

Google Sheet Data as JSON

The Google Visualization API Query Language lets you perform various data manipulations with the query to the data source.

https://developers.google.com/chart/interactive/docs/querylanguage#top_of_page

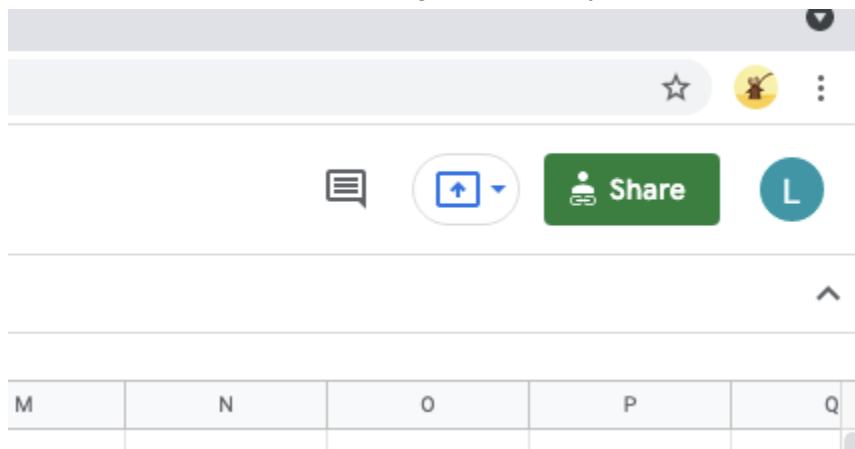
- gviz is short for Google Visualization API
- Google Visualization API Query Language is similar to SQL

Spreadsheet Setup and Shares

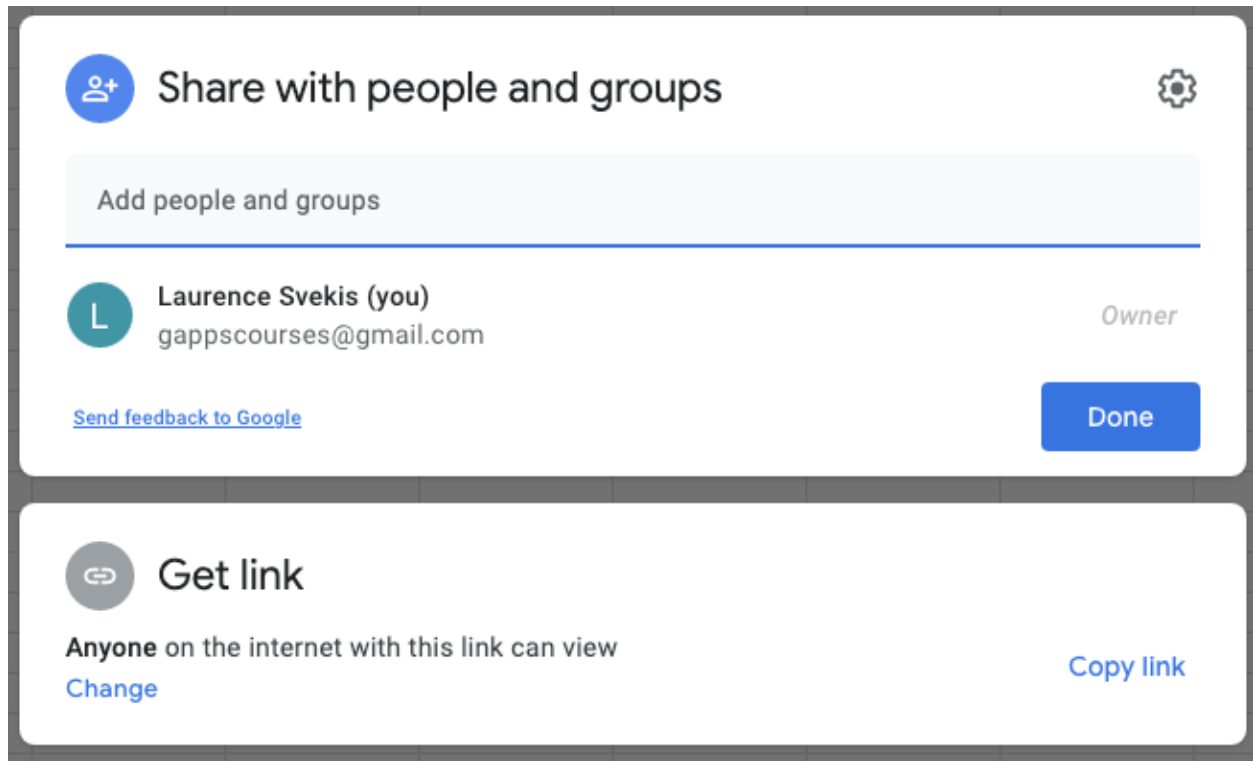
Setting up a share of your spreadsheet

Create a spreadsheet in your Google Drive. Add some dummy values in the spreadsheet, using numeric values.

Select Share button in the top right corner of your spreadsheet



In the Popup window select the change link to Anyone on the internet with this link can view.



Copy your sheet URL

docs.google.com/spreadsheets/d/1EgnjzKuodX9tdwUqtW92WSdIlIBZvTnrgf9cXgGGL03Y/edit#gid=0

ed Sheet ☆ 📁 ☁

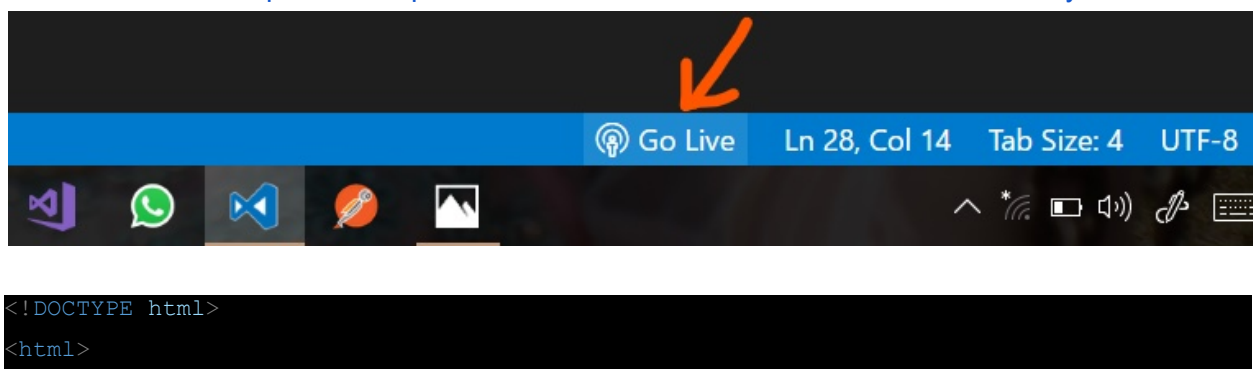
You are ready to share your Sheet as an endpoint!

Using JavaScript Fetch to connect to the sheet Data

Create an index.html page and link to a JavaScript file. Suggested editor is Visual Studio Code and using Live Server to output the page as HTTP page in the Browser

Get Visual Studio Code - <https://visualstudio.microsoft.com/>

Get Live Server - <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>



```

<head>
  <title>Share your Google Sheets</title>
  <style>
    .box {
      display: inline-block;
      width: 25%;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="output"></div>
</div>
  <script src="apps1.js"></script>
</body>
</html>

```

```

//https://docs.google.com/spreadsheets/d/1EgnjzKuodX9tdwUqtW92WSdlIBZvTnrgf9cXgGGL03Y/
edit?usp=sharing
const output = document.querySelector('.output');
const url = 'https://docs.google.com/spreadsheets/d/';
const ssid = '1EgnjzKuodX9tdwUqtW92WSdlIBZvTnrgf9cXgGGL03Y';
const query1 = `/gviz/tq?`;
const endpoint1 = `${url}${ssid}${query1}`;
fetch(endpoint1)
  .then(res => res.text())
  .then(data => {
    const temp = data.substr(47).slice(0, -2);
    const json = JSON.parse(temp);
    const rows = json.table.rows;
    rows.forEach((row) => {
      const div = document.createElement('div');
      const templ = row.c;
      templ.forEach((cell) => {
        const box = document.createElement('div');
        box.textContent = cell.v;
        box.classList.add('box');
        div.append(box);
      })
      output.append(div);
    })
  })
})

```

Get Sheet data as JSON with Headings from sheets

First row of data automatically becomes headings if they contain string values. You can now loop through the first row cols and get the heading values for each one of the rows. Get your Content in CSV format and specify the format using the txout value. csv or json

```
'txout=json';
```

```
'txout=csv';
```

Select you sheet using the sheet value.

```
sheet=Sheet6
```

Source code below for HTML and JavaScript

```
<!DOCTYPE html>
<html>
<head>
  <title>Share your Google Sheets</title>
  <style>
    .box {
      display: inline-block;
      width: 20%;
      text-align: center;
    }
    .heading{
      font-size:1.2em;
      color:white;
      background-color:black;
    }
    .row{

    }
  </style>
</head>
<body>
  <button>Request</button>
  <div class="output"></div>
</div>
```

```
<script src="apps2.js"></script>
</body>
</html>
```

```
const output = document.querySelector('.output');
const btn = document.querySelector('button');
const url = 'https://docs.google.com/spreadsheets/d/';
const ssid = '1EgnjzKuodX9tdwUqtW92WSdlIBZvTnrgf9cXgGGL03Y';
const q1 = '/gviz/tq?';
const q2 = 'tqx=out:json';
const q3 = 'sheet=Sheet6';

btn.addEventListener('click',getData);

function getData(){
  let url1 = `${url}${ssid}${q1}&${q2}&${q3}`;
  output.innerHTML = '';
  fetch(url1)
    .then(res => res.text())
    .then(data => {
      const json = JSON.parse(data.substr(47).slice(0,-2));
      console.log(json.table);
      const headings = makeCell(output, '', 'heading');
      json.table.cols.forEach((col)=>{
        const el = makeCell(headings,col.label,'box');
      })
      json.table.rows.forEach((row)=>{
        //console.log(row);
        const div = makeCell(output, '', 'row');
        row.c.forEach((cell)=>{
          const ele1 = makeCell(div,`${cell.v}`, 'box');
        })
      })
    })
}

function makeCell(parent,html,classAdd){
  const ele = document.createElement('div');
  parent.append(ele);
  ele.innerHTML = html;
  ele.classList.add(classAdd);
}
```

```
    return ele;
}
```

Using Query Language Select Custom Sheet Data as JSON

With Google Sheets you can make custom query requests to the Spreadsheet to return specific results. MakeSQL like queries to your spreadsheet data. Google Sheets can select the sheet and the response format like JSON or CSV. Perfect to use Google Sheets data directly on your Website.

```
const output = document.querySelector('.output');
const btn = document.querySelector('button');
const url = 'https://docs.google.com/spreadsheets/d/';
const ssid = '1EgnjzKuodX9tdwUqtW92WSdlIBZvTnrgf9cXgGGL03Y';
const q1 = '/gviz/tq?';
const q2 = 'tqx=out:json';
const q3 = 'sheet=Sheet6';
//const query1 = 'select A,B where C > 100 ';
const query1 = `select A,B,C,E where E contains 'a' limit 3`;
const q4 = encodeURIComponent(query1);
btn.addEventListener('click',getData);
function getData(){
    let url1 = `${url}${ssid}${q1}&${q2}&${q3}&tq=${q4}`;
    output.innerHTML = url1;
    fetch(url1)
    .then(res => res.text())
    .then(data => {
        const json = JSON.parse(data.substr(47).slice(0,-2));
        console.log(json.table);
        const headings = makeCell(output, '', 'heading');
        json.table.cols.forEach((col)=>{
            const el = makeCell(headings,col.id,'box');
        })
        json.table.rows.forEach((row)=>{
            //console.log(row);
            const div = makeCell(output, '', 'row');
            row.c.forEach((cell)=>{
                const ele1 = makeCell(div,`${cell.v}`, 'box');
            })
        })
    })
}
```

```

    })

    })
}

function makeCell(parent,html,classAdd) {
    const ele = document.createElement('div');
    parent.append(ele);
    ele.innerHTML = html;
    ele.classList.add(classAdd);
    return ele;
}

```

Create a Dynamic Web Application to dynamically connect to different sheets

Exercise to make connections to your Google Spreadsheet with Dynamic value from an input field. Set the Spreadsheet ID from the input field, select the sheet you want to return data from and add a query value that gets encoded. Return the results from the interactive web input fields that are specific to that request. Make more request build your endpoint URL.

```

<!DOCTYPE html>
<html>
<head>
    <title>Share your Google Sheets</title>
    <style>
        .box {
            display: inline-block;
            text-align: center;
        }
        .heading{
            font-size:1.2em;
            color:white;
            background-color:black;
        }
        .row{

```

```

    }

    .mySheet, .myQuery{
        width:100%;
    }
</style>
</head>
<body>
    <h1>Google Sheets as JSON data for your Website</h1>
    <div>
        <p>Google Sheets can be used as a source for data to your webpages. New updated
API can now use Database like
        Select options to output the content.
    </p>
    <label for="mySheet">Sheet ID:</label>
    <input type="text" class="mySheet" name="mySheet"
value="1EgnjzKuodX9tdwUqtW92WSdlIBZvTnrgf9cXgGGL03Y">
    <label for="sheets">Select a Sheet:</label>
    <select name="sheets" class="sheets"><br>

    </select><br>
    <label for="myQuery">Add a Query:</label>
    <input type="text" class="myQuery" name="myQuery">
    <br>
    <button>Request</button>
    <div class="output"></div>
    </div>
    <script src="apps4.js"></script>
</body>
</html>

```

```

const output = document.querySelector('.output');
const btn = document.querySelector('button');
const myQuery = document.querySelector('.myQuery');
const mySheet = document.querySelector('.mySheet');
const sheets = document.querySelector('.sheets');
const sheetArr = ['test1', 'Sheet2', 'Sheet6', 'people'];
sheetArr.forEach((sheet) => {
    const tempEle = makeCell(sheets, sheet, 'ele', 'option');
    tempEle.value = sheet;
})
const url = 'https://docs.google.com/spreadsheets/d/';
const q1 = '/gviz/tq?';

```



```

const q2 = 'tqx=out:json';
//const query1 = 'select A,B where C > 100 ';
//const query1 = `select A,B,C,E where E contains 'a' limit 3`;
//const query1 = `select A,B`;
//const q4 = encodeURIComponent(query1);
btn.addEventListener('click', getData);

function getData() {
  const ssid = mySheet.value;
  const q3 = `sheet=${sheets.value}`;
  let url1 = `${url}${ssid}${q1}&${q2}&${q3}`;
  if (myQuery.value.length > 0) {
    const updateCode = encodeURIComponent(myQuery.value);
    url1 += `&tq=${updateCode}`;
  }
  output.innerHTML = url1;
  fetch(url1)
    .then(res => res.text())
    .then(data => {

      const json = JSON.parse(data.substr(47).slice(0, -2));
      //console.log(json);
      const headings = makeCell(output, '', 'heading');
      let wid = 100 / json.table.cols.length;
      json.table.cols.forEach((col) => {
        console.log(col);
        const val = col.label || col.id;
        const el = makeCell(headings, val, 'box');
        el.style.width = wid + '%';
      })
      json.table.rows.forEach((row) => {
        //console.log(row);
        const div = makeCell(output, '', 'row');
        row.c.forEach((cell) => {
          const ele1 = makeCell(div, `${cell.v}`, 'box');
          ele1.style.width = wid + '%';
        })
      })
    })
}

```

```
function makeCell(parent, html, classAdd, eleType = 'div') {  
  const ele = document.createElement(eleType);  
  parent.append(ele);  
  ele.innerHTML = html;  
  ele.classList.add(classAdd);  
  return ele;  
}
```