

# GWS Apps Script demo scripts

## 1. Appsscript.json

```
{
  "timeZone": "Asia/Taipei",
  "exceptionLogging": "STACKDRIVER",
  "runtimeVersion": "V8",
  "dependencies": {
  "enabledAdvancedServices": [
    {
      "userSymbol": "Calendar",
      "version": "v3",
      "serviceId": "calendar"
    }
  ]
},
  "oauthScopes": [
    "https://www.googleapis.com/auth/calendar.readonly",
    "https://mail.google.com/",
    "https://www.googleapis.com/auth/userinfo.email",
    "https://www.googleapis.com/auth/calendar",
    "https://www.googleapis.com/auth/drive",
    "https://www.googleapis.com/auth/spreadsheets",
    "https://www.googleapis.com/auth/sqlservice",
    "https://www.googleapis.com/auth/script.external_request"
  ],
  "webapp": {
    "executeAs": "USER_DEPLOYING",
    "access": "ANYONE_ANONYMOUS"
  }
}
```

## 2. Rendering the Google Sheet data on Website

### 2.1 網頁 Script 檔

```
function doGet() {  
  return HtmlService  
    .createTemplateFromFile('Index')  
    .evaluate();  
}  
  
function getData() {  
  const data = SpreadsheetApp  
    .openByUrl('https://docs.google.com/spreadsheets/d/1TF0QAZYCdRsAjnNYY  
6nRzKpi5siFU-AS-ziItUkOSTk/edit#gid=0')  
    .getActiveSheet()  
    .getDataRange()  
    .getValues();  
  console.log(data);  
  return data  
}
```

### 2.2 網頁 HTML 檔案

```
<!doctype html>  
<html lang="en">  
  <head>  
    <!-- Required meta tags -->  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1, shrink-to-fit=no">  
  
    <!-- Bootstrap CSS -->  
    <link rel="stylesheet"  
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap  
.min.css"
```

```
integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x
9JvoRxT2MZw1T" crossorigin="anonymous">
```

```
<title>Hello, world!</title>
```

```
</head>
```

```
<body>
```

```
<? const option2 = SpreadsheetApp
```

```
.openByUrl('https://docs.google.com/spreadsheets/d/1TF0QAZYCdRsAjnNYY
6nRzKpi5siFU-AS-ziItUk0STk/edit#gid=0')
```

```
.getActiveSheet()
```

```
.getDataRange()
```

```
.getValues(); ?>
```

```
<? var data = getData(); ?>
```

```
<table class="table table-bordered table-dark">
```

```
<thead class="thead-dark">
```

```
<? for (var i = 0; i < 1; i++) { ?>
```

```
<tr>
```

```
<? for (var j = 0; j < data[i].length; j++) { ?>
```

```
<th scope="col"><?= data[i][j] ?></th>
```

```
<? } ?>
```

```
</tr>
```

```
<? } ?>
```

```
</thead>
```

```
<tbody>
```

```
<? for (var i = 1; i < data.length; i++) { ?>
```

```
<tr>
```

```
<? for (var j = 0; j < data[i].length; j++) { ?>
```

```
<td><?= data[i][j] ?></td>
```

```
<? } ?>
```

```
</tr>
```

```
<? } ?>
```

```
</tbody>
```

```
</table>
```

```

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.mi
n.js"
integrity="sha384-U02eT0CpHqdSJKQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4
sF86dIHNDz0W1" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.m
in.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDs
4x0xIM+B07jRM" crossorigin="anonymous"></script>
</body>
</html>

```

### 3. MySQL(JDBC)

#### 3.1 輸入資料庫設定檔

Apps Script => 設定 => Property:

dbUrl : jdbc:mysql://{host}:{port}/{db\_name}

password : 20230915

root : root

```
const scriptProperties = PropertiesService.getScriptProperties();
```

```
const dbUrl = scriptProperties.getProperty("dbUrl");
```

```
const root = scriptProperties.getProperty("root");
```

```
const rootPwd = scriptProperties.getProperty("password");
```

```
function checkProperties() {
```

```
  console.log(dbUrl)
```

```
  console.log(root)
```



```
console.log(rootPwd)
}
```

### 3.2 建立資料表

```
function createTable() {
  try {
    var conn = Jdbc.getConnection(dbUrl, root, rootPwd);
    conn.createStatement().execute('CREATE TABLE {填入自己的表名}' +
      '(guestName VARCHAR(255), content VARCHAR(255), ' +
      'entryID INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(entryID));');
  } catch (err) {
    // TODO(developer) - Handle exception from the API
    console.log('Failed with an error %s', err.message);
  }
}
```

### 3.3 寫入資料

```
function writeOneRecord() {
  try {
    var conn = Jdbc.getConnection(dbUrl, root, rootPwd);
    const stmt = conn.prepareStatement('INSERT INTO {填入上述建立的表名}' +
      '(guestName, content) values (?, ?)');
    stmt.setString(1, 'First Guest');
    stmt.setString(2, 'Hello, world');
    stmt.execute();
  } catch (err) {
    // TODO(developer) - Handle exception from the API
    console.log('Failed with an error %s', err.message);
  }
}
```

### 3.4 讀取資料

```
function readFromTable() {
  try {
    const conn = Jdbc.getConnection(dbUrl, root, rootPwd);
    const start = new Date();
    const stmt = conn.createStatement();
    stmt.setMaxRows(1000);
    const results = stmt.executeQuery('SELECT * FROM {填入上述建立的表名}');
    const numCols = results.getMetaData().getColumnCount();

    while (results.next()) {
      let rowString = '';
      for (let col = 0; col < numCols; col++) {
        rowString += results.getString(col + 1) + '\t';
      }
      console.log(rowString);
    }
    results.close();
    stmt.close();

    const end = new Date();
    console.log('Time elapsed: %sms', end - start);
  } catch (err) {
    // TODO(developer) - Handle exception from the API
    console.log('Failed with an error %s', err.message);
  }
}
```

## 4. Line Notify

### 4.1 監控網站並發送 Line Notify 訊息

```
const token = "{自己的token}";

const monitors = [
  'https://notify-bot.line.me/doc/en/',
]

// 執行監測
const limit = 10; // 超過幾毫秒就推播訊息, 1000毫秒 = 1秒
function doMonitor() {
  for(let i = 0, len = monitors.length; i < len; i++) {
    const url = monitors[i];
    const start = new Date().getTime();
    const res = UrIFetchApp.fetch(url, { muteHttpExceptions: true
  }).getResponseCode();
    // 先檢查回應碼是否為 200
    if(res === 200) {
      // 是 200 就計算時間
      const end = new Date().getTime();
      const distance = end - start;
      if(distance > limit) {
        sendMessage(url + '\n回應時間 ' + (Math.ceil(distance / 1000)) + ' 秒!
        回應碼: ' + res)
      }
    }
    // 不是 200 就發訊息
    else {
      sendMessage(url + '\n回應碼異常: ' + res)
    }
  }
}
```

```
function calanderEvent(e) {
var now = new Date();
var sec2Before = new Date(now.getTime() - 2000);
var events = CalendarApp.getDefaultCalendar().getEvents(sec2Before,
now);
if(events.length>0)
{
const title = events[0].getTitle();
sendMessage(title);
}
}

// 發送訊息
function sendMessage(message) {
const option = {
method: 'post',
headers: { Authorization: 'Bearer ' + token },
payload: {
message: message
}
};
UrlFetchApp.fetch('https://notify-api.line.me/api/notify', option);
}
```