Server Code.gs

Backend modifications on form submits and Sheet interactions. Click on the Function Hyperlinks to be directed to more information regarding their process.

|  |  |  |
| --- | --- | --- |
| Function | Return type | Brief description |
| [a onSubmit(e)](#onSubmit) | **void** | Sets the specified range as the [active range](https://developers.google.com/apps-script/reference/spreadsheet/selection.html#getActiveRange()), with the top left cell in the range as the [current cell](https://developers.google.com/apps-script/reference/spreadsheet/selection.html#getCurrentCell()). |
| [a onOpen()](#onOpen) | [Menu](https://developers.google.com/apps-script/reference/base/menu.html) | Adds custom [menu items](https://developers.google.com/apps-script/guides/menus) to the Google Sheets toolbar |
| [a doGet()](#doGet) | [HtmlTemplate](https://developers.google.com/apps-script/reference/html/html-template.html) | Creates a new [HtmlTemplate](https://developers.google.com/apps-script/reference/html/html-template.html) object from a file in the code editor. |
| [a include(fileName)](#include) | [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) | Creates a new [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) object from a file in the code editor. |
| [a openApplication()](#openApplication) | **void** | Opens the HTML page Application.html |
| [a openLaundryApp()](#openLaundryApp) | **void** | Opens the HTML page Laundry.html |
| [a changeValueOnSubmit(e)](#changeValueOnSubmit) | **void** | Edits the form values for Image and Location. |
| [a setIDOnSubmit(e)](#setIDOnSUbmit) | **void** | Sets the unique identifier for the new range with a random number generator. |

onSubmit(e)

The onSubmit(e) trigger is set up in the custom trigger menu on the Google App Dashboard. It is set to run automatically when a user submits a Google Form connected to this Google Sheet.

Code

Server Code.gs

function onSubmit() {  
 setIDOnSubmit(e);

Logger.log(e);  
}

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| e | Object | The data from the form submission |

Return

None

onOpen()

The onOpen(e) trigger runs automatically when a user opens a spreadsheet, that they have permission to edit. It is used to add custom [menu items](https://developers.google.com/apps-script/guides/menus) to the Google Sheets toolbar.

Code

Server Code.gs

function onOpen() {  
  // Add a custom menu to the spreadsheet.  
  SpreadsheetApp .getUi()

.createMenu('App View')

.addItem('Get Link', 'openApplication')

.addItem('Laundry Mode', 'openLaundryApp')

.addToUi();  
}

Parameters

None

Return

[Menu](https://developers.google.com/apps-script/reference/base/menu) — This [Menu](https://developers.google.com/apps-script/reference/base/menu), for chaining.

doGet()

The [a oGet()](#doGet)trigger runs automatically when a user visits a [web app](https://developers.google.com/apps-script/guides/web). This fetches the HTML Code Application.html.

Code

Server Code.gs

function doGet() {  
  return HtmlService.createTemplateFromFile('Application').evaluate();  
}

Parameters

None

Return

[HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) — the new HtmlOutput object

include(*fileName*)

Creates a new [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) object from a file in the code editor.

Code

Server Code.gs

return HtmlService.createHtmlOutputFromFile(fileName);

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| filename | String | the name of the file to use |

Return

[HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) — the new HtmlOutput object

openApplication()

Creates a new [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) object from the Redirect.htmlfile. Will display as a modal inside of Google Sheets.

Code

Server Code.gs

var html = HtmlService.createHtmlOutputFromFile('Redirect');

SpreadsheetApp.getUi()

.showModalDialog(html, 'Go To Application');

Parameters

None

Return

[HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) — the new HtmlOutput object

openLaundryApp()

Creates a new [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) object the Laundry.htmlfile. Will display as a modal inside of Google Sheets.

Code

Server Code.gs

var html = HtmlService.createTemplateFromFile('Laundry').evaluate();

SpreadsheetApp.getUi()

.showModalDialog(html, 'Laundry Mode');

Parameters

None

Return

[HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) — the new HtmlOutput object

changeValueOnSubmit(*e*)

Creates a new [HtmlOutput](https://developers.google.com/apps-script/reference/html/html-output.html) object from a file in the code editor.

Code

Server Code.gs

function changeValueOnSubmit(e) {

var DataChange = e.namedValues["Item Picture"];

var DefaultLocation = e.namedValues["Where are you keeping this article of clothing?"];

Logger.log("Default Location: " + DefaultLocation);

Logger.log("Data Change: " + DataChange);

var ColD = ColumnID\_("Item Picture") +1;

var ColLoc = ColumnID\_("Default Location")+1;

DataChange = DataChange.toString().replace("open?", "uc?export=view&");

Logger.log("Data Change: " + DataChange);

e.source.getActiveSheet()

.getRange(e.range.rowStart, ColD)

.setValue(DataChange);

e.source.getActiveSheet()

.getRange(e.range.rowStart, ColLoc)

.setValue(DefaultLocation);

}

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| e | Object | The data from the form submission |

Return

None

setIDOnSubmit(*e*)

* I previously asked that the demo values be left in the table until the user has submitted at least one value for themselves. This is because the Auto ID generator relies on the last value in the database to populate.
* I fetch the last 2nd to last row because the last row is our new value and the 1st column for the ID value.
* I then randomly generate a number between 5 and 15 and add it to the last value. \*
* Finally, I place this value in the ID column of the last row.
* Next we call the *changeValueOnSubmit(e)* function.

Code

Server Code.gs

function setIDOnSubmit(e) {

var ss = SpreadsheetApp.getActiveSpreadsheet();

var sheet = ss.getSheetByName("Form Responses 1");

var range = sheet.getRange(1,1, sheet.getMaxRows());

var rowNum = range.getLastRow();

var LastID =range.getCell(rowNum-1, 1);

var CellValue = Number(LastID.getValue());

var ColA = 1;

var max =15; var min=5;

CellValue = CellValue+ Math.round((Math.random()\* (max - min) + min));

e.source.getActiveSheet()

.getRange(range.getLastRow(), ColA)

.setValue(CellValue);

changeValueOnSubmit(e);

}

Parameters

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| e | Object | The data from the form submission |

Return

None