

```
function postArticleToShopify() {  
    // Get the active document  
    var doc = DocumentApp.getActiveDocument();  
    // Get the document title  
    var title = doc.getName();  
    // Get the body of the document and convert it to HTML  
    var body = doc.getBody();  
    var bodyHtml = convertBodyToHTML(body);  
    // Get the current date and time in the required format  
    var publishedAt = formatCurrentDate();  
    // Create the payload for the POST request  
    var payload = {  
        "article": {  
            "title": title,  
            "author": "Armando Roggio",  
            "tags": "Test",  
            "body_html": bodyHtml,  
            "published_at": publishedAt  
        }  
    };  
    // Convert the payload to JSON  
    var jsonData = JSON.stringify(payload);  
    // Define the headers for the POST request  
    var headers = {
```

```
"X-Shopify-Access-Token": [ACCESS TOKEN GOES HERE],
"Content-Type": "application/json"
};

// Define the POST request options
var options = {
  "method": "POST",
  "headers": headers,
  "payload": jsonData,
  "muteHttpExceptions": true// To prevent stopping on HTTP errors
};

// Send the POST request
var url = "https://[SHOPIFY_URL].myshopify.com/admin/api/2024-07/blogs/[BLOG ID]/articles.json";

var response = UrlFetchApp.fetch(url, options);

// Log the response for debugging
Logger.log(response.getContentText());
}

// Helper function to format the current date to the required format
function formatCurrentDate() {
  var date = new Date();

  var options = { weekday: 'short', year: 'numeric', month: 'short', day: 'numeric', hour: '2-digit', minute: '2-digit', second: '2-digit', timeZone: 'UTC', timeZoneName: 'short' };

  return date.toLocaleString('en-US', options).replace(", ", ""); // e.g., "Thu Sep 04 15:45:47 UTC 2024"
}
```

```
// Helper function to convert the document body to HTML
function convertBodyToHTML(body) {
    var html = "";
    var numChildren = body.getNumChildren();
    // Loop through all elements in the document body
    for (var i = 0; i < numChildren; i++) {
        var child = body.getChild(i);
        // Handle paragraphs
        if (child.getType() == DocumentApp.ElementType.PARAGRAPH) {
            var paragraph = child.asParagraph();
            var paragraphText = "";
            var numParagraphElements = paragraph.getNumChildren();
            // Loop through the children of the paragraph (to check for images, text, links, etc.)
            for (var j = 0; j < numParagraphElements; j++) {
                var paragraphChild = paragraph.getChild(j);
                // Handle inline images inside the paragraph
                if (paragraphChild.getType() == DocumentApp.ElementType.INLINE_IMAGE) {
                    var image = paragraphChild.asInlineImage();
                    var blob = image.getBlob();
                    var mimeType = blob.getContentType();
                    var base64 = Utilities.base64Encode(blob.getBytes());
                    paragraphText += '';
                }
                // Handle text and rich links
            }
        }
    }
}
```

```

} else if (paragraphChild.getType() == DocumentApp.ElementType.TEXT) {
    var textElement = paragraphChild.asText();
    var textContent = textElement.getText();
    // Check for rich links (hyperlinks)
    for (var k = 0; k < textElement.getTextAttributeIndices().length; k++) {
        var url = textElement.getLinkUrl(k);
        if (url && (url.includes("youtube.com") || url.includes("youtu.be"))) {
            // Extract the video ID from the URL
            var videoid = extractYouTubeVideoid(url);
            // If it's a valid YouTube link, generate the embed code
            if (videoid) {
                var embedHTML = generateYouTubeEmbed(videoid);
                paragraphText = embedHTML;
            }
        } else {
            // Append normal text
            paragraphText += escapeHTML(textElement.getText());
        }
    }
}

// Check if the paragraph is a heading or normal text
if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING1) {
    html += '<h1>' + paragraphText + '</h1>';
}

```

```

} else if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING2) {
html += '<h2>' + paragraphText + '</h2>';
} else if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING3) {
html += '<h3>' + paragraphText + '</h3>';
} else if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING4) {
html += '<h4>' + paragraphText + '</h4>';
} else if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING5) {
html += '<h5>' + paragraphText + '</h5>';
} else if (paragraph.getHeading() == DocumentApp.ParagraphHeading.HEADING6) {
html += '<h6>' + paragraphText + '</h6>';
} else {
// Normal paragraph
html += '<p>' + paragraphText + '</p>';
}
// Handle list items (bulleted and numbered)
} else if (child.getType() == DocumentApp.ElementType.LIST_ITEM) {
var listItem = child.asListItem();
var tag = listItem.getGlyphType() == DocumentApp.GlyphType.BULLET ? 'ul' : 'ol';
html += '<' + tag + '><li>' + escapeHTML(listItem.getText()) + '</li></' + tag + '>';
// Handle tables
} else if (child.getType() == DocumentApp.ElementType.TABLE) {
html += '<table border="1">';
var table = child.asTable();

```

```

for (var row = 0; row < table.getNumRows(); row++) {
html += '<tr>';
for (var col = 0; col < table.getRow(row).getNumCells(); col++) {
var cellText = table.getRow(row).getCell(col).getText();
html += '<td>' + escapeHTML(cellText) + '</td>';
}
html += '</tr>';
}
html += '</table>';
}
}
returnhtml;
}

// Helper function to extract YouTube video ID from URL
function extractYouTubeVideoid(url) {
var videoid = null;

// Check for standard YouTube URL
var youtubeMatch = url.match(/(?:https?:\W)?(?:www\.)?youtube\.com\/watch?v=([^\&]+)/);

// Check for YouTube short link
var youtuMatch = url.match(/(?:https?:\W)?youtu\.beV([^\?]+)/);

if (youtubeMatch) {
videoid = youtubeMatch[1];
} else if (youtuMatch) {

```

```
videoid = youtuMatch[1];
}
returnvideoid;
}

// Helper function to generate responsive YouTube embed HTML
function generateYouTubeEmbed(videoid) {
return`
<div style="position: relative; padding-bottom: 56.25%; height: 0; overflow: hidden; max-
width: 100%; height: auto;">

<iframe style="position: absolute; top: 0; left: 0; width: 100%; height: 100%;"
src="https://www.youtube.com/embed/${videoid}?rel=0"
frameborder="0" allow="autoplay; encrypted-media" allowfullscreen>

</iframe>

</div>`;
}

// Helper function to escape HTML entities
function escapeHTML(text) {
returntext.replace(/&/g, '&amp;')
.replace(/</g, '&lt;')
.replace(/>/g, '&gt;')
.replace(/"/g, '&quot;')
.replace(/'/g, '&#39;');
}
```