

```

        } ()function myFunction
        } ()function loginToWiki
var wikiApiUrl = "http://example.com/mediawiki/api.php"; // Replace with your wiki's API
URL
var username = "yourUsername"; // Replace with your username
var password = "yourPassword"; // Replace with your password

Step 1: Get login token //
    } = var params
    , "action: "login
    ,lgname: username
    ,lgpassword: password
    "format: "json
    ;{

    } = var options
    , "method: "post
    muteHttpExceptions: true
    ;{

var response = UrlFetchApp.fetch(wikiApiUrl + "?" + new URLSearchParams(params),
    ;(options
    ;(var data = JSON.parse(response.getContentText

    } ("if (data.login.result === "Success
    ;("Logger.log("Logged in successfully
    } else {
    ;(Logger.log("Login failed: " + data.login.result
    {

    ;return data.login.token
    {

    } ()function syncGoogleDocsToMediaWiki
var folderId = "yourFolderId"; // Replace with your Google Drive folder ID
var wikiApiUrl = "http://example.com/mediawiki/api.php"; // Replace with your wiki's API
URL
var token = loginToWiki(); // Get login token

;(var folder = DriveApp.getFolderById(folderId
    ;(var files = folder.GetFiles

    } ()while (files.hasNext

```

```

                                ;()var file = files.next
    } ("if (file.getMimeType() === "application/vnd.google-apps.document
        ;()var doc = DocumentApp.openById(file.getId
        ;()var content = doc.getBody().getText
    var pageTitle = file.getName(); // Page title will be the same as the file name

```

```

                                Convert content to wiki markup //
    ;(var wikiContent = convertToWikiMarkup(content)

```

```

    Step 2: Get CSRF token for editing //

```

```

        } = var params
        ,"action: "query
        ,"meta: "tokens
        ,"type: "csrf
        "format: "json
        ;{

```

```

        } = var options
        ,"method: "get
        muteHttpExceptions: true
        ;{

```

```

    var response = UrlFetchApp.fetch(wikiApiUrl + "?" + new
        ;(URLSearchParams(params), options
    ;()var data = JSON.parse(response.getContentText
        ;var csrfToken = data.query.tokens.csrfToken
    Logger.log("CSRF Token: " + csrfToken); // Log the CSRF token

```

```

    Step 3: Check if the page exists, if not create it //

```

```

        } = var queryParams
        ,"action: "query
        ,titles: pageTitle
        "format: "json
        ;{

```

```

        } = var queryOptions
        ,"method: "get
        muteHttpExceptions: true
        ;{

```

```

    var queryResponse = UrlFetchApp.fetch(wikiApiUrl + "?" + new
        ;(URLSearchParams(queryParams), queryOptions
    ;()var queryData = JSON.parse(queryResponse.getContentText

```

```

Logger.log("Query Response: " + JSON.stringify(queryData)); // Log the query
                                                                    response
;var pageExists = queryData.query.pages

    If the page does not exist, create it //
        } (["if (pageExists["-1
    Step 4: Create a new page //
        } = var createParams
            , "action: "edit
            , title: pageTitle
            , text: wikiContent
            , token: csrfToken
            "format: "json
            ;{

        } = var createOptions
            , "method: "post
            , "contentType: "application/x-www-form-urlencoded
            , payload: createParams
            muteHttpExceptions: true
            ;{

;(var createResponse = UrlFetchApp.fetch(wikiApiUrl, createOptions
    ;(Logger.log("Created new page: " + pageTitle
Logger.log("Create Response: " + createResponse.getContentText()); // Log the
                                                                    API response
    } else {
        If the page exists, update it //
        } = var editParams
            , "action: "edit
            , title: pageTitle
            , text: wikiContent
            , token: csrfToken
            "format: "json
            ;{

        } = var editOptions
            , "method: "post
            , "contentType: "application/x-www-form-urlencoded
            , payload: editParams
            muteHttpExceptions: true
            ;{

;(var editResponse = UrlFetchApp.fetch(wikiApiUrl, editOptions

```

```

;(Logger.log("Updated page: " + pageTitle
Logger.log("Edit Response: " + editResponse.getContentText()); // Log the API

response
{
    {
        {
            {

Function to convert text to wiki markup //
} (function convertToWikiMarkup(content
Convert headings //
== content = content.replace(/^# (.+)$/gm, "== $1 =="); // Level 1 heading -> == Heading
content = content.replace(/^## (.+)$/gm, "=== $1 ==="); // Level 2 heading -> === Heading
===
content = content.replace(/^### (.+)$/gm, "==== $1 ====="); // Level 3 heading -> =====
===== Heading

Convert paragraphs //
content = content.replace(/\n/g, "\n\n"); // New paragraph separated by a blank line

Convert lists //
content = content.replace(/^\. (.+)$/gm, "* $1"); // Unordered list
content = content.replace(/^d+\. (.+)$/gm, "# $1"); // Ordered list

Convert links //
content = content.replace(/\[([^\]]+)\]\([([^\)]+)\)/g, "[Description](URL) -> [URL
[Description]

(Convert tables (simple //
content = content.replace(/\| (.+)/g, "| $1"); // Table

;return content
{
    {

```