Script below. Please watch the video tutorial to know in more detail how this code works. const populateGoogleForms = () => { const GOOGLE\_SHEET\_NAME = '<Put the= name= of= sheet= here=>&gt;'; const GOOGLE\_FORM\_ID = '<Put your= google= form= id= here=>&gt;'; const ss = SpreadsheetApp.getActiveSpreadsheet(); const [header, ... data] = ss.getSheetByName(GOOGLE\_SHEET\_NAME).getRanges().getDisplayValues(); const selection = {}; header.forEach(title, i) => { choices[title] = data.map(d => d[i]).filter(e => e); }); FormApp.openById(GOOGLE\_FORM\_ID).getItems().map(item => ({ item, values: choices[item.getTitle()] })).filter(values => values)&lt;Put&gt; &lt;Put&gt;item, values }) => { switch (item.getType()) { case FormApp.ItemType.CHECKBOX: item.asCheckboxItem().setChoiceValues(values); break; case FormApp.ItemType.LIST\_ITEM: item.asListItem().setChoiceValues(values); break; case FormApp.ItemType.MULTIPLE\_CHOICE: item.asMultipleChoiceItem().setChoiceValues(values); break; default: } }); ss.toast('Google Form Updated!'); }; You need to replace GOOGLE\_SHEET\_NAME and GOOGLE\_FORM\_ID with your own values. Go to the Run menu inside the Scripts editor, select populateGoogleForms function and it should immediately select for all specified queries in Google Form. Going forward, whenever you need to update the answer selections in Google Forms, you can simply update the values inside the worksheet and run the same auto-populate function from the Scripts editor. Or, to make things even easier, you can add a button to the spreadsheet page that, when clicked, will automatically update the responses in Google Forms for you. In Google sheet, go to the Insert menu, choose drawing, and choose any shape. You can also add overlay text to the shape. When the shape is placed on the worksheet canvas, click the menu, select assign scripts, and type populateGoogleForms. That's it. You can click this button to update your Google Forms correctly within Google sheets. Do note that the script will not add selections, it will replace all existing selections with those found in your Google Sheet. Also see: Send Pre-Filled Google Forms via Email I looked through all the questions, and I can't really figure out what I'm doing wrong. Here's the background: I created a worksheet that has multiple tabs. Each tab corresponds to types of events (e.g. parades, festivals, sports). Each of these tabs is used to assign specific job numbers to a specific event. Parade job numbers are in the 2000 range, festivals 3000 range, sports 4000 range. I then have another tab that defines an array that collects all job numbers and events, and sorts them alphabetically. This tab is called Active\_Job\_Numbers. My goal is to create multiple forms that call this worksheet for the values on the Active\_Job\_Numbers tab. I have this to work, but not completely. Each form has a purpose such as sending a request to ledger to process a credit card transaction; another form to submit a request to cut a check. All these requests must be linked to an ever-changing list of job numbers. When I update the worksheet, the dropdown in the worksheet will populate based on Active\_Job\_Numbers data. The code I use works. BUT if I update a job number or event name on the spreadsheet, the form still sees the old value. I have triggers set up for onSubmit and onOpen and still, the only way I can get the dropdown in the form to update is if I change one of each labels. Then save it. Then open the form again. This is a showstopper. This the code I use: function updateForm() { // call your shape form connecting to the drop-down article was creditCardRequestGoogleForm = FormApp.openById(1LqtPLGEhocGdkg75mkldfplxSxWJmYZCp3u1bMwVQQ); var jobNumberDropDown = creditCardRequestGoogleForm.getItemById(1632374512).asListItem(); identify the sheet where data is needed to fill drop-down where ss = SpreadsheetApp.openById(179e9Jf4Aske85mfkwsphBQpZgtGhR8Q); SpreadsheetApp.setActiveSpreadsheet(ss); var activeJobNumbers = ss.getSheetByName('Active\_Job\_Numbers'); grab the values in the first column of the sheet - use 2 to skip header row where jobNumValues = activeJobNumbers.getRange(2, 1, activeJobNumbers.getMaxRows() - 1).getValues(); where jobNums = []; convert array ignore blank cells for(var i = 0; i < jobNumValues.length; i++) if(jobNumValues[i][0] != '') jobNums[i] = jobNumValues[i][0]; fill the drop-down value with the NumberDropDown.setChoiceValues(jobNums); } A separate but related problem... there are a few shapes I have that have two different drop-down elements that call to the same worksheet but to different tabs. I'm not sure how to group this script so I have to separate out so: function updateForm() { // call your form and connect to the drop-down article was ccrForm = FormApp.openById(1r5n6Sr7ftGCqht\_QHy9qWR1v2ESabxdE-bMxjR-M); was jobNumberDropDown = ccrForm.getItemById(167\_8955392).asListItem(); // identify the sheet where data is needed to fill in the drop-down where ss = SpreadsheetApp.openById(17eGAlcnBB7ejdfq3HBQBK9wJAKWYhQpZgtGhR8Q); SpreadsheetApp.setActiveSpreadsheet(ss); where currentJobNumbers = ss.getSheetByName('Active job numbers'); grab the values in the first column of the sheet - use 2 to skip header row where jobNumValues = currentJobNumbers.getRange(2, 1, currentJobNumbers.getMaxRows() - 1).getValues(); where jobNums = []; convert array ignore blank cells for(var i = 0; i < jobNumValues.length; i++) if(jobNumValues[i][0] != '') jobNums[i] = jobNumValues[i][0]; fill the drop-down value with the NumberDropDown.setChoiceValues(jobNums); } feature updateForm() { // This calls your form and connects to the drop-down object form was = FormApp.openById(1r5n6Sr7ftGCqht\_QHy9qWR1v2ESabxdE-bMxjR-M); was poNumberDropDown = form.getItemById(57865789991).asListItem(); // This is the second dropdown that calls the same worksheet but different tab NEEDS HELP COMBINING THESE TWO MORE ELEGANT WERE SS = SpreadsheetApp.openById(17eGAlcnBB7ejdfq3HBQBK9wJAKWYhQpZgtGhR8Q); SpreadsheetApp.setActiveSpreadsheet(ss); where currentPONumbers = ss.getSheetByName('PO\_Number'); grab the values in the first column of the sheet - use 2 to skip header row where poNumValues = currentPONumbers.getRange(2, 1, currentPONumbers.getMaxRows() - 1).getValues(); var poNums = []; convert array ignore blank cells for(var i = 0; i < poNumValues.length; i++) if(poNumValues[i][0] != '') poNums[i] = poNumValues[i][0]; drop-down drop-down with array data array data } Any insight would be greatly appreciated. Appreciated.