

Google Apps Script Examples 7



Example 1: Create a new Google Slides presentation	1
Example 2: Send an email notification to the user	3
Example 3: Create a Google Drive file	4
Example 4: Get the current time	5
Example 5: Create Google Forms from Sheets	5
Example 6: Track Changes to Google Sheet	7
Example 7: Generate Slides from Sheets data	8

Example 1: Create a new Google Slides presentation

This code will create a new Google Slides presentation.

```
function createPresentation() {
```

```
// Create a new Google Slides presentation.
var presentation = SlidesApp.create('My
Presentation');

// Add a new slide to the presentation.
var slide = presentation.insertSlide();

// Add a title slide to the presentation.
slide.setTitle('My Presentation');

// Add a subtitle slide to the presentation.
slide = presentation.insertSlide();
slide.setSubtitle('This is a subtitle slide.');
```

```
// Add a new slide for each row of data in the sheet.
for (var i = 0; i < data.length; i++) {
  slide = presentation.insertSlide();

  // Add a title to the slide.
  slide.setTitle(data[i][0]);

  // Add a paragraph to the slide for each column of
data in the row.
  for (var j = 1; j < data[i].length; j++) {
    slide.appendParagraph(data[i][j]);
  }
}

// Save the presentation.
presentation.save();
}
```

This code works by first creating a new Google Slides presentation object. Then, it adds a new slide to the presentation, adds a title slide to the presentation, adds a subtitle slide to the presentation, adds a new slide for each row of data in the sheet, adds a title to each slide, and adds a paragraph to each slide for each column of data in the row. Finally, it saves the presentation.

This code could be used to create a presentation from data that is stored in a Google Sheet. For example, you could use this code to create a presentation from data that stores the names and contact information of your customers.

Example 2: Send an email notification to the user

This code will send an email notification to the user.

```
function sendEmail() {
  // Get the user's email address.
  var userEmail = Utilities.getUserEmail();

  // Create an email message.
  var message = GmailApp.createEmail();
  message.setSubject('Your email notification');
  message.setBody('This is the body of your email notification.');
```



```
  // Send the email notification.
  message.sendTo(userEmail);
}
```

This code works by first getting the user's email address. Then, it creates an email message and sets the subject line and body of the message. Finally, it sends the email notification.

This code could be used to send email notifications to users about important events or updates. For example, you could use this code to send email notifications to users when their account is created or when their password is changed.

Example 3: Create a Google Drive file

This code will create a new Google Drive file.

```
function createFile() {  
  // Get the file name.  
  var fileName = 'My File.txt';  
  
  // Get the content of the file.  
  var content = 'This is the content of my file.';  
  
  // Create a new Google Drive file.  
  var file = DriveApp.createFile(fileName);  
  
  // Set the content of the file.  
  file.setContent(content);  
  
  // Save the file.  
  file.save();  
}
```

This code works by first getting the file name and then getting the content of the file. Once it has these two pieces of information, it creates a new Google Drive file, sets the content of the file, and then saves the file.

This code could be used to create a file that users can download. For example, you could use this code to create a file that users can download that contains a list of your products or services.

Example 4: Get the current time

This code will get the current time.

```
function getCurrentTime() {  
    // Get the current time.  
    var currentTime = Utilities.now();  
  
    // Display the current time.  
    Logger.log('The current time is: ' + currentTime);  
}
```

This code works by first getting the current time using the `Utilities.now()` method. Then, it displays the current time using the `

Example 5: Create Google Forms from Sheets

Automatically Create Google Forms from Spreadsheet Data: This code automatically creates Google Forms from data in a specified spreadsheet. It first retrieves the data from the spreadsheet and

loops through it. For each row of data, it creates a new form with the specified title and questions.

```
function createFormsFromSpreadsheet() {
  var ss = SpreadsheetApp.openById('spreadsheet_id');
  // specify spreadsheet ID
  var sheet = ss.getSheetByName('Sheet1'); // specify
sheet name
  var data = sheet.getDataRange().getValues();

  for (var i = 1; i < data.length; i++) {
    var row = data[i];
    var form = FormApp.create(row[0]); // specify form
title
    form.addTextItem().setTitle(row[1]); // specify
first question

    form.addMultipleChoiceItem().setTitle(row[2]).setChoice
s([row[3], row[4], row[5]]); // specify second question
and choices
  }
}
```

The SpreadsheetApp service is used to access the specified spreadsheet and retrieve the data from the specified sheet. The for loop iterates through each row of data (starting from the second row since the first row is assumed to contain column headers). For each row, a new form is created with the specified title. Two questions are added to the form: a text question with the title specified in the second column of the row, and a multiple choice question with the title specified in the third column of the row and the choices specified in columns 4-6.

Example 6: Track Changes to Google Sheet

Track Changes to Google Sheet Data: This code tracks changes to data in a specified sheet and sends an email notification when a change occurs. It first creates a copy of the specified sheet and sets up a trigger to run the trackChanges function whenever the sheet is edited. When the function is run, it retrieves the previous and current values of the edited cell(s) and sends an email with the old and new values.

```
function trackChanges() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheetByName('Sheet1'); // specify
sheet name
  var range = sheet.getActiveRange();
  var oldValue = range.getDisplayValue();
  var newValue = e.value;

  if (oldValue != newValue) {
    var recipient = 'email@example.com'; // specify
recipient email address
    var subject = 'Change to Sheet1';
    var body = 'Old Value: ' + oldValue + '\nNew Value:
' + newValue;

    MailApp.sendEmail(recipient, subject, body);
  }
}

function createTrigger() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
```

```

    var sheet = ss.getSheetByName('Sheet1'); // specify
sheet name
    var copy = sheet.copyTo(ss).setName('Sheet1 Copy');

ScriptApp.newTrigger('trackChanges').forSpreadsheet(ss)
.onEdit().create();
}

```

The SpreadsheetApp service is used to access the specified spreadsheet and sheet. The trackChanges function is set up to run whenever the sheet is edited. It retrieves the previous and current values of the edited cell(s) using the getActiveRange() method and compares them. If they are different, an email is sent with the old and new values. The createTrigger function creates a copy of the sheet and sets up a trigger to run the trackChanges function whenever the sheet is edited.

Example 7: Generate Slides from Sheets data

Automatically Generate Google Slides from Google Sheets Data: This code automatically generates Google Slides from data in a specified sheet. It first retrieves the data from the sheet and loops through it. For each row of data, a new slide is created with the specified title and content.

```

function generateSlidesFromSheet() {
    var ss = SpreadsheetApp.openById('spreadsheet_id');
// specify spreadsheet ID
    var sheet = ss.getSheetByName('Sheet1'); // specify
sheet name
    var data = sheet.getDataRange().getValues();
}

```

```
var templateId = 'template_id'; // specify slide
template ID
var template = SlidesApp.openById(templateId);
var destinationId = 'destination_id'; // specify
destination folder ID

for (var i = 1; i < data.length; i++) {
  var row = data[i];
  var title = row[0];
  var content = row[1];
  var slide = template.duplicate().setName(title);
  var shapes = slide.getShapes();
  var text = shapes[0].getText();
  text.setText(content);
  var slideBlob = slide.getBlob();
  var folder = DriveApp.getFolderById(destinationId);
  folder.createFile(slideBlob);
}
}
```