Google Apps Script

11-2025

Nhóm thực hiện
- Thân Hoàng Lộc
- Trần Anh Tuấn
- ...

Mục tiêu

- Giới thiệu tổng quan về Google Apps Script, và những khái niệm cơ bản
- Một vài ví dụ ứng dụng Google Apps Script với Google Docs, Sheets

Nội dung

Phần 1

Giới thiệu tổng quan Google Apps Script
Các giới hạn
Các khái niệm cơ bản
Các ví dụ với Google Docs, Sheets

Phần 2

Triggers, các cơ chế kích hoạt
Giao diện người dùng – UI và Services
- Menu tùy chỉnh
- Sidebar tùy chỉnh

Phần 3

Google Apps Script với Firebase

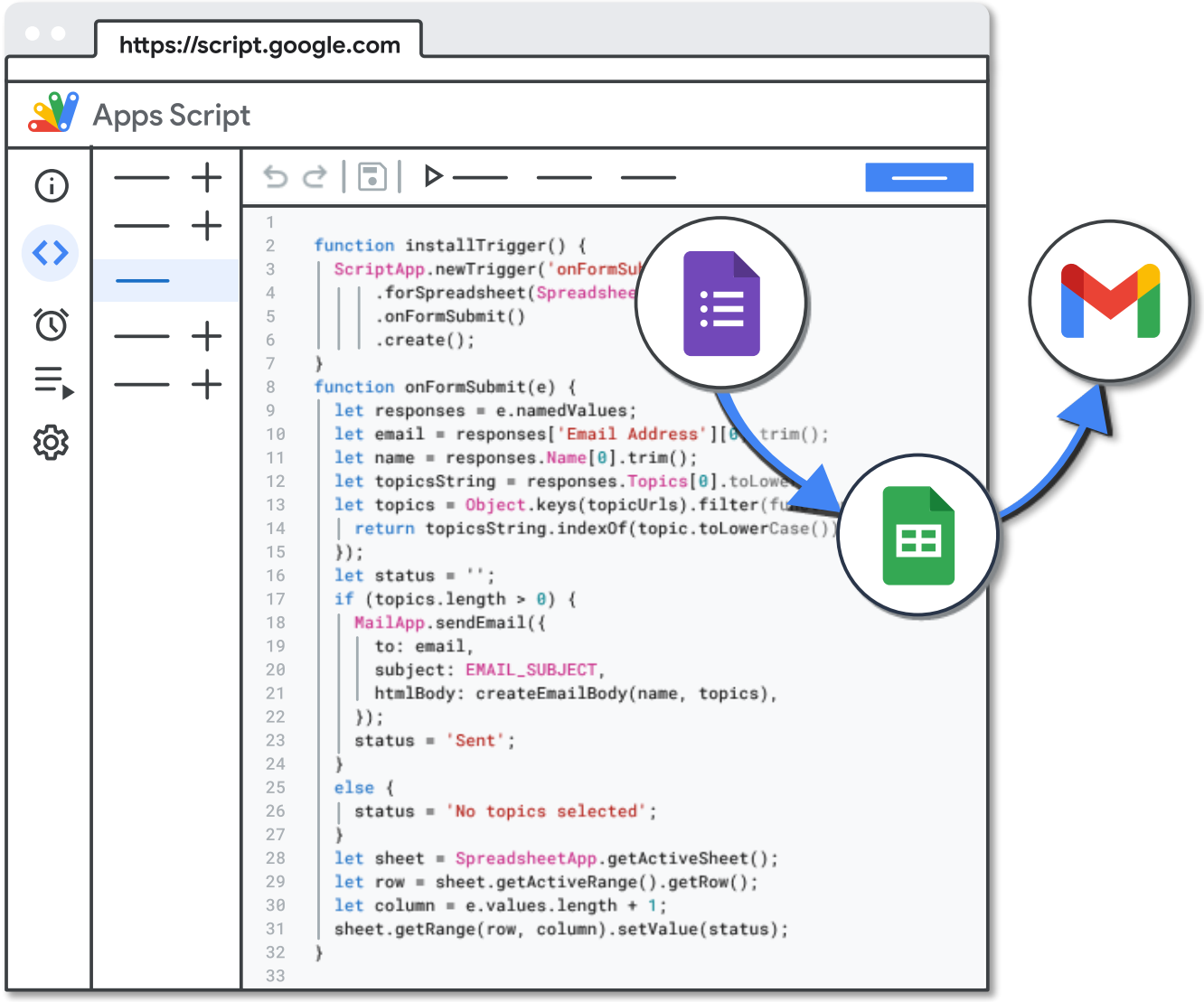
Nội dung

Phần 1

Giới thiệu tổng quan Google Apps Script
Các giới hạn
Các khái niệm cơ bản
Các ví dụ với Google Docs, Sheets

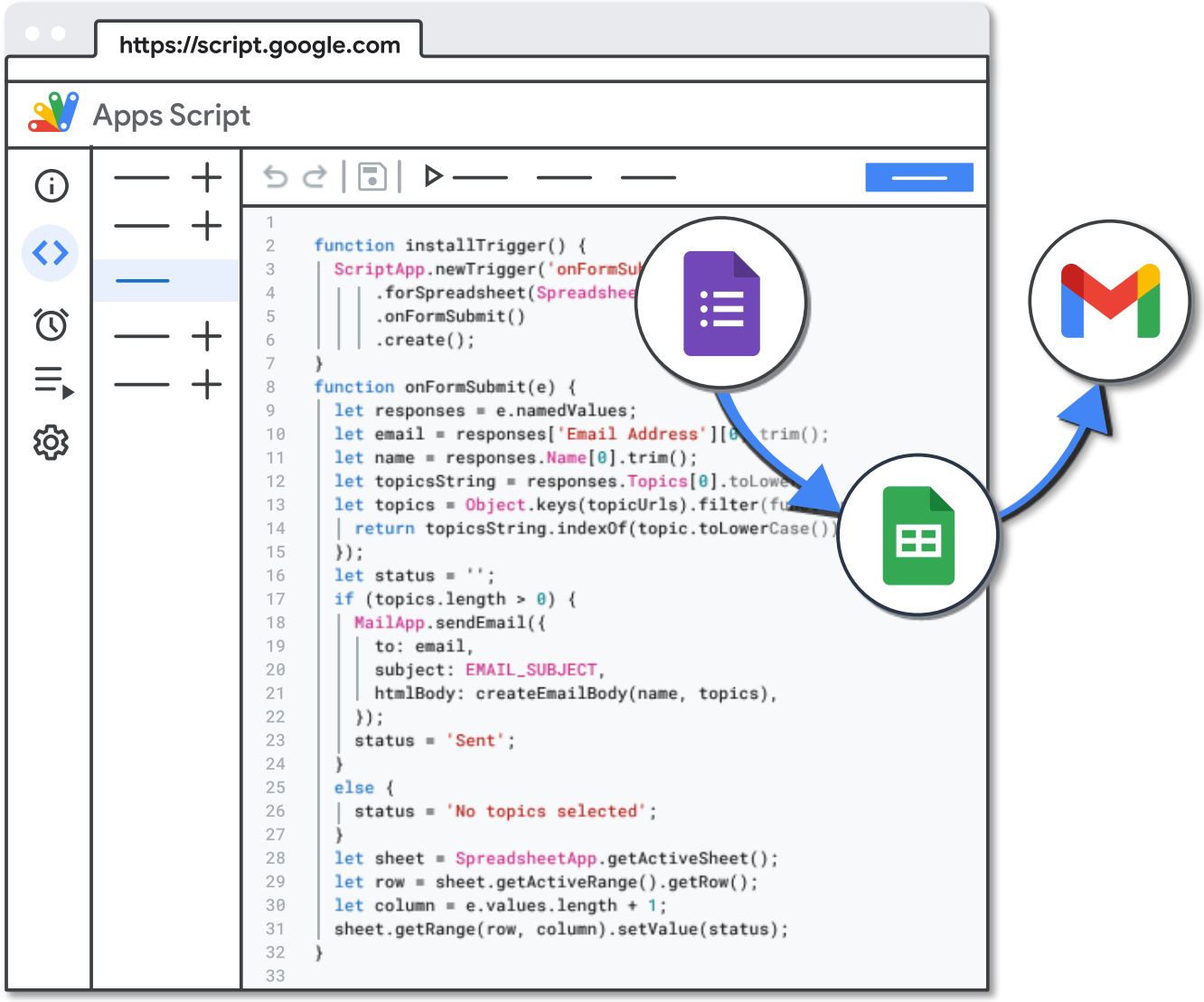
Google Apps Script (GAS) là gì?

- Là một nền tảng phát triển ứng dụng nhanh (Rapid Application Development) dựa trên JavaScript.
- Cho phép bạn tự động hóa, tích hợp và mở rộng các ứng dụng trong Google Workspace.
- Hoạt động trên nền tảng đám mây của Google, không cần cài đặt.



Tại sao nên sử dụng Google Apps Script?

- **Tự động hóa** các tác vụ lặp đi lặp lại, tốn thời gian.
- **Tạo các quy trình làm việc** tùy chỉnh cho doanh nghiệp của bạn.
- **Xây dựng các giải pháp** mà không cần đầu tư vào cơ sở hạ tầng.
- **Tăng năng suất** và hiệu quả công việc.

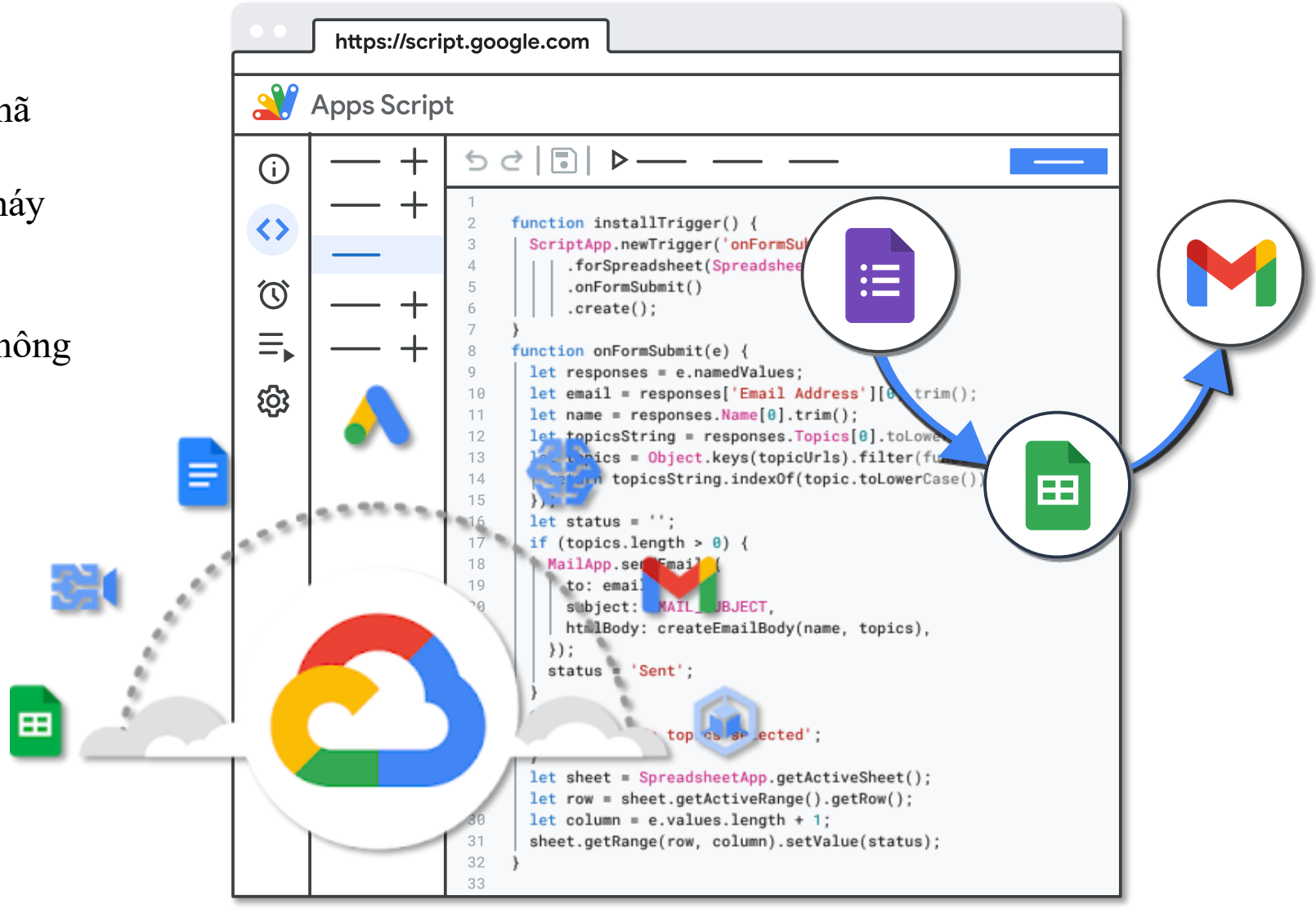


Google Apps Script - Đặc điểm chính

- **Dễ sử dụng**: Dựa trên ngôn ngữ JavaScript phổ biến.
- **Tích hợp sẵn**: Kết nối dễ dàng với Gmail, Sheets, Docs, Drive, Calendar, v.v.
- **Linh hoạt**: Có thể tạo ứng dụng web, add-on, hoặc các hàm tùy chỉnh.
- **Miễn phí**: Với tài khoản Google, bạn có thể bắt đầu ngay mà không cần cài đặt.

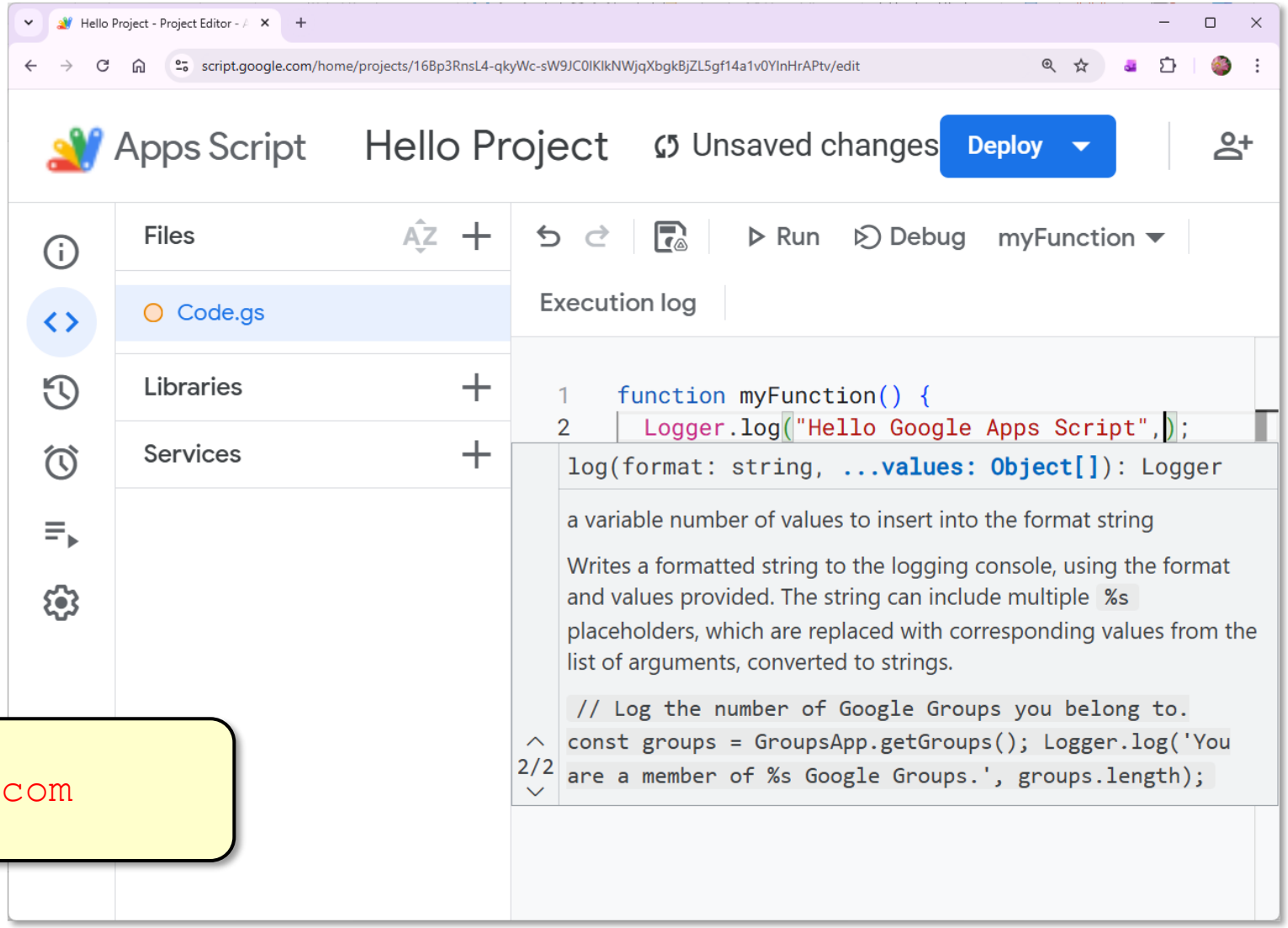
Google Apps Script - Cách hoạt động

- **Viết mã** trong trình soạn thảo mã trực tuyến.
- Mã được lưu và **thực thi** trên máy chủ của Google.
- **Tương tác** với các dịch vụ của Google và các **API** bên ngoài thông qua các lệnh gọi hàm.



Môi trường phát triển, và JavaScript

- Sử dụng cú pháp **JavaScript tiêu chuẩn**.
- Cung cấp một bộ **API** phong phú để tương tác với các dịch vụ của Google.
- Trình soạn thảo mã dựa trên web (**script.google.com**).
- Gỡ lỗi (**debugger**), ghi log (**logger**), và quản lý phiên bản.
- Có thể sử dụng công cụ dòng lệnh `clasp` để phát triển trên máy tính cá nhân.



script.google.com

Các giới hạn, giới hạn thực thi (run)

	Tài khoản Cá nhân/Miễn phí	Google Workspace (Trả phí)
Tổng thời gian chạy/lần	6 phút (360 giây)	6 phút (360 giây)
Tổng thời gian chạy/ngày	90 phút (1.5 giờ)	6 giờ
Số lần chạy đồng thời	30 lần	30 lần
Số lần gọi dịch vụ/ngày	20,000 lần	100,000 lần (hoặc cao hơn)

- **Thời gian chạy/lần (6 phút):**
  Nếu script của bạn chạy lâu hơn 6 phút (ví dụ: xử lý lượng lớn dữ liệu Sheets), nó sẽ bị tự động dừng lại.

- **Tổng thời gian chạy/ngày:**
  Nếu bạn có nhiều script chạy liên tục, tổng thời gian chúng chạy không được vượt quá giới hạn này.

Các giới hạn, giới hạn dịch vụ Google

A. Gmail (GmailApp / MailApp)

Hành động	Tài khoản Cá nhân/Miễn phí	Google Workspace (Trả phí)
Email người nhận/ngày	100 người nhận	1,500 - 2,000 người nhận
Tổng Email được gửi/ngày	100 email	1,500 - 2,000 email

B. Google Drive (DriveApp)

Hành động	Giới hạn hàng ngày (Xấp xỉ)
Tổng số lần gọi DriveApp/ngày	20,000 lần
Số lần đọc/ghi file/ngày	500,000 lần
Số lần tạo file/ngày	10,000 lần

Các giới hạn, giới hạn khác

A. Triggers

	Giới hạn
Tổng số triggers/người dùng	20 triggers
Thời gian chạy định kỳ tối thiểu	1 phút

B. Bộ nhớ Cache (Cache Service)

Kích thước dữ liệu/lần lưu:	100 KB
Thời gian lưu trữ tối đa:	6 giờ

C. Google Sheets

Tổng số cell / ô trong tài liệu, gồm:	10 triệu
sheet x row x column	
Số cột tối đa (columns) :	18,278

D. Giới Hạn Mã (Code Limitations)

	Giới hạn
Kích thước dự án (Project Size)	50 MB
Dung lượng lưu trữ/script	10 MB

Thư viện bên ngoài
Không thể sử dụng thư viện Node.js/npm trực tiếp.
Chỉ có thể sử dụng các Thư viện Apps Script khác.

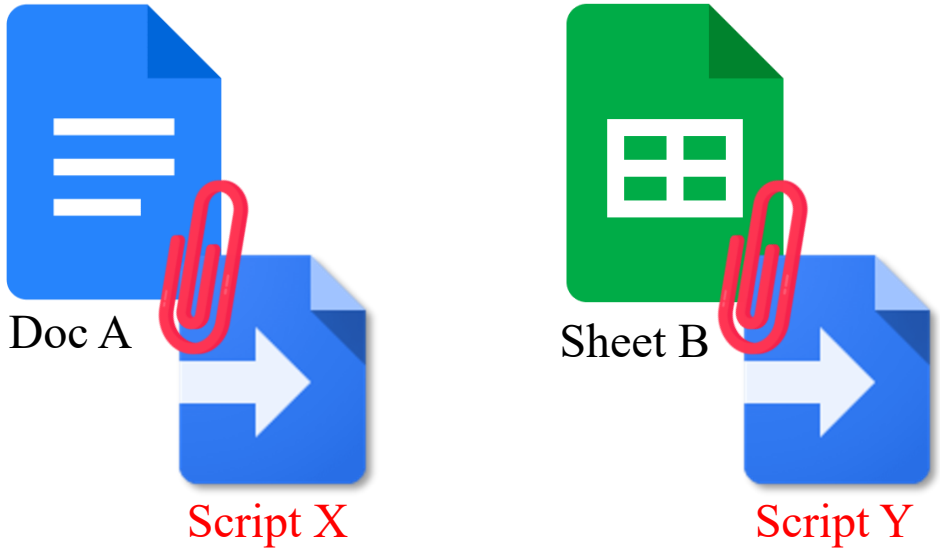
E. UrlFetchApp (Kết nối bên ngoài/REST API)

Tổng số lần gọi UrlFetch/ngày: 20,000 lần
(cho tài khoản miễn phí)

Dịch vụ này được sử dụng khi bạn kết nối GAS với các dịch vụ bên ngoài như Firebase, Slack, hoặc các REST API khác

Bound-Script, và Standalone-Script

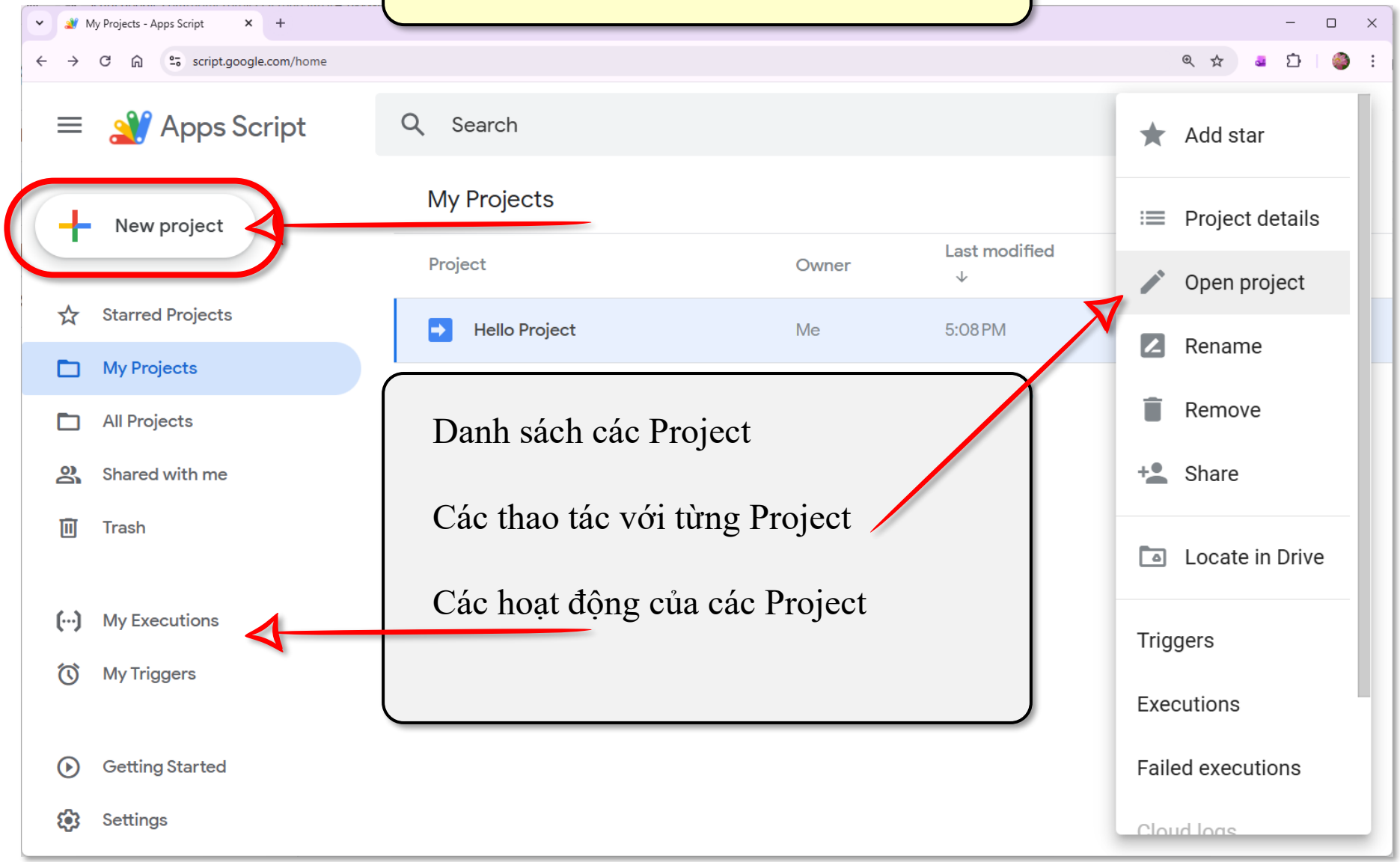
Bound-Script (Gắn kết):
Là script **gắn kết** với một tài liệu cụ thể
ví dụ:
- *Script X được gắn kết với Doc A,*
- *Script Y được gắn kết với Sheet B*



Standalone-Script (Độc lập):
Là script **không gắn kết** với tài liệu cụ thể nào.

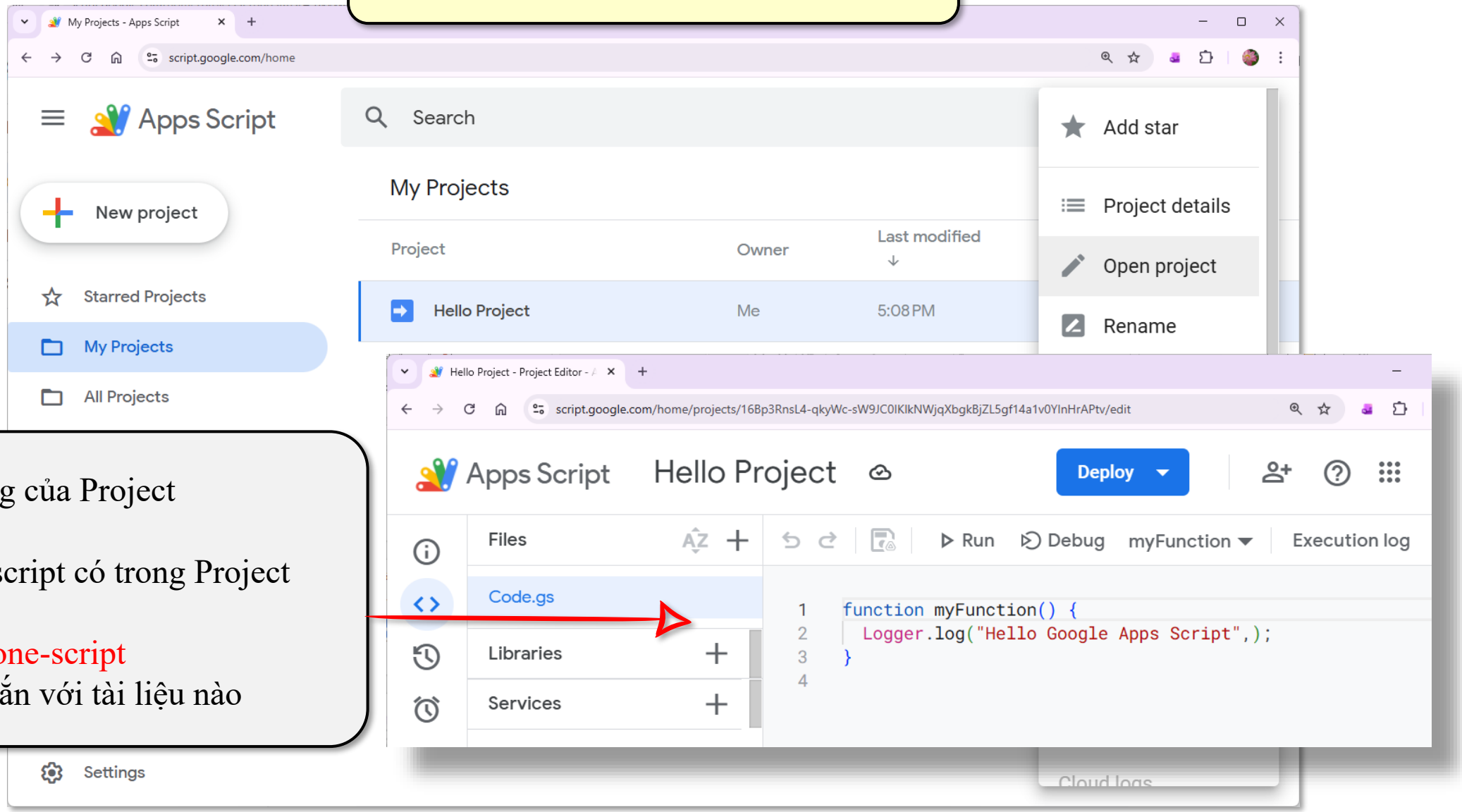
Bắt đầu

`script.google.com`



Bắt đầu

script.google.com

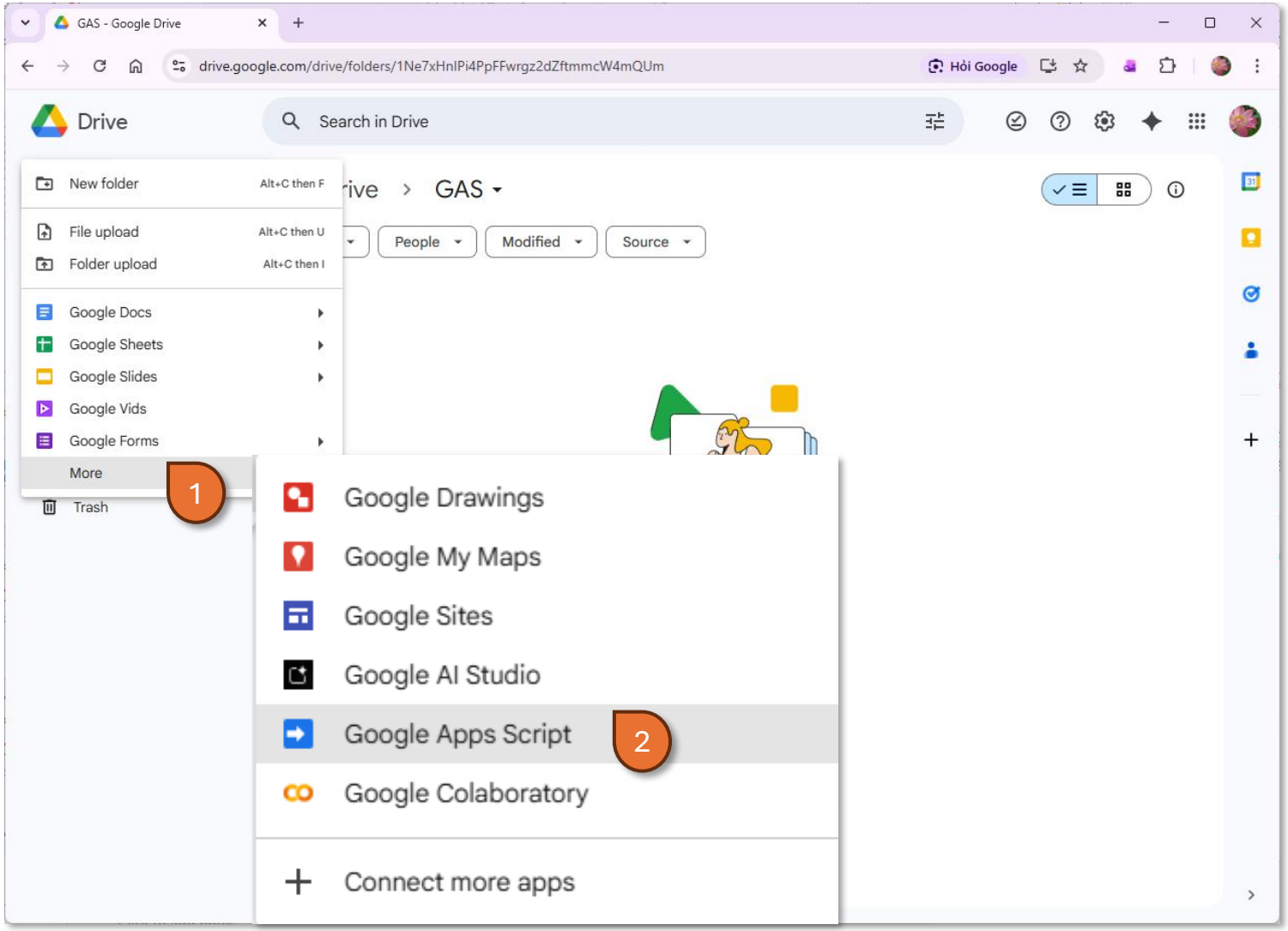


Nội dung của Project

Và các script có trong Project

Standalone-script
không gắn với tài liệu nào

Bound-Script, và Standalone-Script

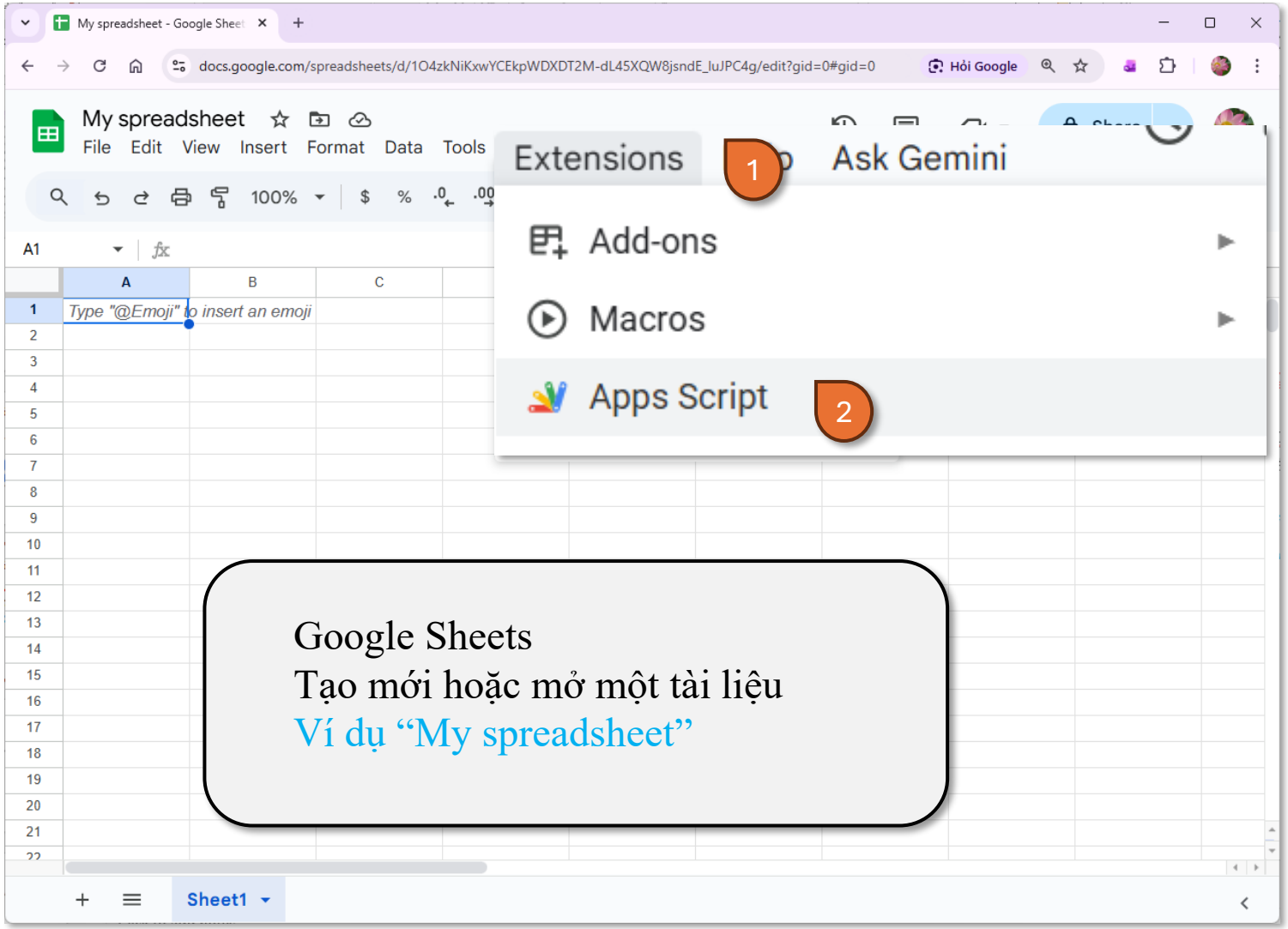


Tạo Standalone-Script (Độc lập):

Dùng dịch vụ Google Drive
Chọn More từ "**New File**" và chọn tài liệu dạng "Google Apps Script"
để mở Tab mới và bắt đầu tạo script

Cách làm này gọi là tạo standalone-script. Nghĩa là script không gắn liền với tài liệu cụ thể nào

Bound-Script, và Standalone-Script

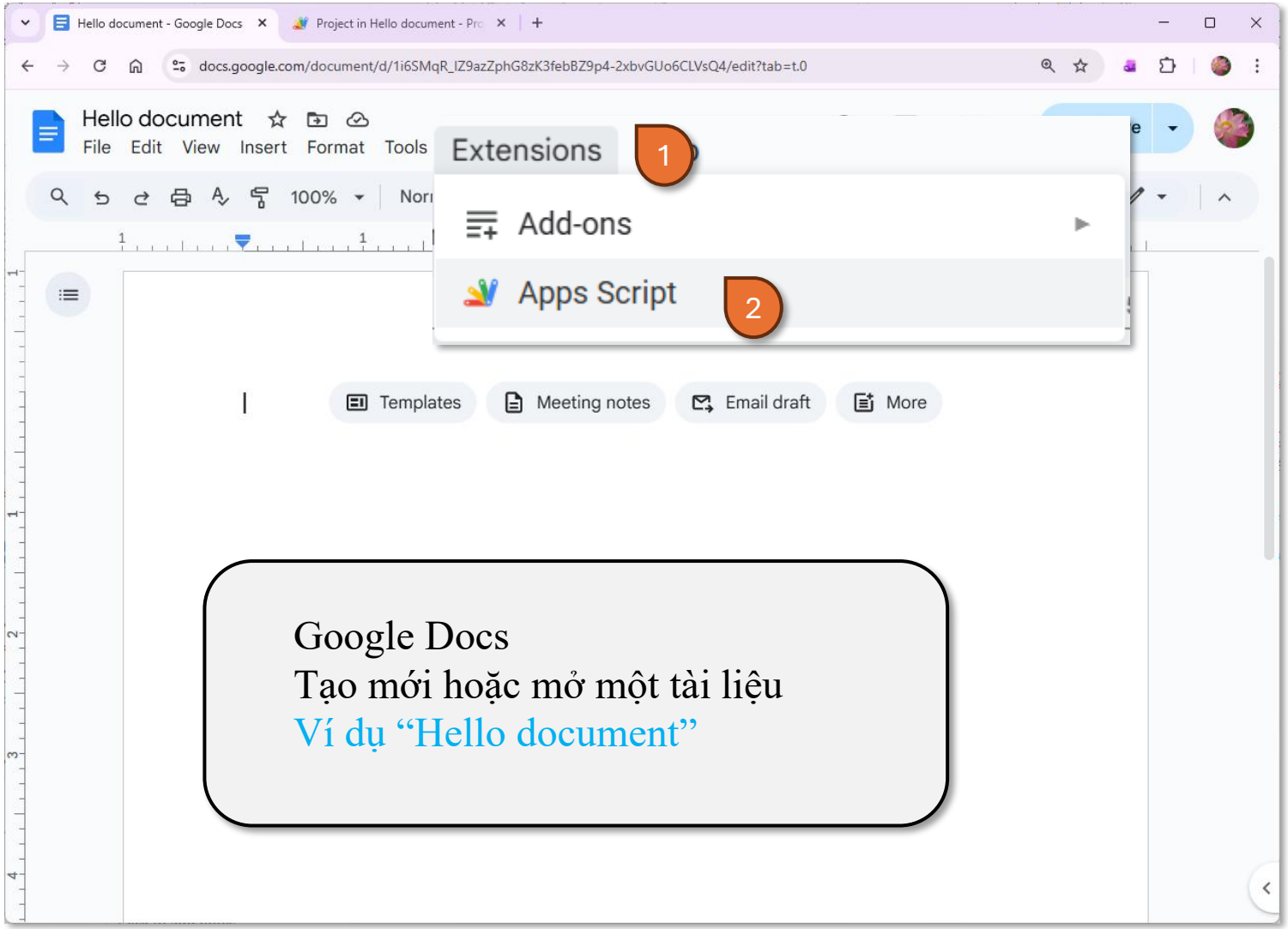


Tạo Bound-Script (Gắn kết):

Chọn **menu** "**Apps Script**"
để mở Tab mới và bắt đầu tạo script

Cách làm này gọi là tạo bound-script.
Nghĩa là script được gắn liền với Sheet

Bound-Script, và Standalone-Script



Tạo Bound-Script (Gắn kết):

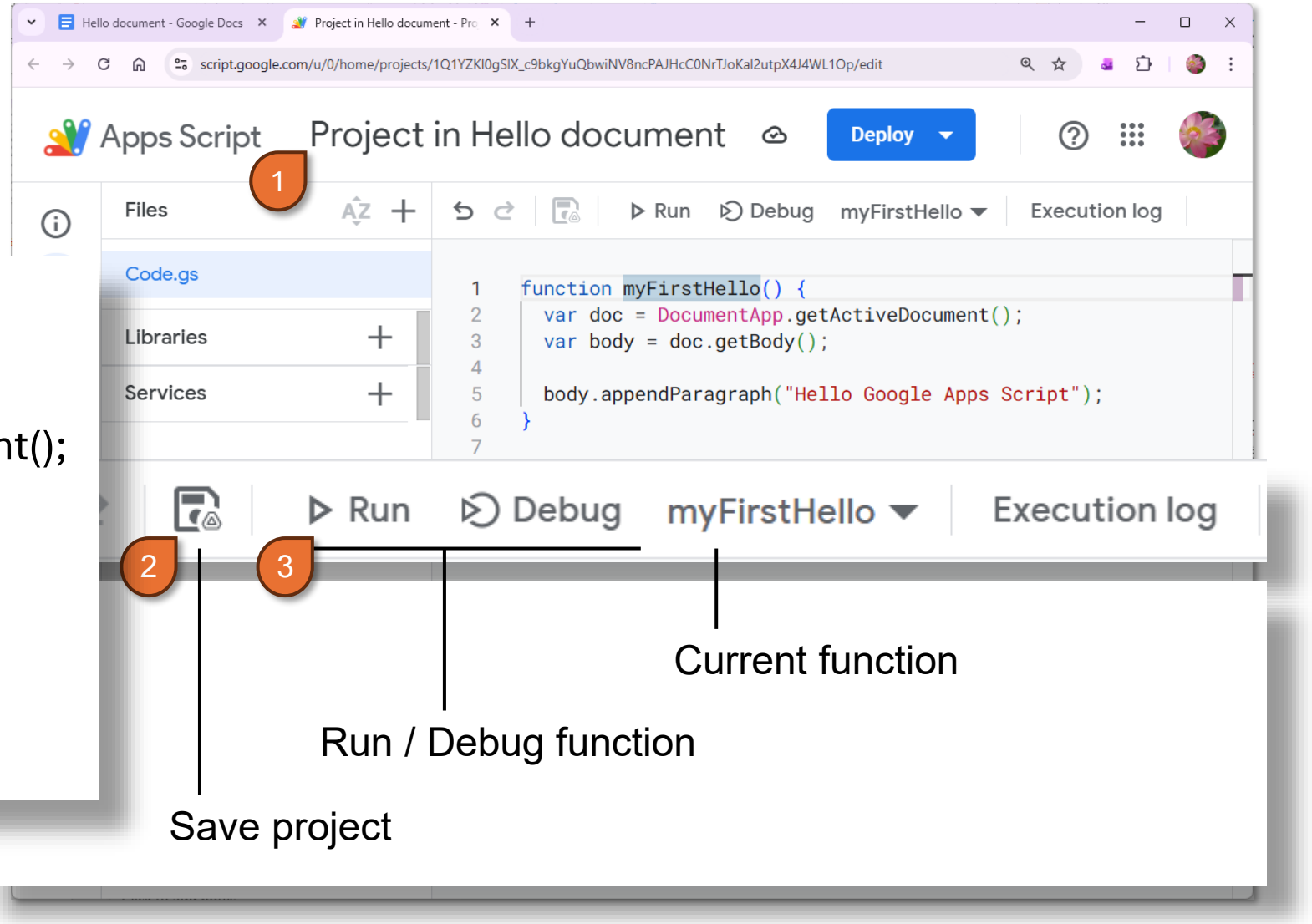
Chọn **menu** "**Apps Script**"
để mở Tab mới và bắt đầu tạo script

Cách làm này gọi là tạo bound-script.
Nghĩa là script được gắn liền với Doc

Ví dụ tạo Bound-Script, từ Google Docs

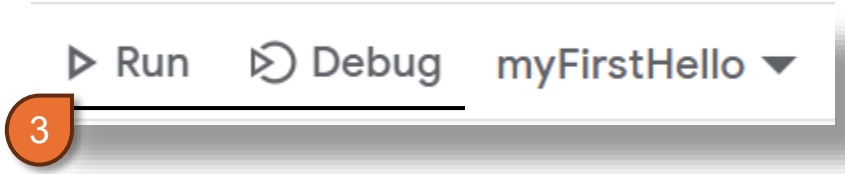
Tạo bound-script từ
Google Docs
Ví dụ "Hello document"

```
function myFirstHello() {
  var doc =
      DocumentApp.getActiveDocument();
  var body = doc.getBody();

  body.appendParagraph("
      Hello Google Apps Script");
}
```



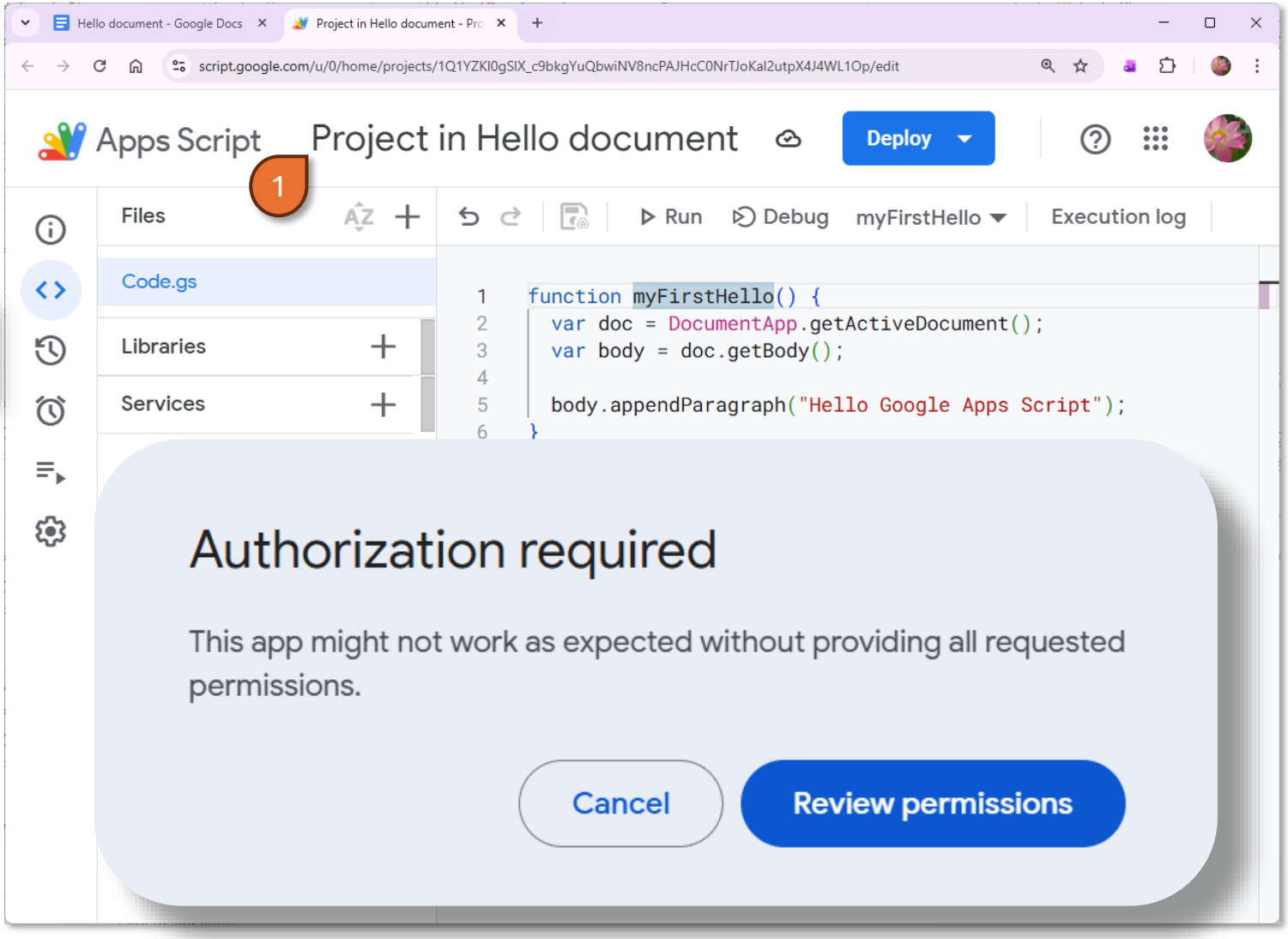
Ví dụ tạo Bound-Script, từ Google Docs

Tạo bound-script từ
Google Docs
Ví dụ "Hello document"



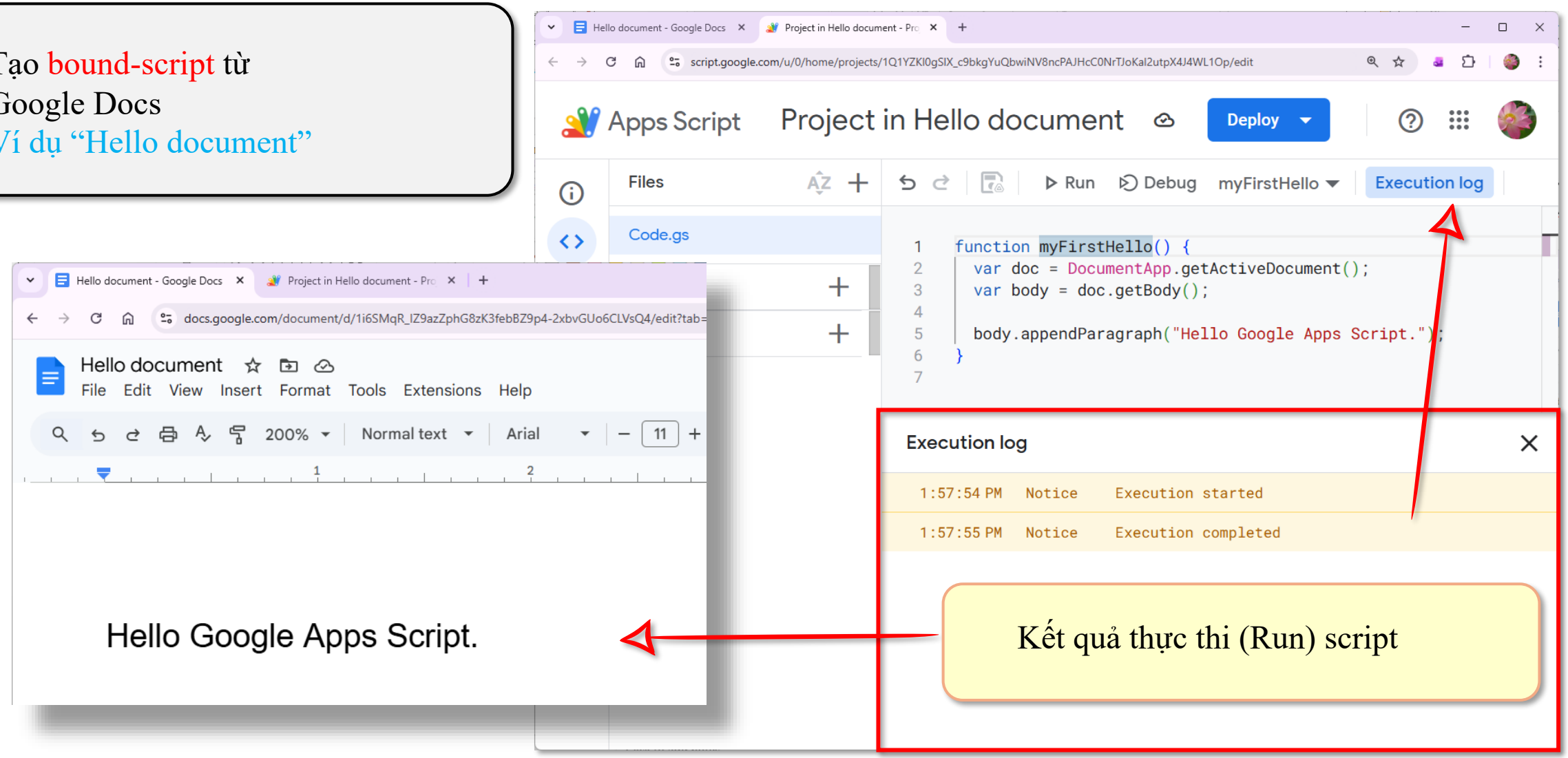
Run và Review permissions

khi Run lần đầu, cần cấp quyền cho Project (chứa script)



Ví dụ tạo Bound-Script, từ Google Docs

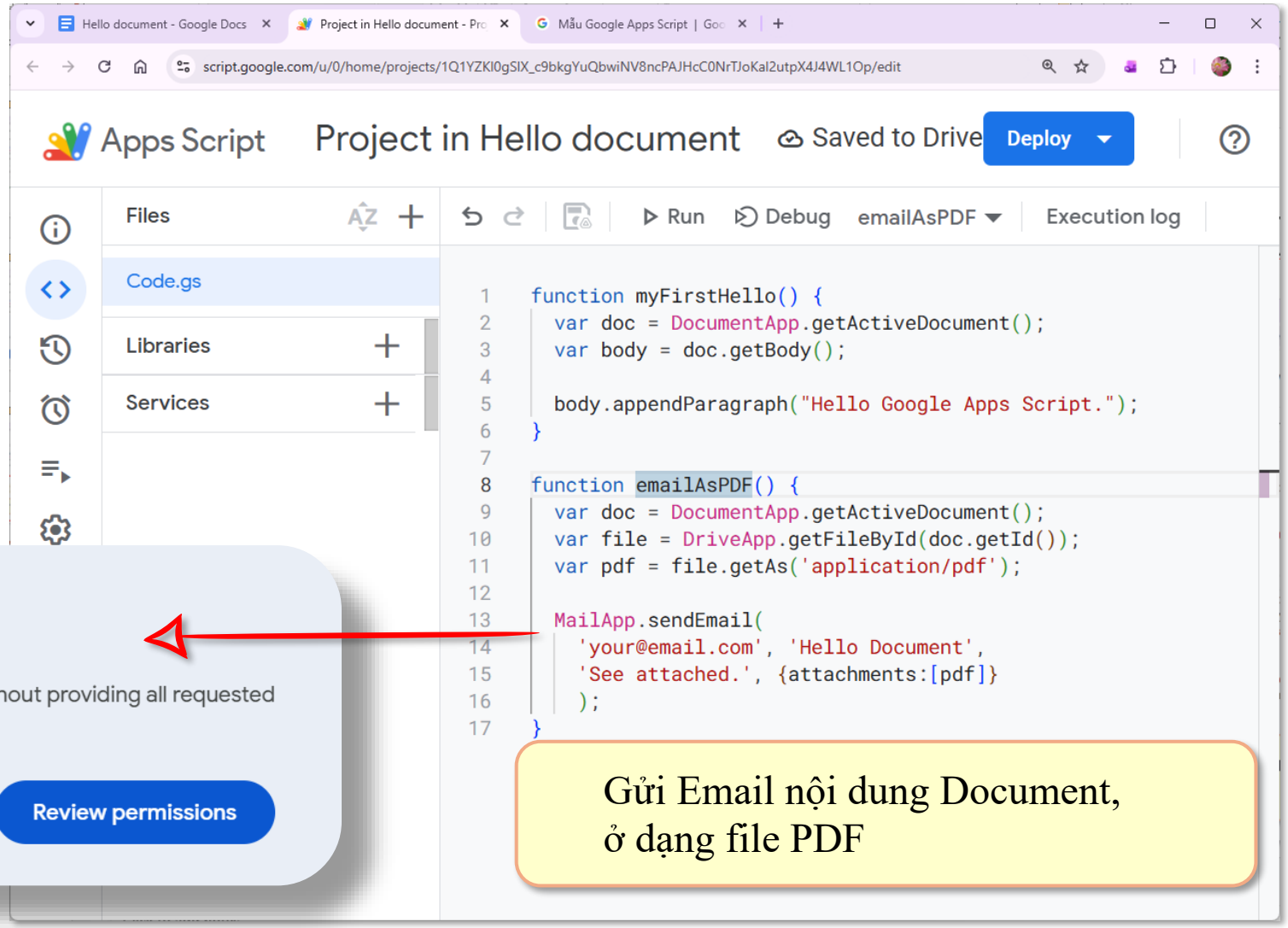
Tạo bound-script từ Google Docs
Ví dụ "Hello document"



Ví dụ tạo Bound-Script, từ Google Docs

Tạo bound-script + gửi email từ Google Docs
Ví dụ "Hello document"

Lưu ý
Các dịch vụ có sử dụng trong script như DocumentApp, DriveApp, MailApp
Cần được **cấp quyền** để Script có thể sử dụng và thực thi được



Các ví dụ Apps Script với Docs & Sheets, …

👉 

100+Apps-Script-Examples.pdf

👉 

GoogleSheet+Apps-Script-Examples.pdf

👉 

Apps+Script+Code+Snippets+V2.pdf

Nội dung

Phần 2

Triggers, các cơ chế kích hoạt
Giao diện người dùng – UI và Services
- Menu tùy chỉnh
- Sidebar tùy chỉnh

Dịch vụ và APIs

* **Dịch vụ Google Workspace:** `SpreadsheetApp`, `GmailApp`, `DocumentApp`, `DriveApp`, v.v.
* **Dịch vụ nâng cao:** BigQuery, Google Analytics, YouTube Analytics, v.v.
* **Dịch vụ tiện ích:** `UrlFetchApp` (gọi API ngoài), `CacheService`, `PropertiesService`.

Triggers (Các cơ chế kích hoạt)

Triggers

* Cho phép script chạy tự động khi một sự kiện xảy ra.
* - **Simple Triggers**: `onOpen()`, `onEdit()`, `onInstall()`.
* - Installable Triggers: Có thể cài đặt để chạy theo thời gian hoặc khi có sự kiện (ví dụ: khi form được gửi).

Time-driven Triggers

Chạy script vào một thời điểm cụ thể hoặc theo một lịch trình lặp lại (hàng giờ, hàng ngày, hàng tuần).

Ví dụ:

- Gửi báo cáo hàng ngày
- Dọn dẹp dữ liệu hàng tuần.

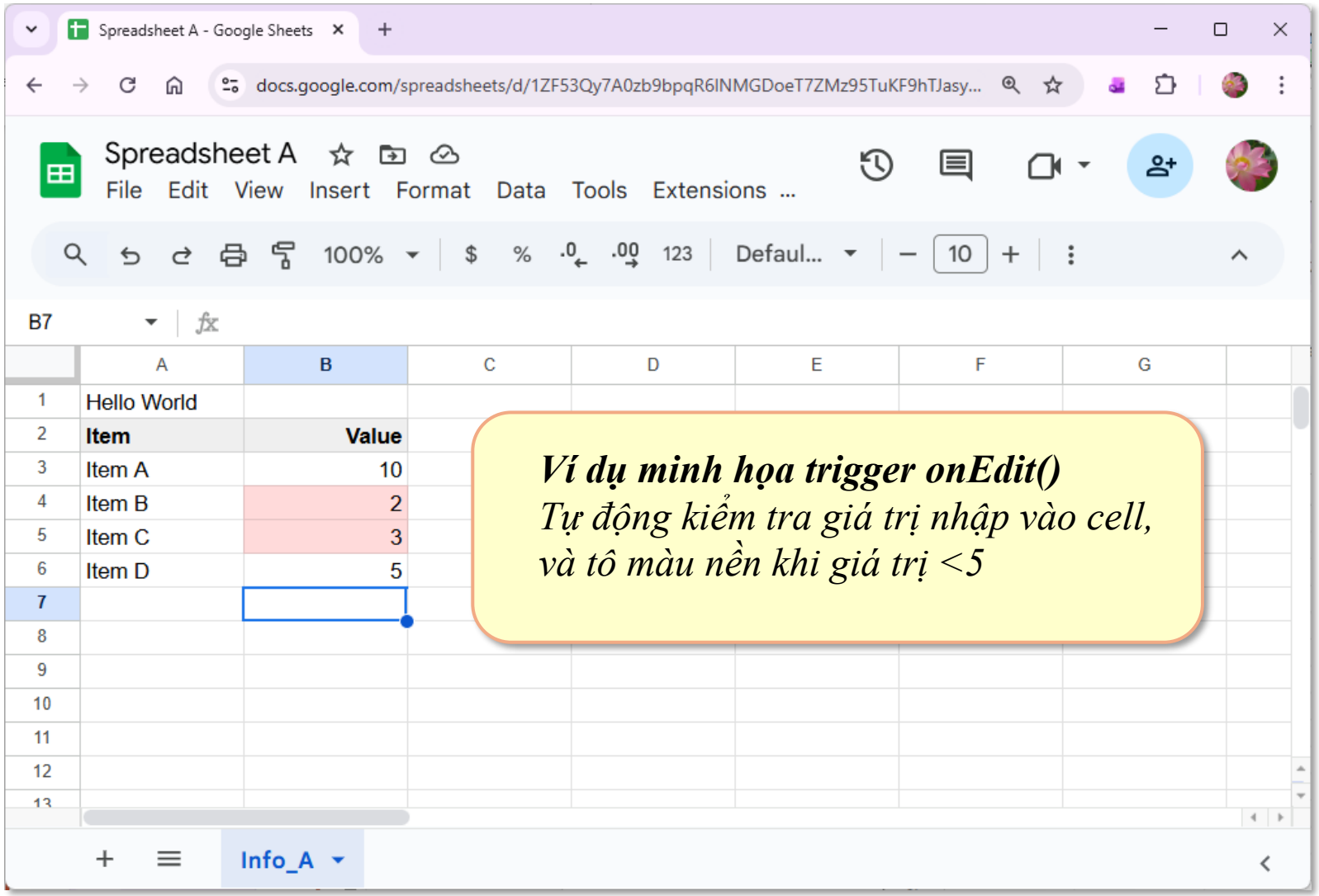
Event-driven Triggers

Chạy script khi có một sự kiện xảy ra trong một ứng dụng Google.

Ví dụ:

- Khi một hàng mới được thêm vào Google Sheets
- Khi một email mới được nhận
- Khi một sự kiện lịch được tạo.

Triggers, onEdit() (ví dụ Google Sheets)



Ví dụ minh họa trigger onEdit()
Tự động kiểm tra giá trị nhập vào cell, và tô màu nền khi giá trị <5

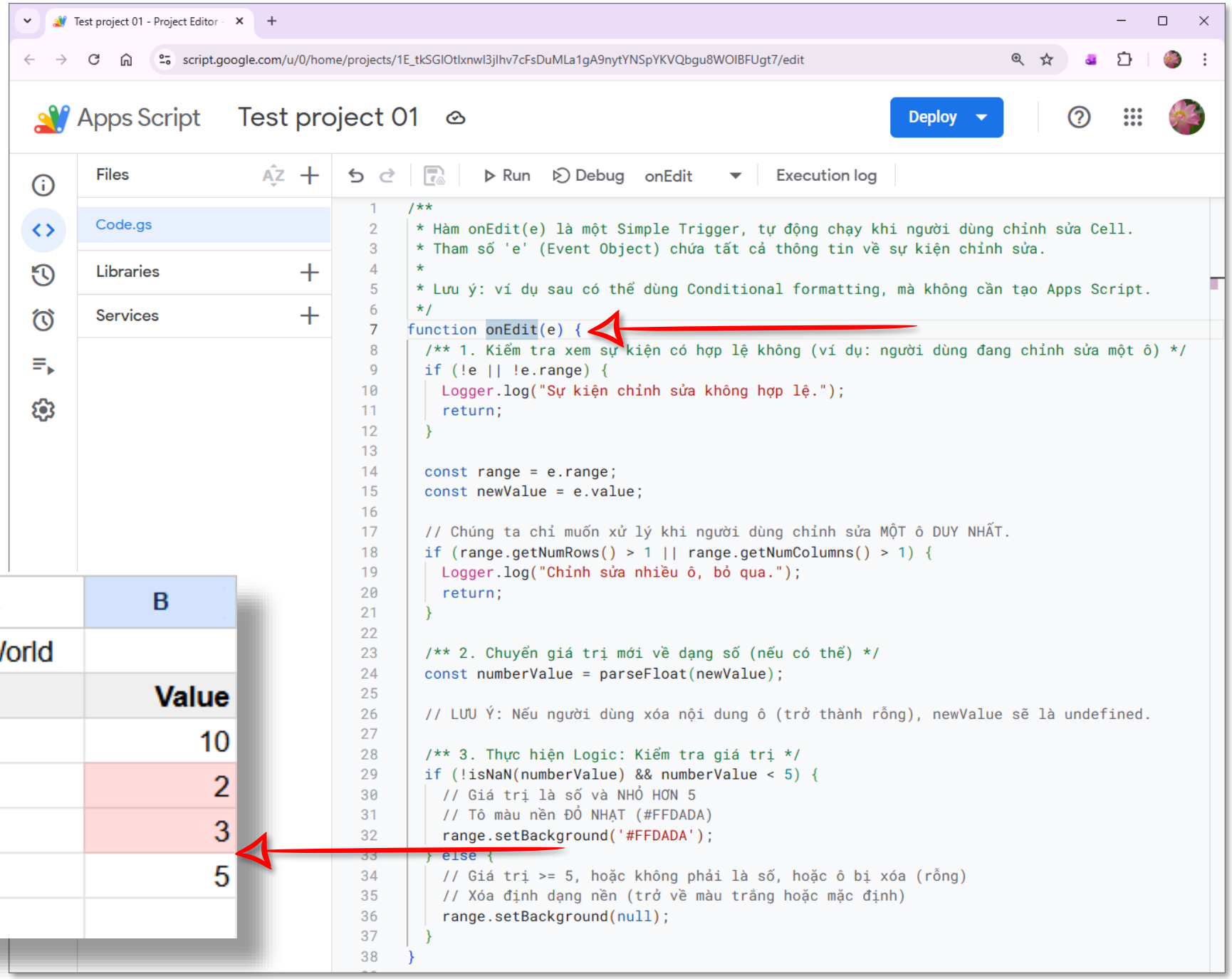
Tạo bound-script
từ Google Sheets, sử dụng
trigger onEdit()

Lưu ý
Có thể sử dụng tính năng có sẵn Format \ **Conditional formatting** để thay thế cho
Script + trigger onEdit()

→ **Nên ưu tiên sử dụng các tính năng có sẵn để tối ưu hiệu suất**

Triggers, onEdit
(ví dụ Google Sheets)

Tạo bound-script
từ Google Sheets, sử dụng
trigger onEdit()



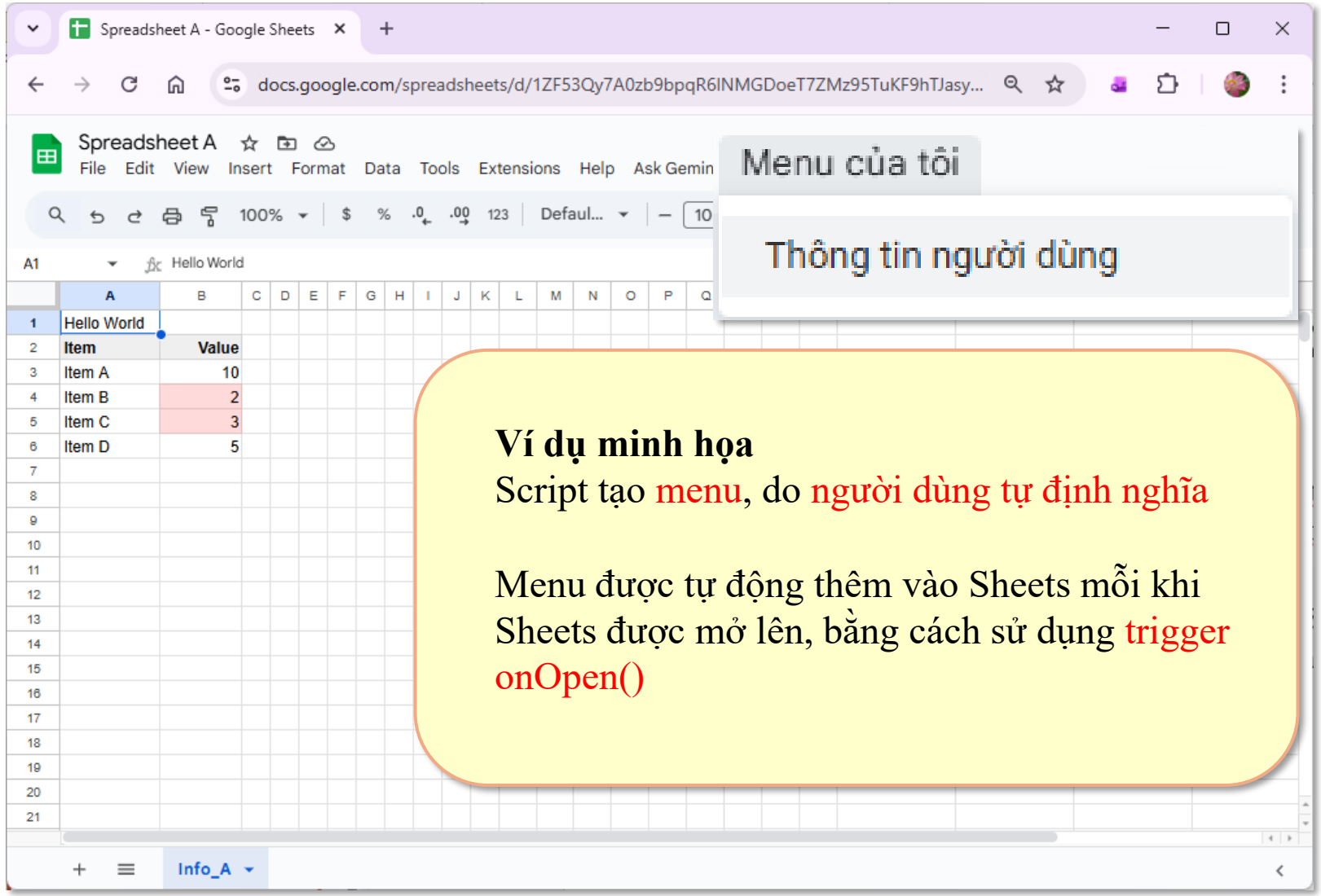
	A	B
1	Hello World	
2	**Item**	**Value**
3	Item A	10
4	Item B	2
5	Item C	3
6	Item D	5
7		

Giao diện người dùng (UI)

* **Hộp thoại (Dialogs)**: `SpreadsheetApp.getUi().alert()` và `prompt()`.
* **Thông báo (Toasts)**: `SpreadsheetApp.getActiveSpreadsheet().toast()`.

* **Menu tùy chỉnh**: Thêm các mục menu mới vào giao diện người dùng của Google Sheets, Docs, Forms.
* **Thanh bên (Sidebars)**: Tạo các thanh bên tùy chỉnh với nội dung HTML.

Giao diện người dùng (UI)



Ví dụ minh họa
Script tạo menu, do người dùng tự định nghĩa

Menu được tự động thêm vào Sheets mỗi khi Sheets được mở lên, bằng cách sử dụng trigger onOpen()

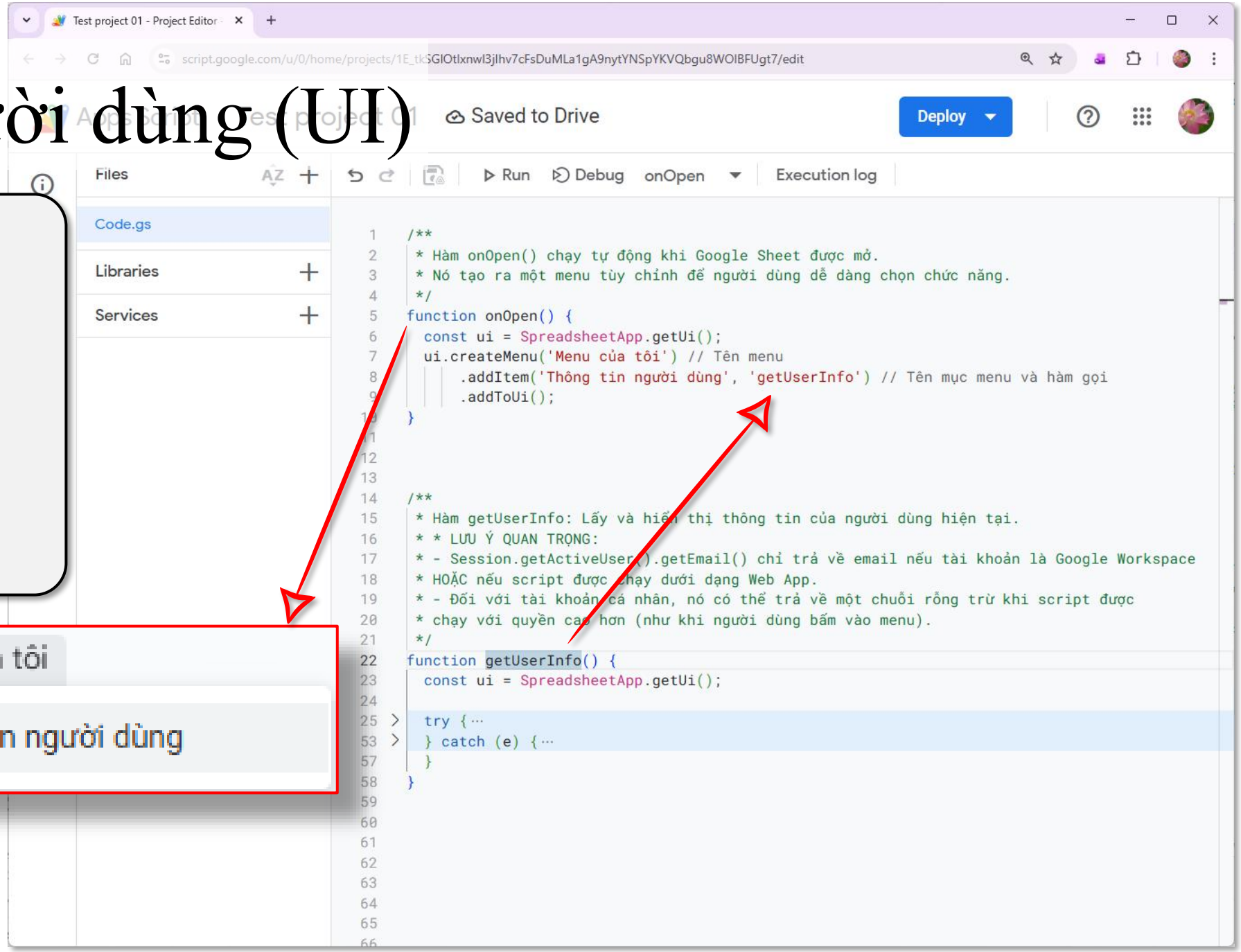
Tạo bound-script từ Google Sheets, sử dụng trigger onOpen()

Để tạo menu tùy chỉnh, và Hộp thoại trình bày thông tin người dùng

Giao diện người dùng (UI)

Tạo bound-script
từ Google Sheets, sử dụng
trigger onOpen()

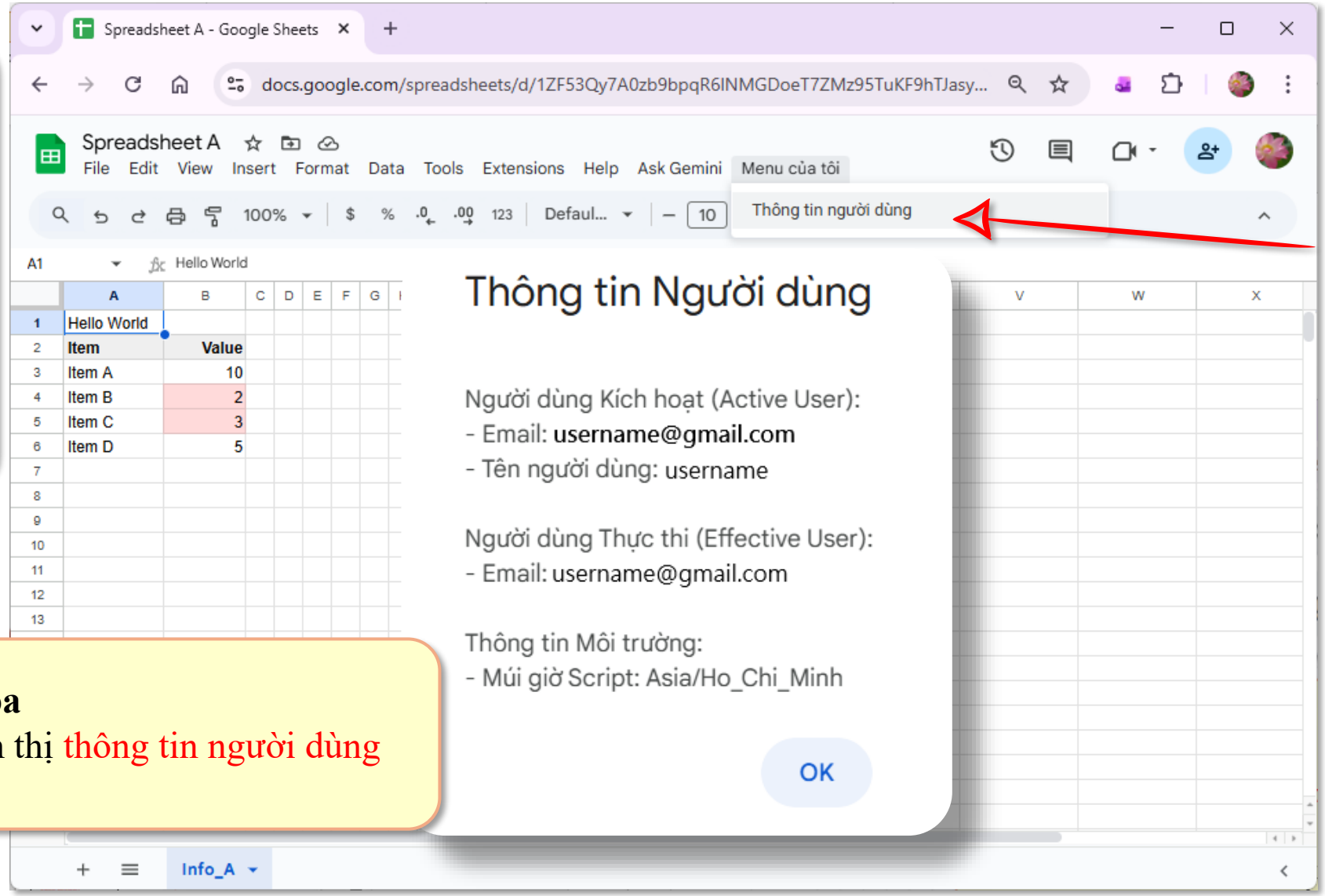
Để tạo menu tùy chỉnh, và
Hộp thoại trình bày thông
tin người dùng



Giao diện người dùng (UI)

Tạo bound-script
từ Google Sheets, sử dụng
trigger onOpen()

Để tạo menu tùy chỉnh, và
Hộp thoại trình bày thông
tin người dùng

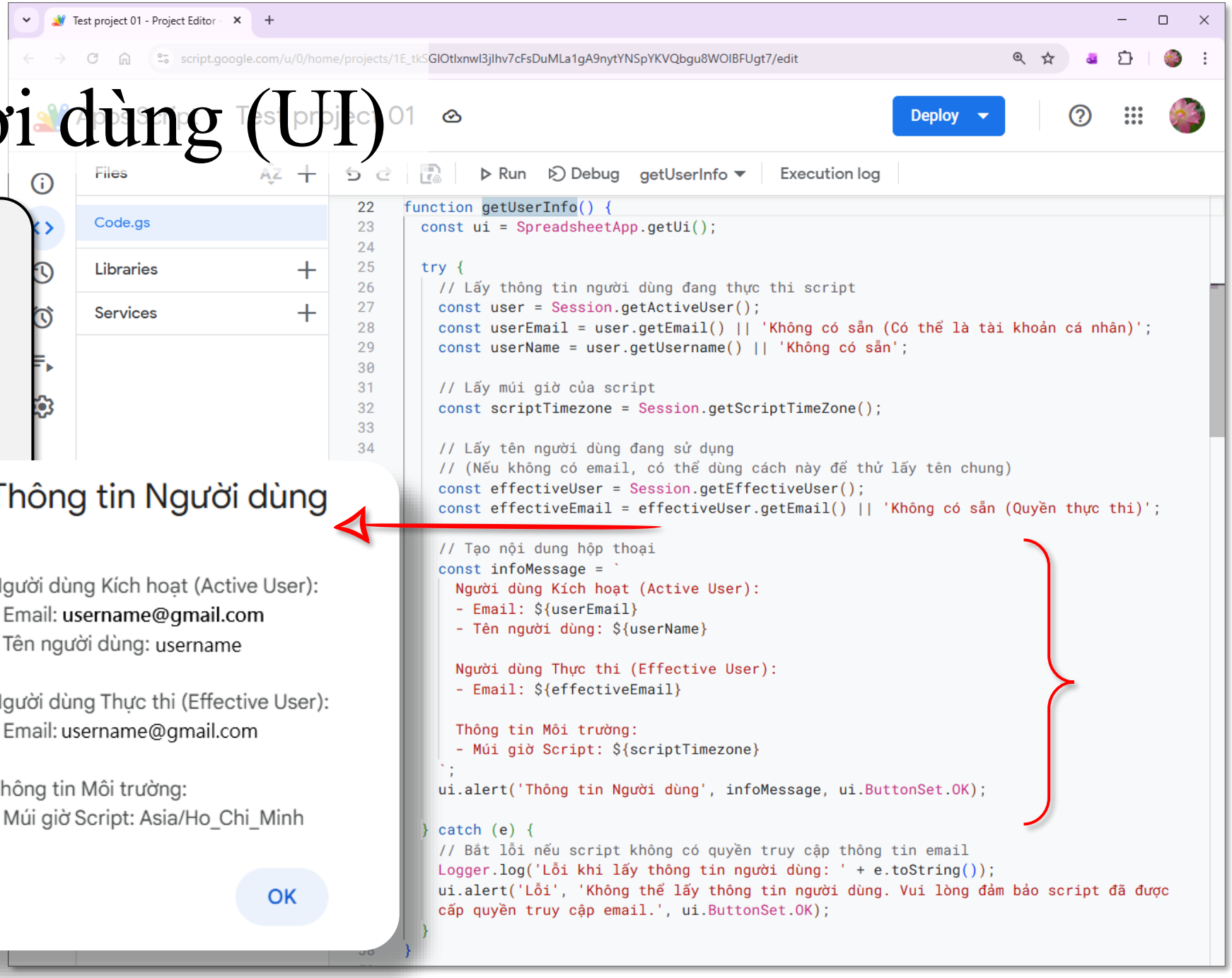


Ví dụ minh họa
Hộp thoại, hiển thị thông tin người dùng

Giao diện người dùng (UI)

Tạo bound-script
từ Google Sheets, sử dụng
trigger onOpen()

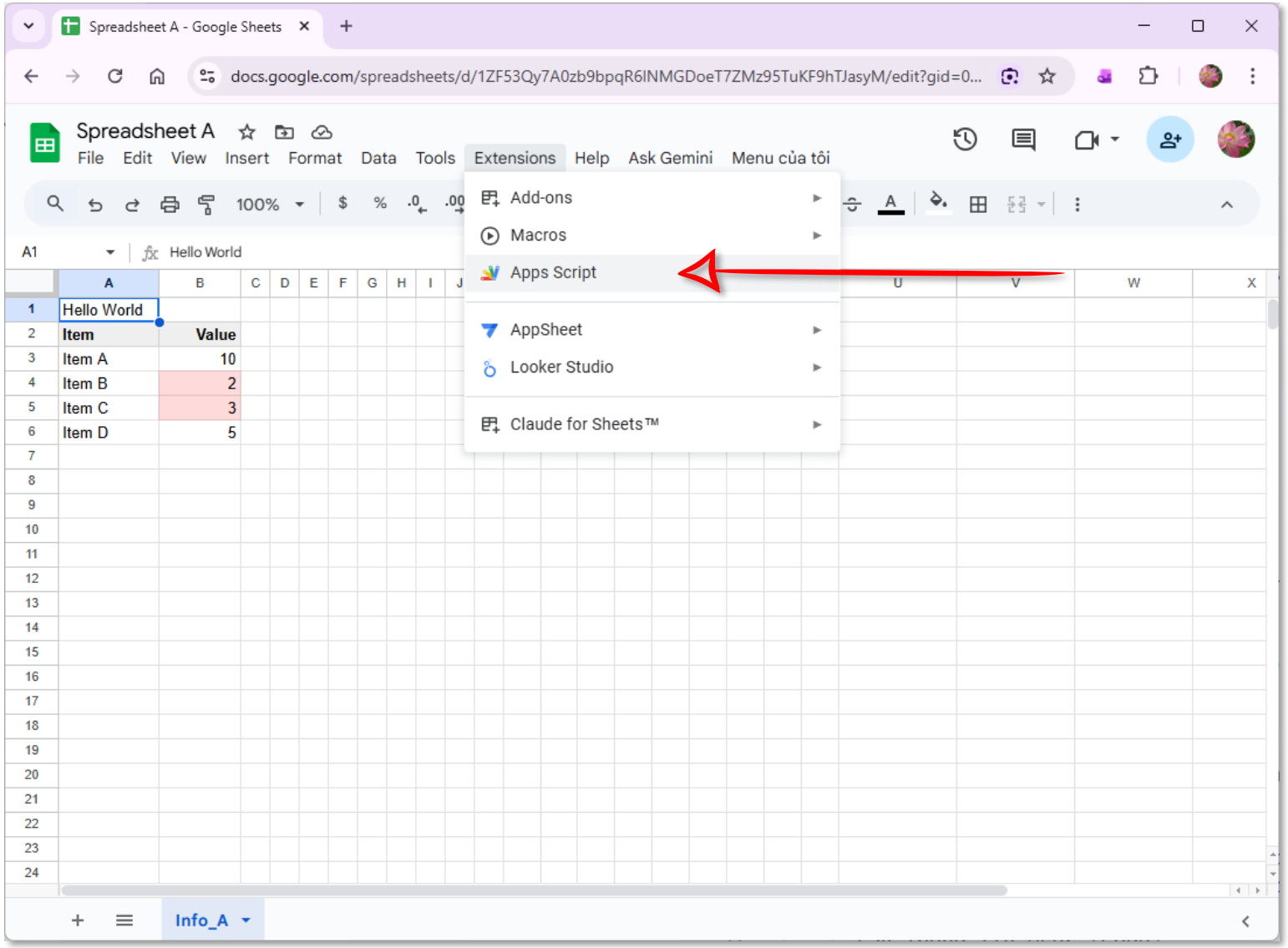
Để tạo menu tùy chỉnh, và
Hộp thoại trình bày thông
tin người dùng




```gs
function getUserInfo() {
  const ui = SpreadsheetApp.getUi();

  try {
    // Lấy thông tin người dùng đang thực thi script
    const user = Session.getActiveUser();
    const userEmail = user.getEmail() || 'Không có sẵn (Có thể là tài khoản cá nhân)';
    const userName = user.getUsername() || 'Không có sẵn';

    // Lấy múi giờ của script
    const scriptTimezone = Session.getScriptTimeZone();

    // Lấy tên người dùng đang sử dụng
    // (Nếu không có email, có thể dùng cách này để thử lấy tên chung)
    const effectiveUser = Session.getEffectiveUser();
    const effectiveEmail = effectiveUser.getEmail() || 'Không có sẵn (Quyền thực thi)';

    // Tạo nội dung hộp thoại
    const infoMessage = `
      Người dùng Kích hoạt (Active User):
      - Email: ${userEmail}
      - Tên người dùng: ${userName}

      Người dùng Thực thi (Effective User):
      - Email: ${effectiveEmail}

      Thông tin Môi trường:
      - Múi giờ Script: ${scriptTimezone}
    `;
    ui.alert('Thông tin Người dùng', infoMessage, ui.ButtonSet.OK);

  } catch (e) {
    // Bắt lỗi nếu script không có quyền truy cập thông tin email
    Logger.log('Lỗi khi lấy thông tin người dùng: ' + e.toString());
    ui.alert('Lỗi', 'Không thể lấy thông tin người dùng. Vui lòng đảm bảo script đã được
    cấp quyền truy cập email.', ui.ButtonSet.OK);
  }
}
```

Giao diện người dùng (UI), Sidebar + HTML service

Tạo Sidebar
từ Google Sheets

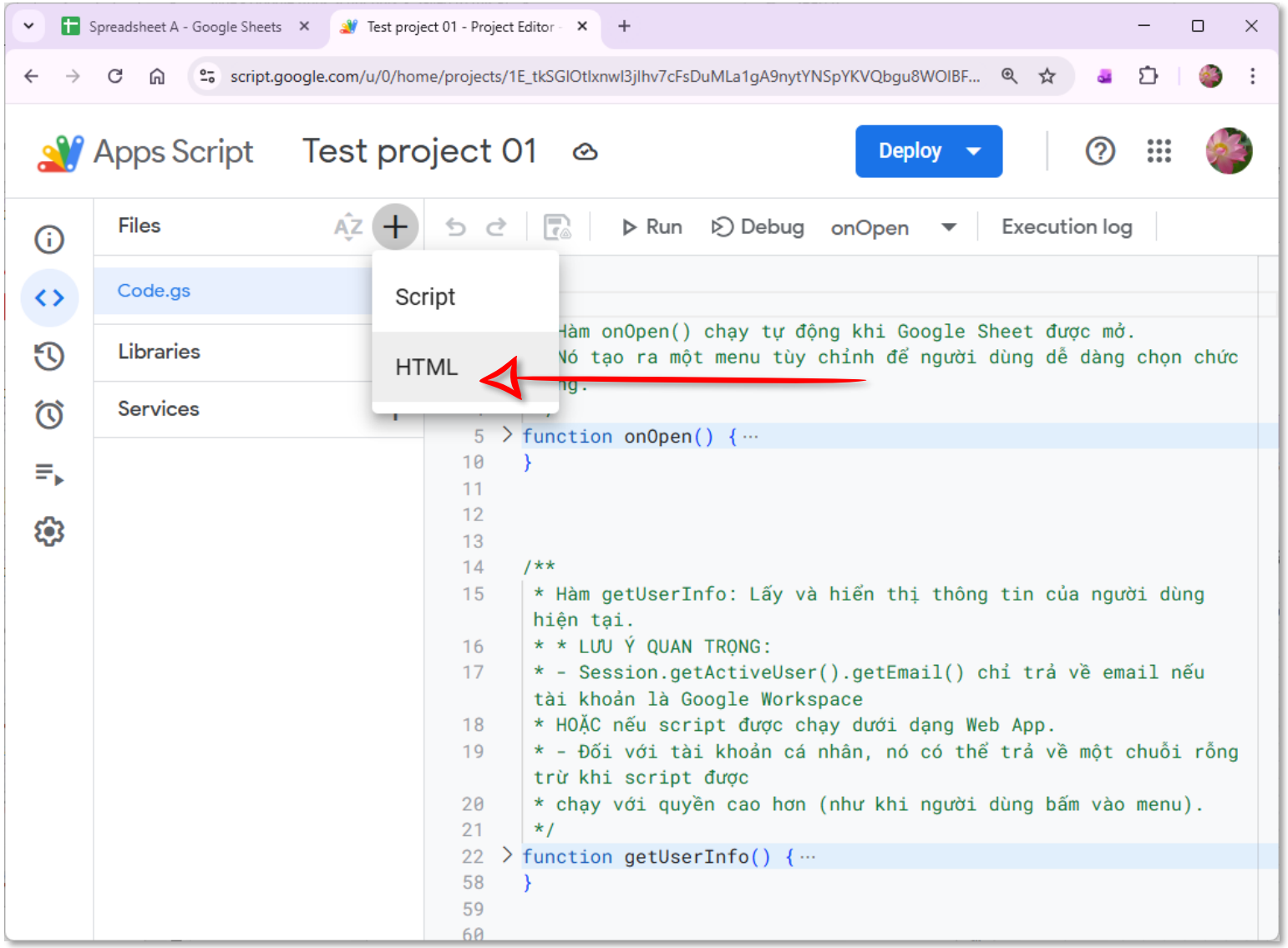


Giao diện người dùng (UI), Sidebar + HTML service

Tạo Sidebar từ Google Sheets

Nội dung gồm 2 file:

1. File Code.gs: để lưu nội dung code để thực thi
2. File Sidebar.html: để mô tả chi tiết các button (nút) trong sidebar, và liên kết với các function trong file Code.gs

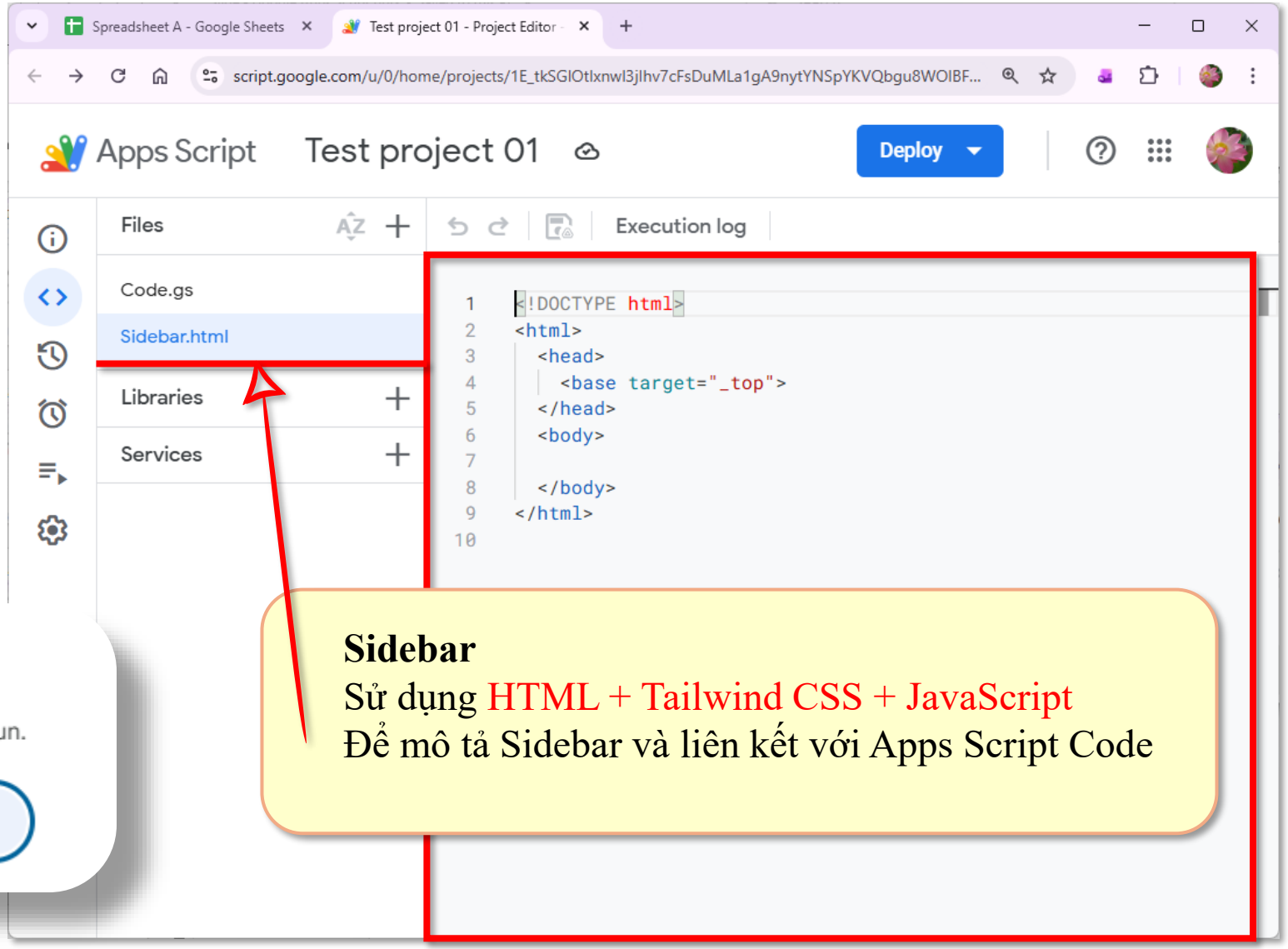


Giao diện người dùng (UI), Sidebar + HTML service

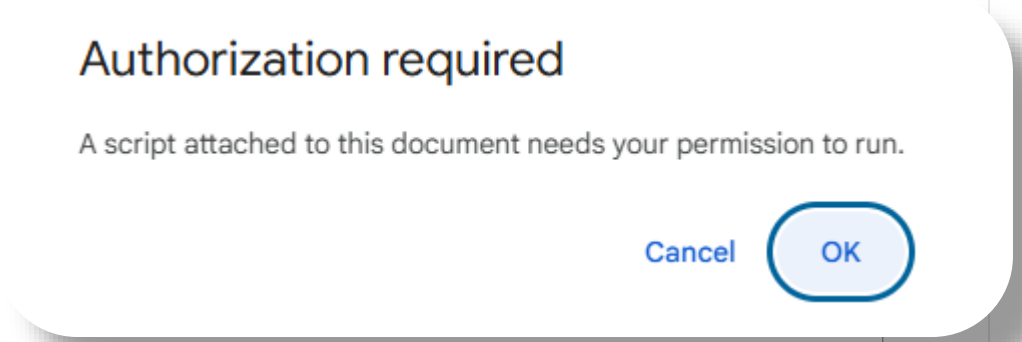
Tạo Sidebar từ Google Sheets

Nội dung gồm 2 file:

1. File Code.gs: để lưu nội dung code để thực thi
2. File Sidebar.html: để mô tả chi tiết các button (nút) trong sidebar, và liên kết với các function trong file Code.gs



Sidebar
Sử dụng HTML + Tailwind CSS + JavaScript
Để mô tả Sidebar và liên kết với Apps Script Code

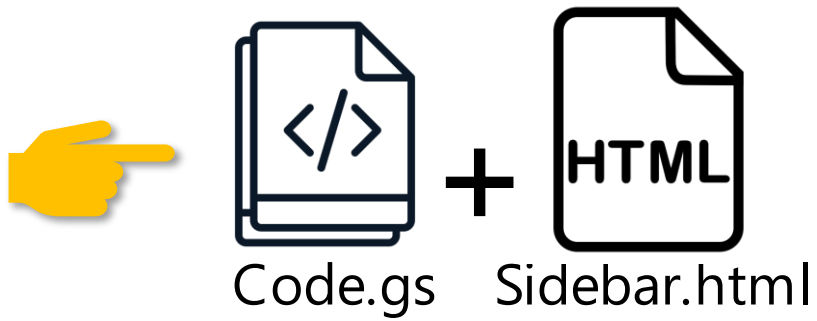


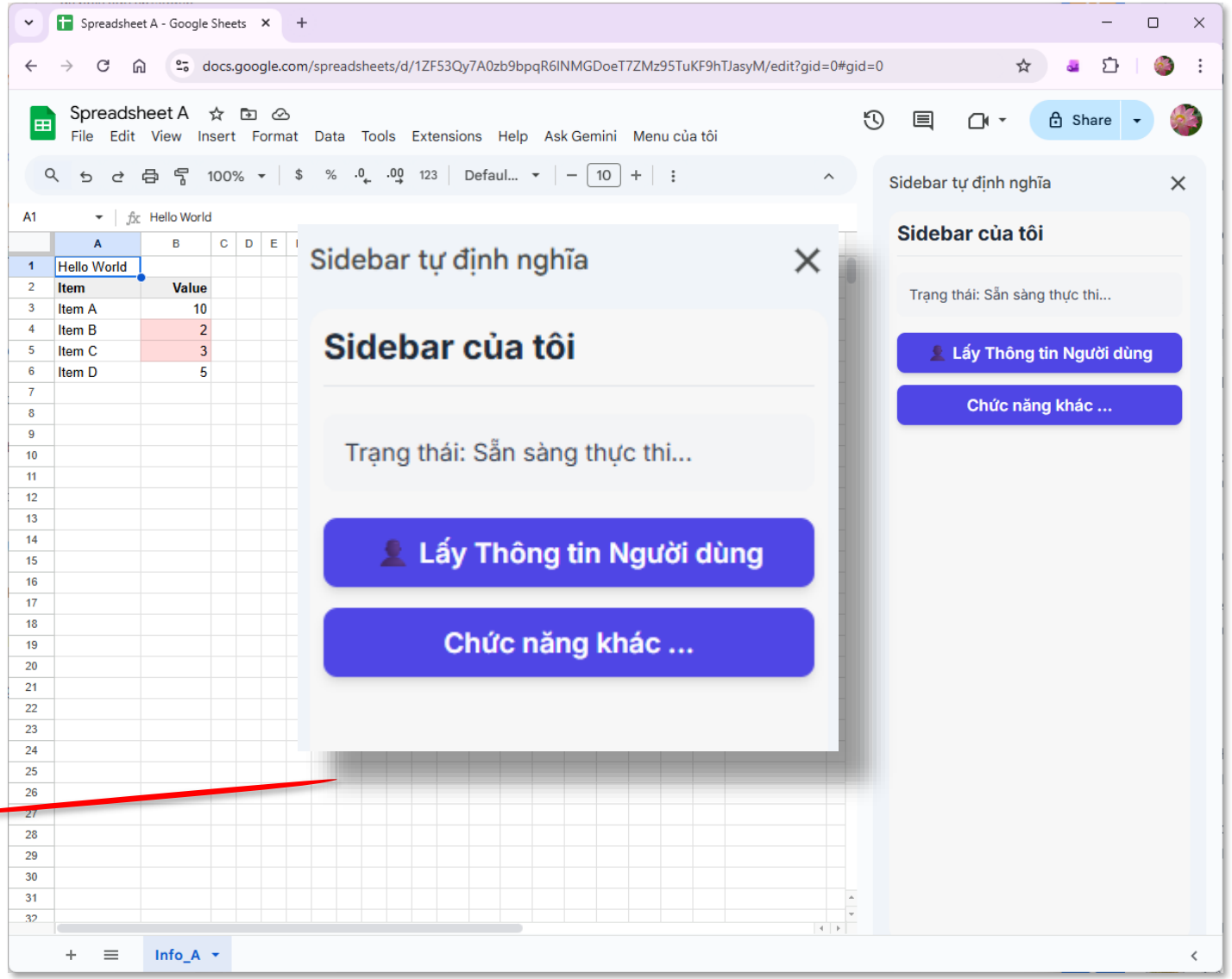
Giao diện người dùng (UI), Sidebar + HTML service

Tạo Sidebar từ Google Sheets

Nội dung gồm 2 file:

1. File Code.gs: để lưu nội dung code để thực thi
2. File Sidebar.html: để mô tả chi tiết các button (nút) trong sidebar, và liên kết với các function trong file Code.gs



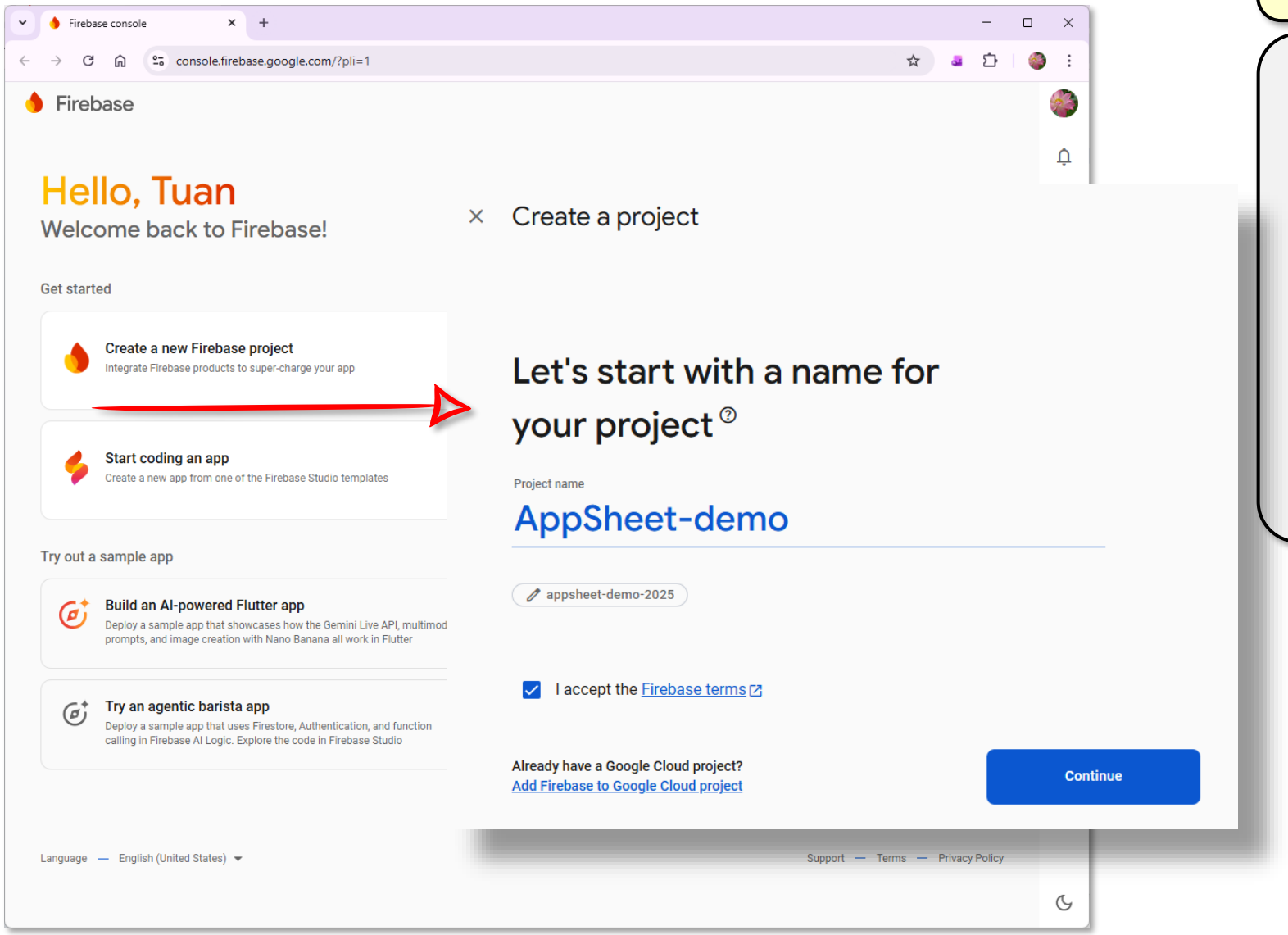


Nội dung

Phần 3

Google Apps Script với Firebase

Apps Script + Firebase



`console.firebase.google.com`

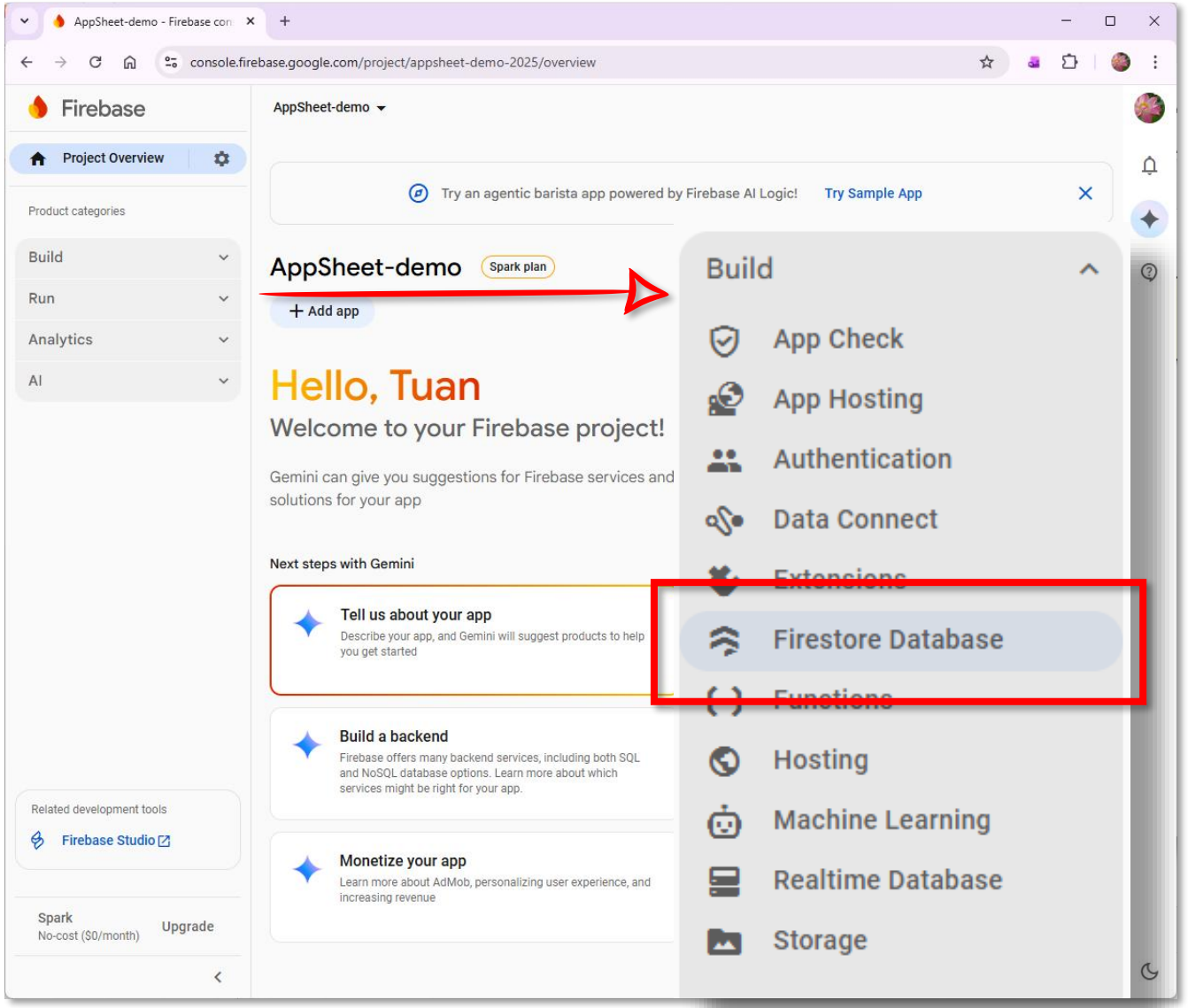
Tạo Apps Script và Firebase
với Google Sheets

Bước 1.1: Tạo Project
từ Firebase Console

Bước 1.2: Tạo Firestore Database

Bước 1.3: Lấy thông tin Project
- Project ID
- Web API key

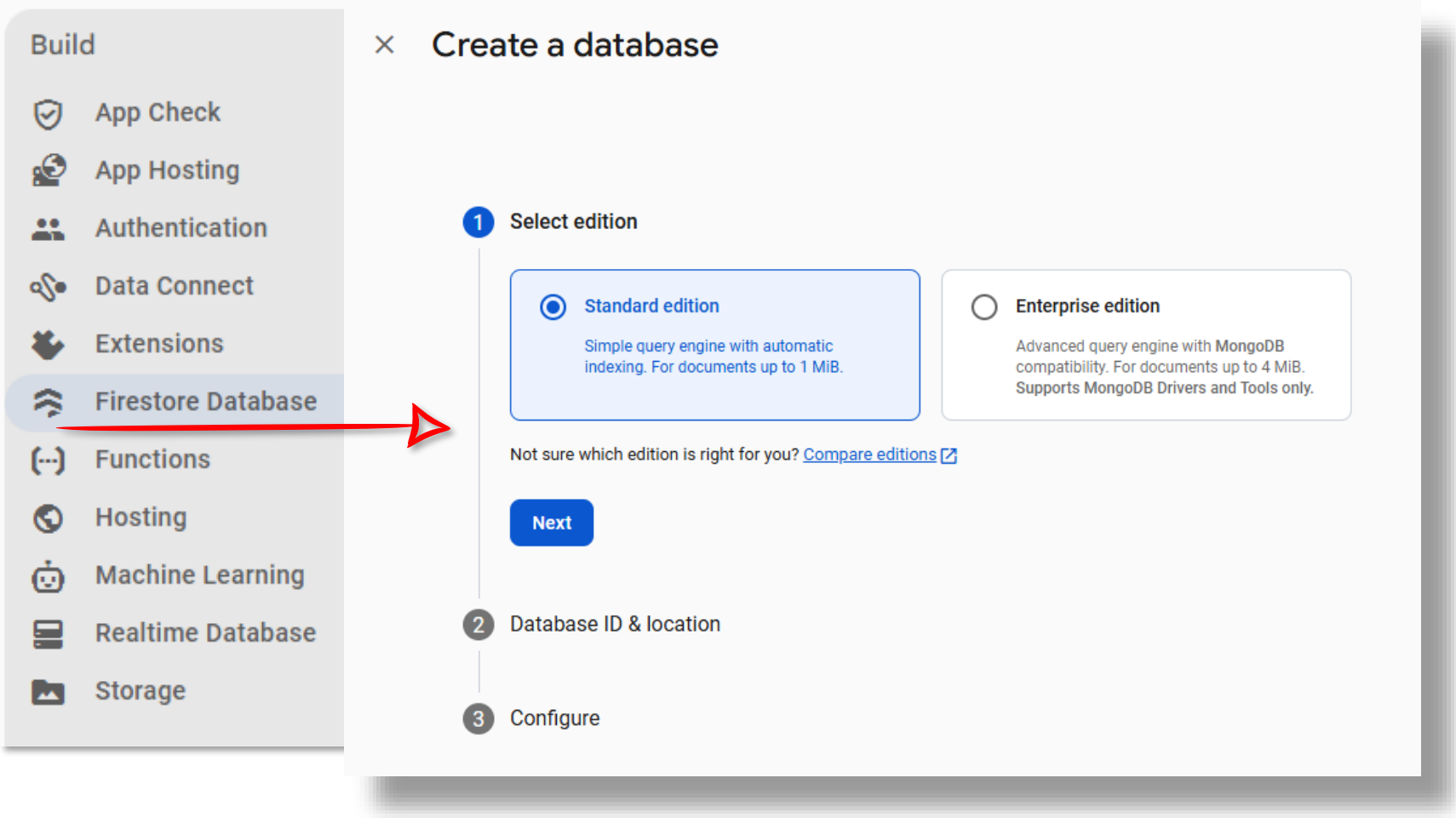
Apps Script + Firebase \ Firestore Database



Tạo Apps Script và Firebase
với Google Sheets

Bước 1.1: Tạo Project
từ Firebase Console
Bước 1.2: Tạo Firestore Database
Bước 1.3: Lấy thông tin Project
- Project ID
- Web API key

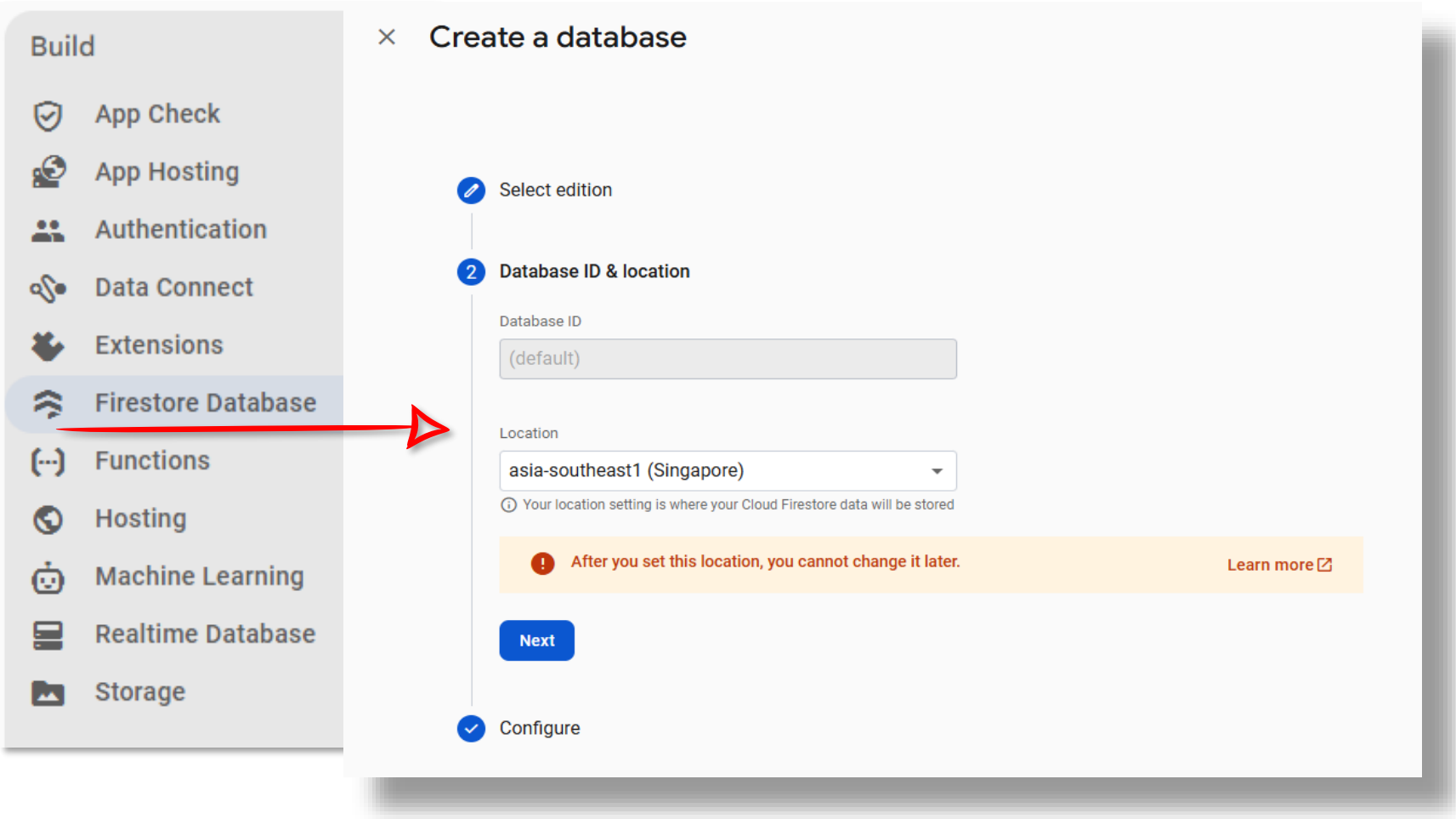
Apps Script + Firebase \ Firestore Database



Bước 1.2:
Tạo Firestore Database

a: Chọn Standard edtion
b: DatabaseID & Location
c: Database Configure

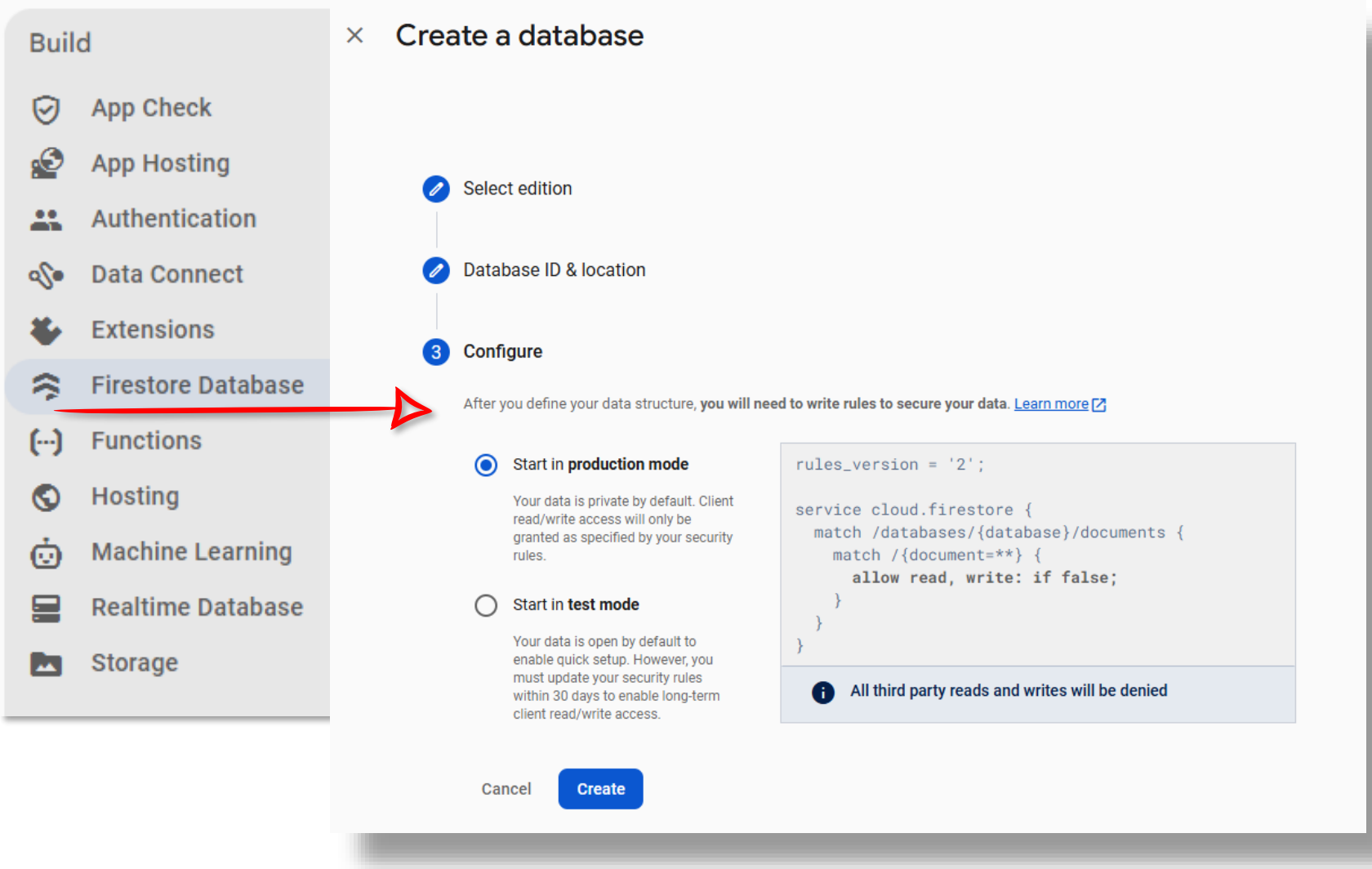
Apps Script + Firebase \ Firestore Database



Bước 1.2:
Tạo Firestore Database

a: Chọn Standard edtion
b: DatabaseID & Location
c: Database Configure

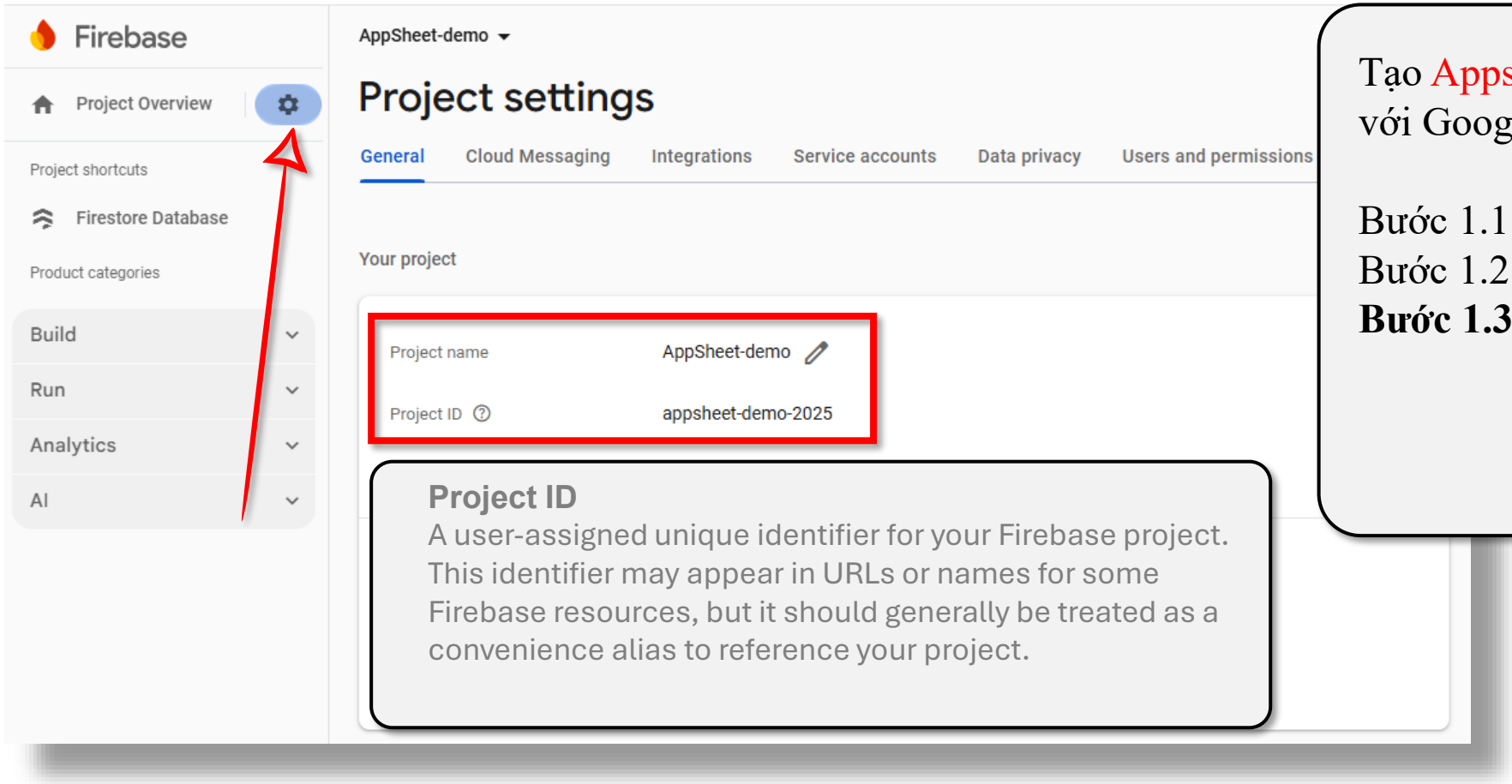
Apps Script + Firebase \ Firestore Database



Bước 1.2:
Tạo Firestore Database

a: Chọn Standard edtion
b: DatabaseID & Location
c: Database Configure

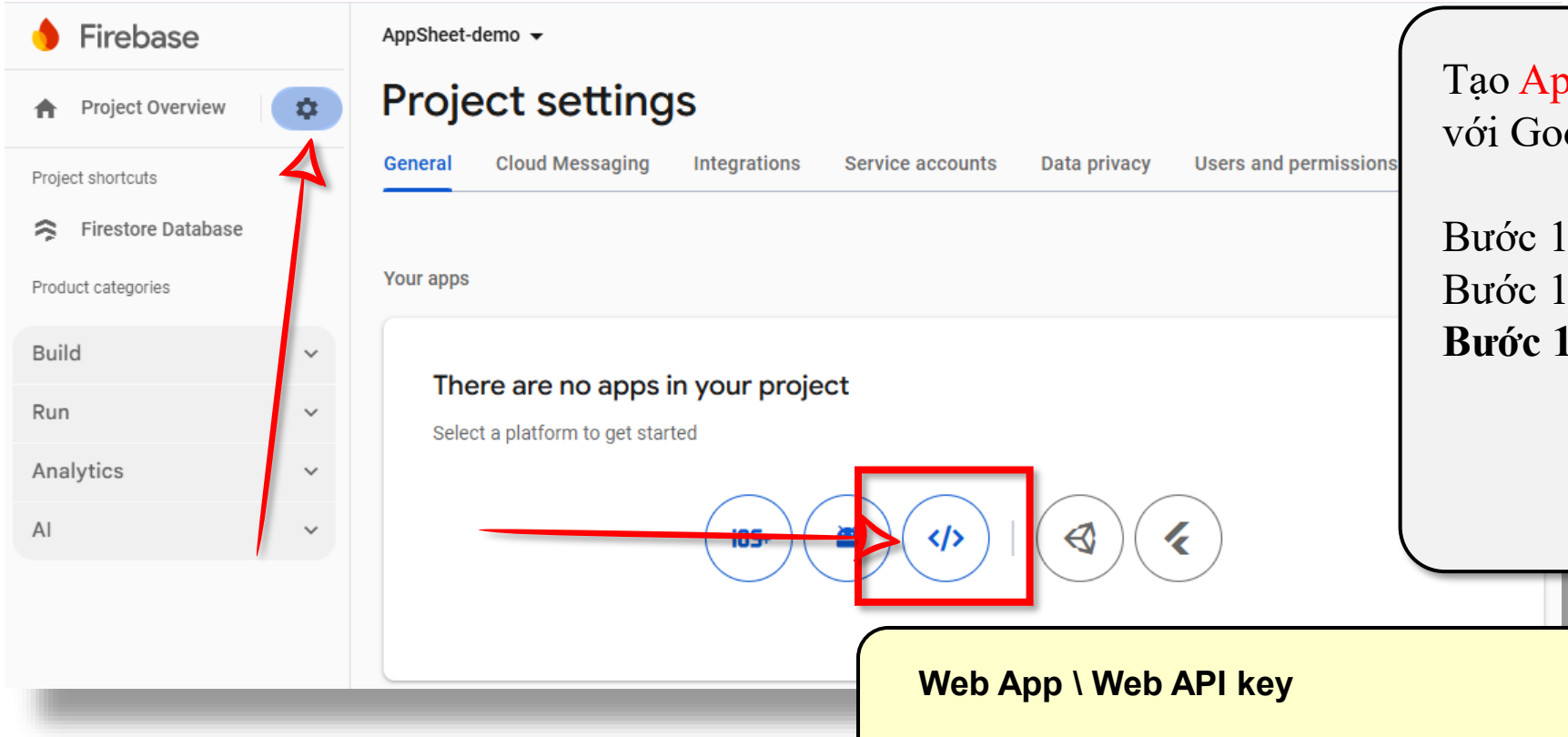
Apps Script + Firebase \ Project \ Project ID



Tạo Apps Script và Firebase
với Google Sheets

Bước 1.1: Tạo Project từ Firebase Console
Bước 1.2: Tạo Firestore Database
Bước 1.3: Lấy thông tin Project
- **Project ID**
- Web API key

Apps Script + Firebase \ Project \ Web API key



Tạo Apps Script và Firebase
với Google Sheets

Bước 1.1: Tạo Project từ Firebase Console
Bước 1.2: Tạo Firestore Database
Bước 1.3: Lấy thông tin Project
- Project ID
- **Web API key**

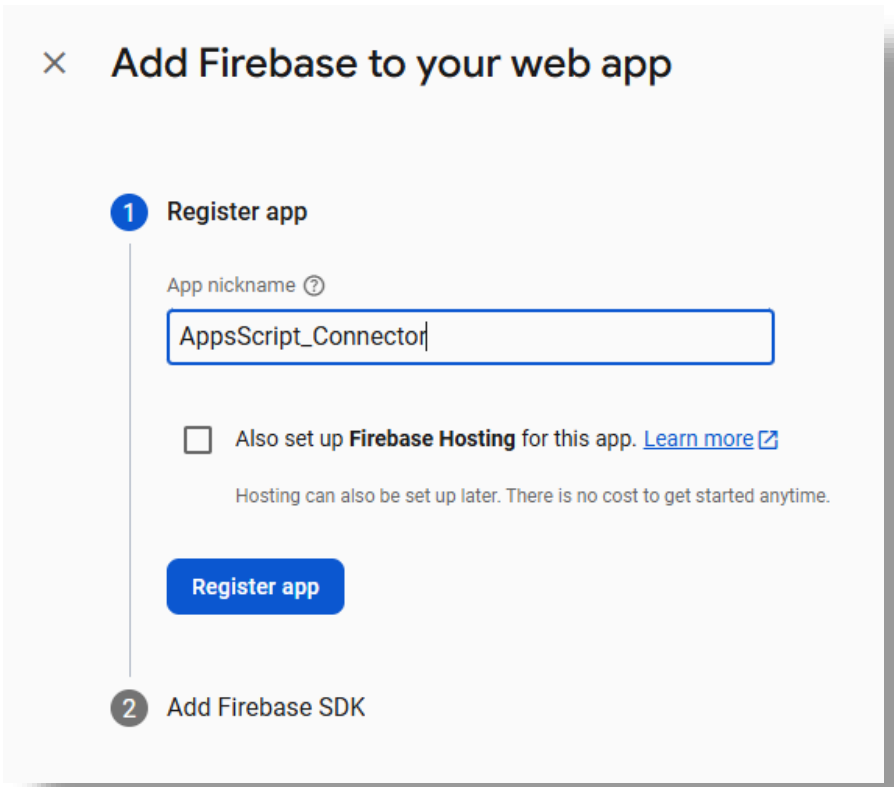
Web App \ Web API key

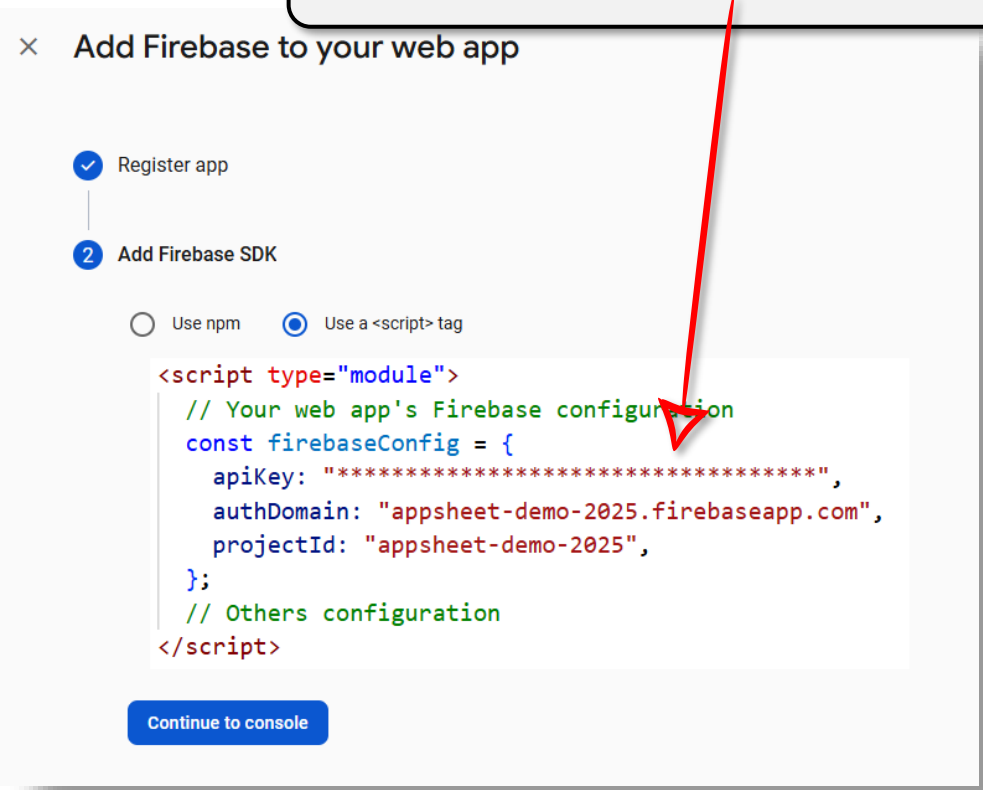
Tạo Web API key từ ứng dụng Web App

Apps Script + Firebase \ Project \ Web API key

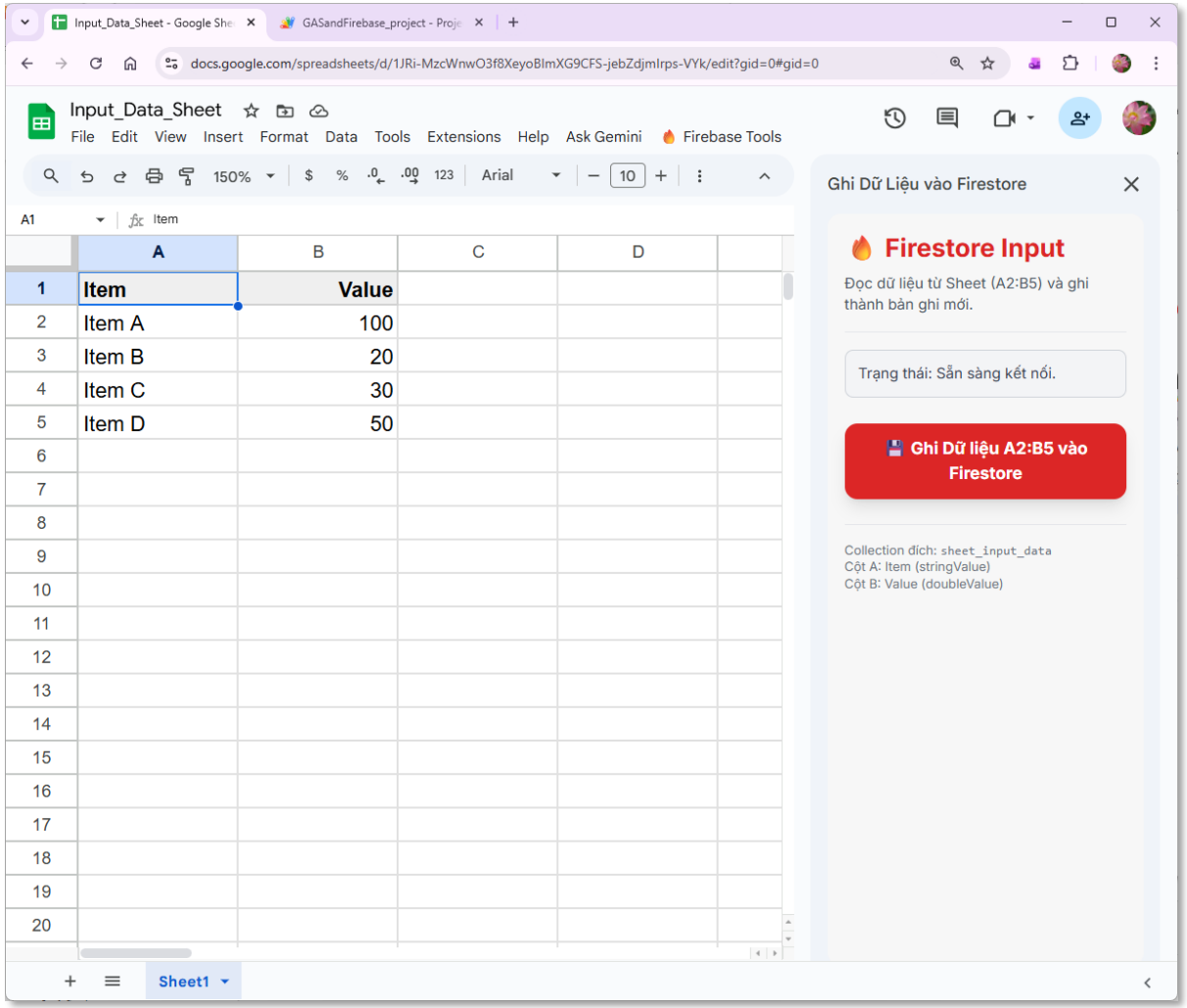
Tạo Web App

Bước 1.3: Lấy thông tin Project
- Project ID
- **Web API key**





Apps Script + Firebase \ Project \ Scripts

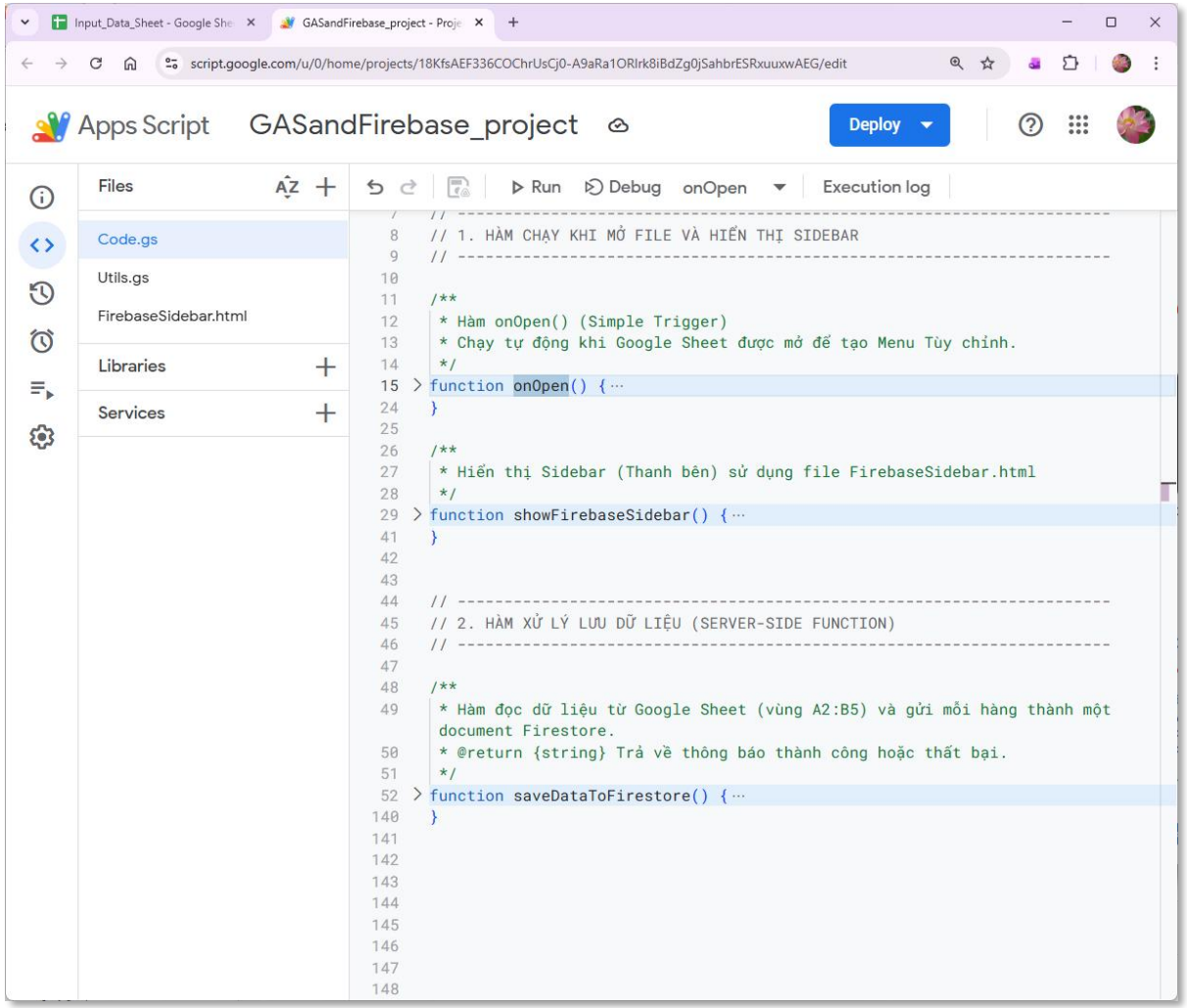


Bước 2: Tạo Sheets và các Scripts, gồm
- Các thông tin cấu hình
- Logic đọc và ghi dữ liệu vào Firestore

Tạo Sheets với nội dung minh họa như hình bên, gồm 2 cột dữ liệu

Tạo các Scripts để :
- Thêm menu “Firebase Tools”
- Sidebar với chức năng đọc và ghi dữ liệu của Sheet vào Firestore Database (đã tạo ở bước 1.2)

Apps Script + Firebase \ Project \ Scripts



Bước 2: Tạo Sheets và các Scripts, gồm
- Các thông tin cấu hình
- Logic đọc và ghi dữ liệu vào Firestore


Code.gs


Utils.gs

FirebaseSidebar.html

Các **Scripts** gồm:
- **Code.gs**: file chính, chứa logic đọc / ghi dữ liệu và các function khác
- **Utils.gs**: file lưu thông tin cấu hình chính như Project, API key, data
- **FirebaseSidebar.html**: mô tả nội dung của sidebar và nút chức năng đọc và ghi dữ liệu

Apps Script + Firebase \ Database Rule

Ví dụ Database Rule

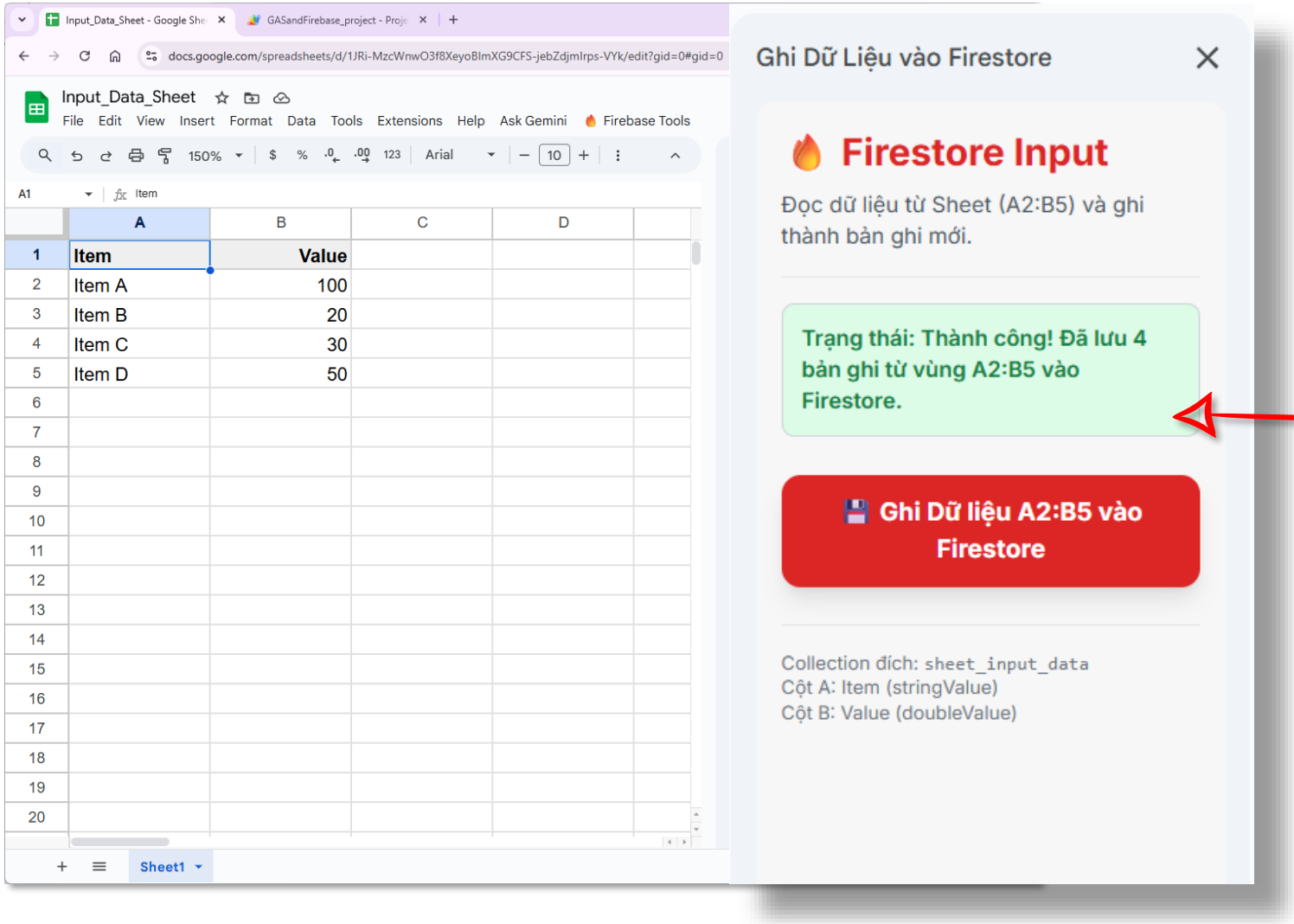
```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /sheet_input_data/{document}{
      allow read: if true;
      allow write: if true;
//      allow write: if request.auth != null;
    }
  }
}
```

Bước 3: Thiết lập Database Rule

Lưu ý:

- Rule : là quy tắc của Firebase Database thiết lập các quyền, permission, để có thể đọc / ghi dữ liệu.
- Các thiết lập này (Web API key + Rule) **phù hợp với ứng dụng nội bộ**, và đơn giản.
- Sử dụng thiết lập **Service Account**, thay cho Web API key, để nâng cao độ bảo mật hơn

Apps Script + Firebase \ Project \ Run



Bước 4: Ghi dữ liệu vào Firestore

Kết quả

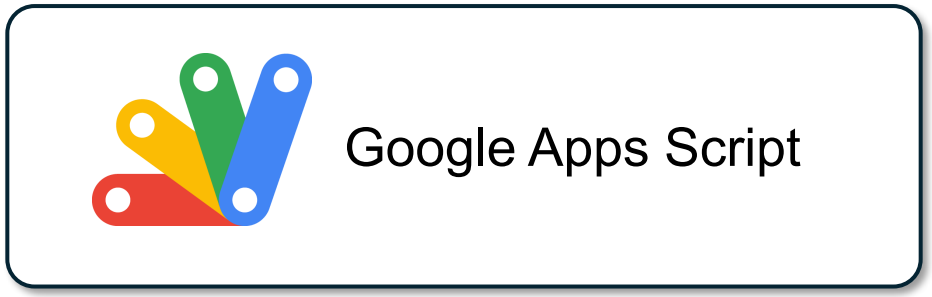
- Thực thi các Scripts thành công.
- Đã ghi dữ liệu từ Sheet vào Firestore

Tham khảo

1. Tổng quan về Google Apps Script
   https://developers.google.com/apps-script/overview?hl=vi

2. Khóa học “Google Apps Script Complete Course Beginner to Advanced”
   https://fpl.udemy.com/course/apps-script-course/learn/lecture/10208194#overview

3. Các ví dụ mẫu
   https://developers.google.com/apps-script/samples?hl=vi



TRÂN TRỌNG CẢM ƠN

11-2025
Nhóm thực hiện
Thân Hoàng Lộc
Trần Anh Tuấn
…