



# 구글 스프레드 시트 연동, Javascript



스마트론 프로젝트에서 슬비책임님이 만드신 구글 스프레드 시트를 연동한 pagelist 참고해서 작업했습니다.

## 파일

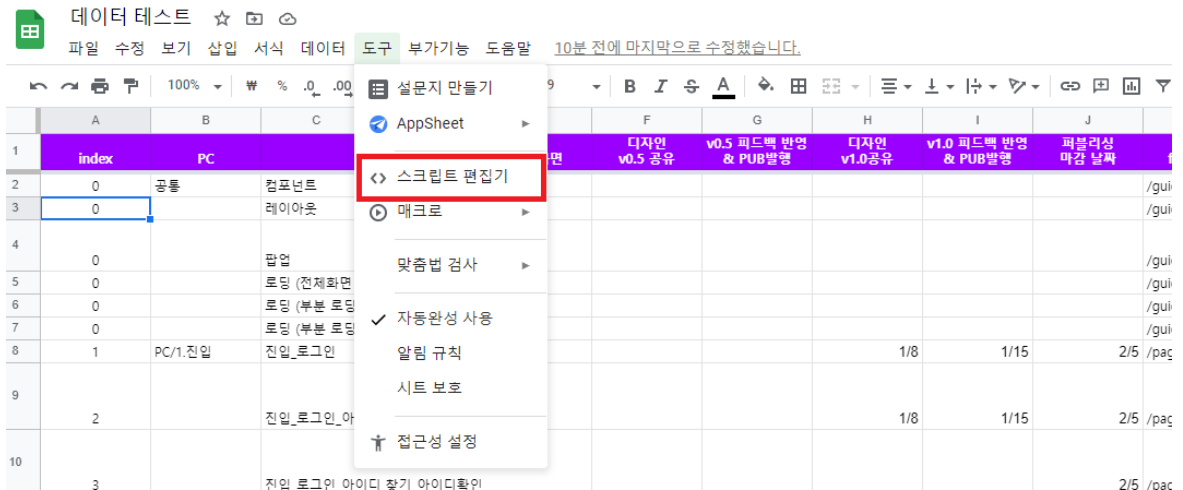
[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0863377f-ee59-40d8-a8af-482474eeb990/pagelist\\_google\\_javascript.html](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0863377f-ee59-40d8-a8af-482474eeb990/pagelist_google_javascript.html)

## 파일에 적용된 테스트 구글스프레드 시트 경로

<https://docs.google.com/spreadsheets/d/1ZwDtAqZxnk9b-GpV-jnkefl3lFerHmpEiO3LBA-YBLM/edit#gid=0>

## 구글 스프레드 시트 json 뽑기

1. 구글 스프레드 시트에서 스크립트 편집기 클릭



2. 파일 Code.gs 수정



1. 이 부분을 setup으로 바꾸는게 중요!! (기본: doGet)
2. 아래 코드로 작성 (이 안에 data 코드는 상황에 맞게 바꿔쓰면 된다.)  
우리가 가져오는 json은 결국 var data = []; 이 안에 있는 값을 쓰게 된다.

```
var SCRIPT_PROP = PropertiesService.getScriptProperties(); // new property service
// If you don't want to expose either GET or POST methods you can comment out the appropriate function
function doGet(e) {
  var lock = LockService.getPublicLock();
  lock.waitLock(30000);
  var rawData;
  try {
    var doc = SpreadsheetApp.openById(SCRIPT_PROP.getProperty("key"));
    var sheetName = e.parameter["sheetName"];
    var sheetInitial = doc.getSheetByName(sheetName);
    // var headers : 고정 header만 분리시킨 코드
    var headers = sheetInitial.getRange(1, 1, 1, sheetInitial.getLastColumn()).getValues()[0];
    // var rawData : 고정된 header 밑에 부분
    var rawData = sheetInitial.getRange(2, 1, sheetInitial.getLastRow(), sheetInitial.getLastColumn()).getValues();
  } catch (error) {
    return ContentService
      .createTextOutput(JSON.stringify({"ok": false, "error": error}))
      .setMimeType(ContentService.MimeType.JSON);
  }
  var data = [];
  var headerLength = headers.length;
  var rowLength = rawData.length;
  if (rawData) {
    for (var r=0; r < rowLength; r++) {
      if (rawData[r][0] === "") {
        continue;
      }
      var row = {};
      for (var i=0; i< headerLength; i++) {
        row[headers[i]] = rawData[r][i];
      }
      data.push(row)
    }
  }
  return ContentService
    .createTextOutput(JSON.stringify({"ok": true, "data": data}))
    .setMimeType(ContentService.MimeType.JSON);
}
function doPost(e){
  // do nothing
}
function setup() {
  var doc = SpreadsheetApp.getActiveSpreadsheet();
  SCRIPT_PROP.setProperty("key", doc.getId());
}
```

3. 배포 → 새 배포 클릭

### 3. 유형 선택 → 웹 앱 → 설명 작성

### 4. 웹 앱 URL이 나오는데 상단 코드는 뒤에 sheetName을 지정해주어야 한다.

(ex. <https://script.google.com/macros/s/AKfycbxX6oljde447IQejqwuFVCLLcdsRd3QxexOeIBFpixdBvfyIQgzwiZB/exec?sheetName=시트이름>)

### 5. json 파일 경로를 받아올 수 있다!

## 보충할 점

- 테이블 dom을 그릴때 구글 배열 값을 아래처럼 인덱스로 찍어주게 되면 매번 프로젝트마다 다시 pagelist를 수정해야하는 불편함이 있다. (그대로 가져다 쓰기에는 불편한 점이 있음)

```
listHtml += "<td>" + forTarget[1] + "</td>";  
listHtml += "<td>" + forTarget[2] + "</td>";
```

→ 인덱스로 찍은 이유를 생각해봤을 때 pagelist는 파일링크를 걸어줘야하고, 날짜 형식을 보기 편하게 바꿔줘야해서 이런 특수한 경우를 고려해서 만드신게 아닌가라는 생각이 들었다.

(+ 이슬비 코멘트 추가 : 페이지리스트와 구글시트가 동일하게 보여지지 않기에 인덱스로 찍은것 인데, 해당 부분은 단순히 페이지 리스트 입력을 구글시트로 받는 편리함을 위한것만이 아니라 UI/GUI 와 함께보는 IA 구글시트로 옮겨놓은 것이였고, 해서 IA가 업데이트 될 때 페이지 리스트도 자동으로 업데이트 하게 하는게 목적이였습니다. 때문에 시트의 내용이 페이지 리스트에서는 굳이 보여질 필요가 없는 정보가 너무 많았기에 정보를 가려서 넣을 필요성이 있었습니다. IA와 페이지리스트 연동시 동일한 문제는 모든 플젝에서 발생할 듯 하여 해당 부분은 인덱스 찍을때 주석으로 해당 값이 어디 값인지 표시 해두는게 좋을 것 같네요. 플젝마다 수정은 어쩔수 없을듯 합니다.)

```
listHtml += "<td class='filelink'><a href='" + initData.list[idx].type + baseURL + forTarget[4] + "' target='_blank'>" + forTarget  
listHtml += "<td class='state'>" + forTarget[5] + "</td>";
```

```
listHtml += "<td class='date'>" + getGoggleSheet.changeDate(new Date(forTarget[6])) + "</td>";
```

→ 해결방법 :

- 구글엑셀과 pagelist는 동일하게 보여지게 한다. (pagelist만 다르게 수정하지 않게 동일하게 맞춘다.)
- 날짜나, 링크를 걸어줘야할 때 구분자가 있으면 좋을 듯하다.