# How the link library code works:

## Google Sheet with links: https://docs.google.com/spreadsheets/d/10pUk3trlJApXoVZuwnkJKAQS3f36UU0Y43SBTRB1IUI/edit#gid=364644472

## The Api Call:

handleClientLoad -> initClient -> makeApiCall

First, the html calls the handleClientLoad function, which then calls the initClient function, which then calls the makeApiCall function. All of these functions were provided by google and should not need to be modified except for updating variables.

In the initClient function it has the api key and client id, which should be updated for the move.

The makeApiCall function has the spreadsheetId to tell the code what google sheet to pull.

## Processing the response:

processTheResponse -> processTheSheet -> processTheRow

So the response from the api call is just a big json object, so we need to process it.

**processTheResponse**

processTheResponse just loops through each sheet and calls it to be processed.

It also removes the loading text on the website when all sheets have been processed.

**processTheSheet**

The code grabs all the rows on the sheet and stores them in an array.

The first row is processed in the first for loop. First it grabs the first value in the first row and sets it as the title, which will be the name on the button on the html page. It then loops through every other cell in the first row. If it has an = in the cell, it checks if the cell has the word “target” in it. If it does then it grabs the information on the right hand side of the = and sets the target variable to have its value (example: it becomes “target=’\_blank’” if \_blank is the given value)

The second for loop goes through every other row and calls the processTheRow function on it.

Once it has all the html from processing every row, it adds a button to the top, sets it to display the html that was made from all the processed rows when clicked.

**processTheRow**

This function formats a given row to fit in a dictionary object where the key is the name (example: “Invisalign”) and the value is the url (example: /dear\_doctor/invisalign).

It then makes html with the information that have onclick events and returns that to the processTheSheet function to add it to the html page when the entire sheet is processed.

## Html Functions

These are functions that are called from buttons on the html page. All buttons are made in the processing of the sheets.

**copylink**

This is a function added to all links to call when they are right clicked. It makes a hidden text area, adds the link to the text area, highlights it, then runs the “copy” command so it puts the text in the clipboard.

It also removes then adds the “animation” class to the button you click so it does that red flash.

**generateLinks**

This function is called when the “generate” button is clicked. It checks which buttons are checked, makes a link for each one that is checked, then outputs it to the output div.

It also scrolls down to the output.

**selectAll**

Called by the select all button. Marks every link as checked. Also sets them to be highlighted

**selectNone**

Called by the select none button. Marks every link as not checked. Also removes the highlight from them.

**copyText**

Called by the copy text button. Selects everything in the output and copies it to the clipboard.

**displayCheckboxes**

This function is called by the library buttons at the top of the page. Creates the “generate”, “select all”, “select none” buttons, search bar, and link buttons on the page. The code for the link buttons is made during the process sheets function and is just placed on the page when this function is called.

**searchCheckboxes**

Called by the search bar. Grabs the text content in the search bar, removes spaces and makes it lowercase. Compares the text to the titles in the checkboxes and sets them to display or not display if they match or not.

**highlight**

Called when a link is clicked. Toggles the active class to add or remove the highlight.

## Helper Functions

These are functions that are used by other functions.

**getTarget**

Takes in a cell with target= in it and returns the code to add to the link

**getPrepend**

Takes in a cell with = and doesn’t have a target. Makes a radio button to change the /articles/ to something else.

**cleanText**

Gets just the text of a link button.

**makeLink**

Makes a link with given text and url.

**makeTargetLink**

Makes a link with a target defined with given text and url

**addHighlight**

Adds the active class to highlight the element

**removeHighlight**

Removes the active class to remove the highlight