

# Wysyłanie komend z systemu Grenton do Asystenta Google za pomocą Home Assistant

W tym tutorialu przedstawiona została możliwość wysyłania komend z systemu Grenton do Asystenta Google za pomocą Gate Http oraz Home Assistant.

Przedstawiona konfiguracja została wykonana na:

- GATE HTTP w wersji `1.1.0 (build 2034C)`,
- OM w wersji `v1.6.0 (build 214801)`,
- Home Assistant w wersji `2022.2.8`.

## 1. Instalacja Home Assistant

Szczegółowe informacje jak zainstalować Home Assistant na różnych platformach można znaleźć na stronie: <https://www.home-assistant.io/installation/>.

Zalecane jest ustawienie/zarezerwowanie adresu IP dla Home Assistant w sieci lokalnej, aby urządzenie zawsze miało ten sam adres IP. Należy to zrobić w ustawieniach routera (szczegóły w instrukcji routera). Przykładowo dla routera TP-link rezerwowanie znajduje się w zakładce `DHCP->Address Reservation`.

## 2. Konfiguracja połączenia z Google Assistant

W pierwszej kolejności należy zainstalować dodatek Assistant Relay w Home Assistant. Aby to zrobić należy wejść w `Konfiguracja -> Dodatki, kopie zapasowe oraz Supervisor -> SKLEP Z DODATKAMI`, następnie w menu w prawym górnym rogu ekranu wybrać `Repozytoria` i dodać <https://github.com/Apipal69/Assistant-Relay-for-Hassio>.


Po dodaniu należy zainstalować oraz uruchomić dodatek `Assistant Relay`.

## Assistant Relay

Current version: 0.7.4 ([Lista zmian](#))

5 Ocena

Send (broadcast) commands to your Google Assistant via REST command.  
Odwiedź stronę [Assistant Relay](#), aby uzyskać więcej szczegółów.



Uruchamianie przy starcie

Uruchom dodatek podczas startu systemu

Obserwator

Ponowne uruchomienie dodatku, jeśli się zawiesi

Automatyczna aktualizacja

Automatycznie aktualizuj dodatek, gdy pojawi się nowa wersja

☒

☐

☐

☐

Nazwa hosta

6a3d001f-assistant-relay

Użycie procesora przez dodatek

0.3 %

Użycie pamięci przez dodatek

2.3 %


ZATRZYMAJ

URUCHOM PONOWNIE

OTWÓRZ INTERFEJS UŻYTKOWNIKA

ODINSTALUJ

Po uruchomieniu panel dodatku dostępny będzie pod tym samym adresem IP, na porcie `3000`, przykładowo `http://homeassistant.local:3000/`.



1 Setting up your Project

2 Adding Client Secrets

3 Submit Token

## Introduction

To get started with Assistant Relay, you first need to setup a project in the Google Console to integrate this. You will need to repeat these steps for every Google Account you want to add to Assistant Relay.

Note: Make sure you are signed into the Google Account you want this to work with

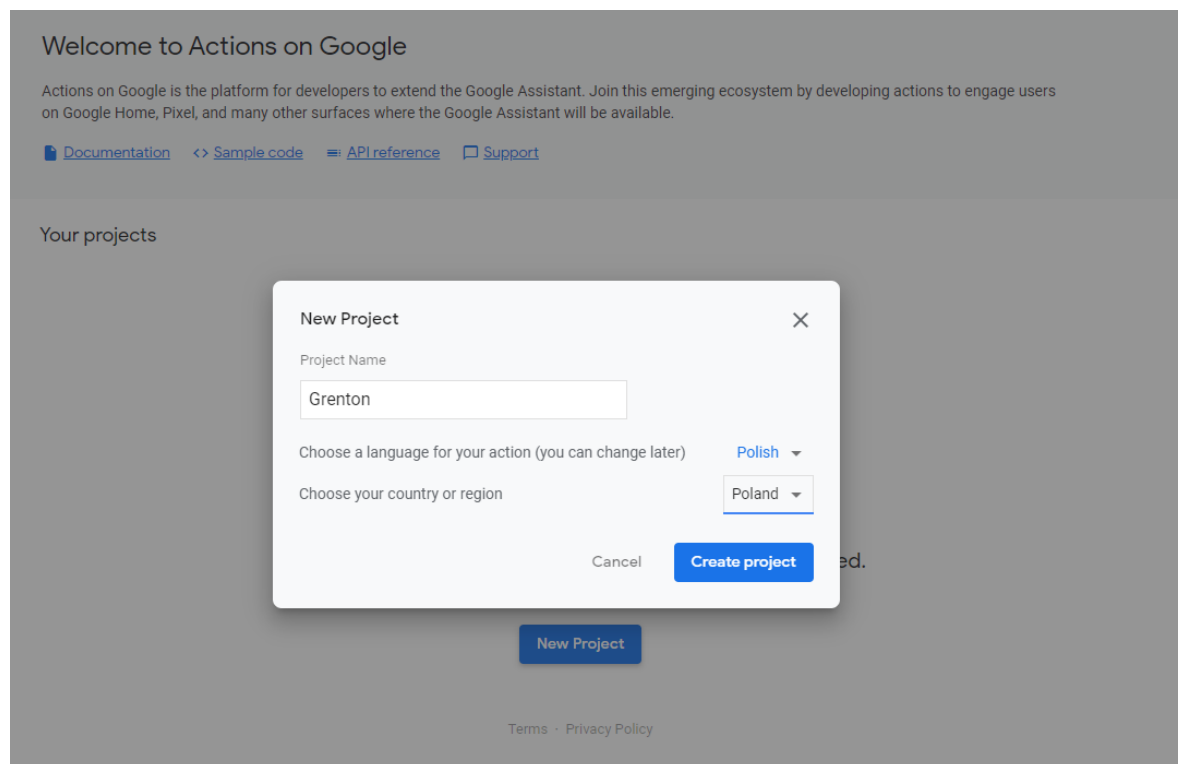
## Setup Steps

- Follow [this guide](#) and come back once Step 5 is complete
- Now the Assistant API is enabled, click [Credentials](#) in the left hand menu
- On this page, select [Create Credentials](#), and choose [OAuth client ID](#)
- Select [Other](#) as the application type and give it a name such as [Assistant Relay Client](#). Then click create
- You'll get a popup, but just click OK to close this
- Back on the Credentials page, click the download icon next to the new credential you made. Save it somewhere safe
- Now click [Configure Consent Screen](#)
- Give the application a name such as Assistant Relay, add an email under Support Email and click Save

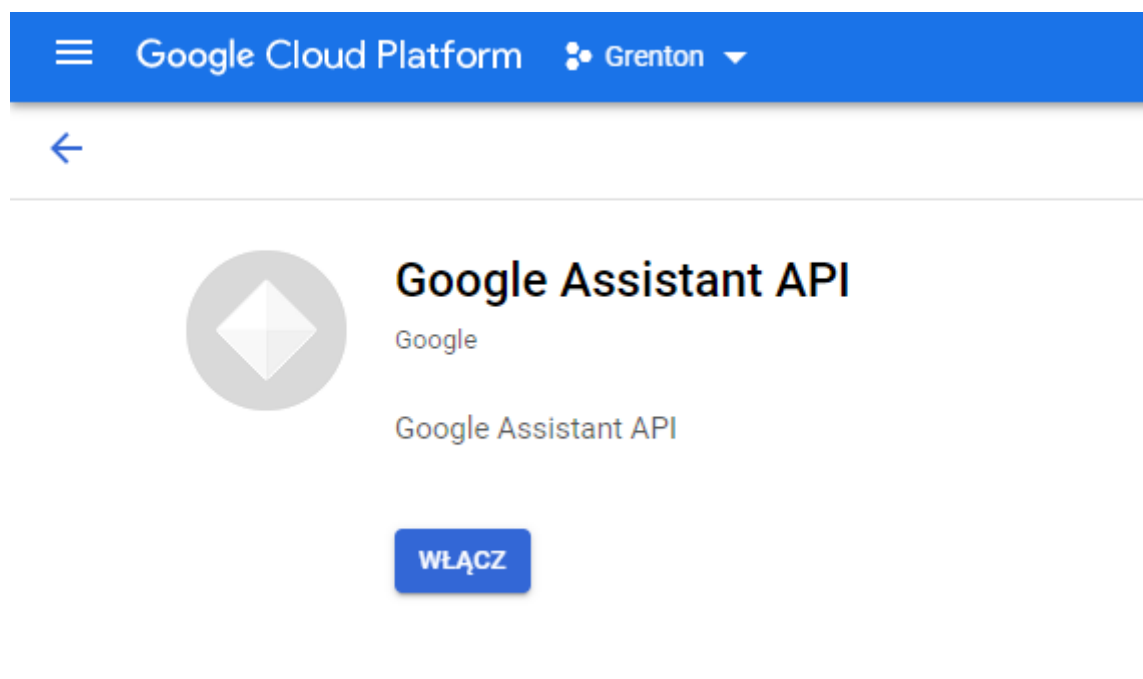
Note: You can repeat these steps for each Google Account you wish to configure Assistant Relay for

Next

Kolejnym krokiem będzie przejście do <https://console.actions.google.com/> i stworzenie projektu.



Następnym krokiem będzie włączenie Google Assistant API. W tym celu należy przejść do <https://console.developers.google.com/apis/api/embeddedassistant.googleapis.com/overview>, upewnić się, że wybrany jest właściwy projekt i wybrać `Włącz`.



Po włączeniu Google Assistant API pojawi się okno `Interfejsy API i usługi`. Należy przejść do `Ekran zgody OAuth`, wybrać typ `Zewnętrzny` i `Utwórz`.

Google Cloud Platform

Grenton

Szukaj Produkty, zas

API

Interfejsy API i usługi

Włączone interfejsy API i usł...

Biblioteka

Dane logowania

**Ekran zgody OAuth**

Weryfikacja domen

Zgody dotyczące użytkow...

Wybierz sposób, w jaki chcesz skonfigurować i zarejestrować aplikację, w tym użytkowników docelowych. Możesz powiązać ze swoim projektem tylko jedną aplikację.

User Type

☐ Wewnętrzny

Dostęp tylko dla użytkowników z Twojej organizacji. Nie musisz przysyłać aplikacji do weryfikacji. [Więcej informacji o typie użytkownika](#)

☒ **Zewnętrzny**

Każdy użytkownik mający konto Google może uzyskać dostęp. Twoja aplikacja będzie uruchamiana w trybie testowym i będzie dostępna tylko dla użytkowników, których dodasz do listy testowych. Gdy Twoja aplikacja będzie gotowa do przeniesienia do środowiska produkcyjnego, może być konieczne przesłanie jej do weryfikacji. [Więcej informacji o typie użytkownika](#)

UTWÓRZ

[Podziel się z nami swoją opinią](#) o tym sposobie obsługi protokołu OAuth.

W kolejnym oknie większość pól będzie opcjonalne. Należy uzupełnić nazwę aplikacji, np. `Assistant Relay`, oraz wpisać adres email w pola wymagane i wybrać `ZAPISZ I KONTYNUUJ` aż do zakładki `Podsumowanie` i wybrać `POWRÓT DO PANELU`.

Na utworzonej stronie aplikacji należy dodać użytkownika testowego - wpisać adres konta Google.

API

Interfejsy API i usługi

Włączone interfejsy API i usł...

Biblioteka

Dane logowania

**Ekran zgody OAuth**

Weryfikacja domen

Zgody dotyczące użytkow...

Stan publikacji

Testuję

OPUBLIKUJ APLIKACJĘ

Typ użytkowników

Zewnętrzni

ZMIEN NA WEWNĘTRZNY

Limit użytkowników OAuth

Gdy stan publikowania jest ustawiony na „Testowanie”, dostęp do aplikacji mają tylko użytkownicy testowi. Dozwolony limit użytkowników przed weryfikacją aplikacji to 100. Limit ten jest liczony w ciągu całego cyklu życia aplikacji. [Więcej informacji](#)

0 użytkowników (testowi: 0, inni: 0) / limit liczby użytkowników: 100

Użytkownicy testowi

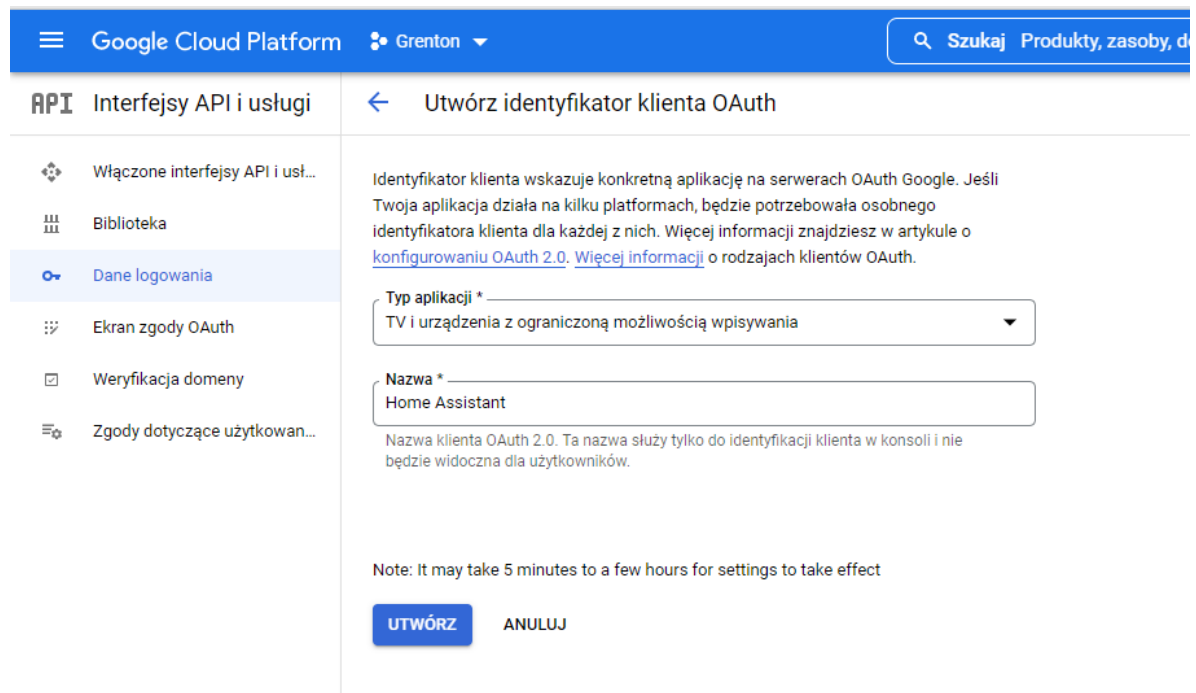
+ ADD USERS

Filtruj Wpisz nazwę lub wartość właściwości

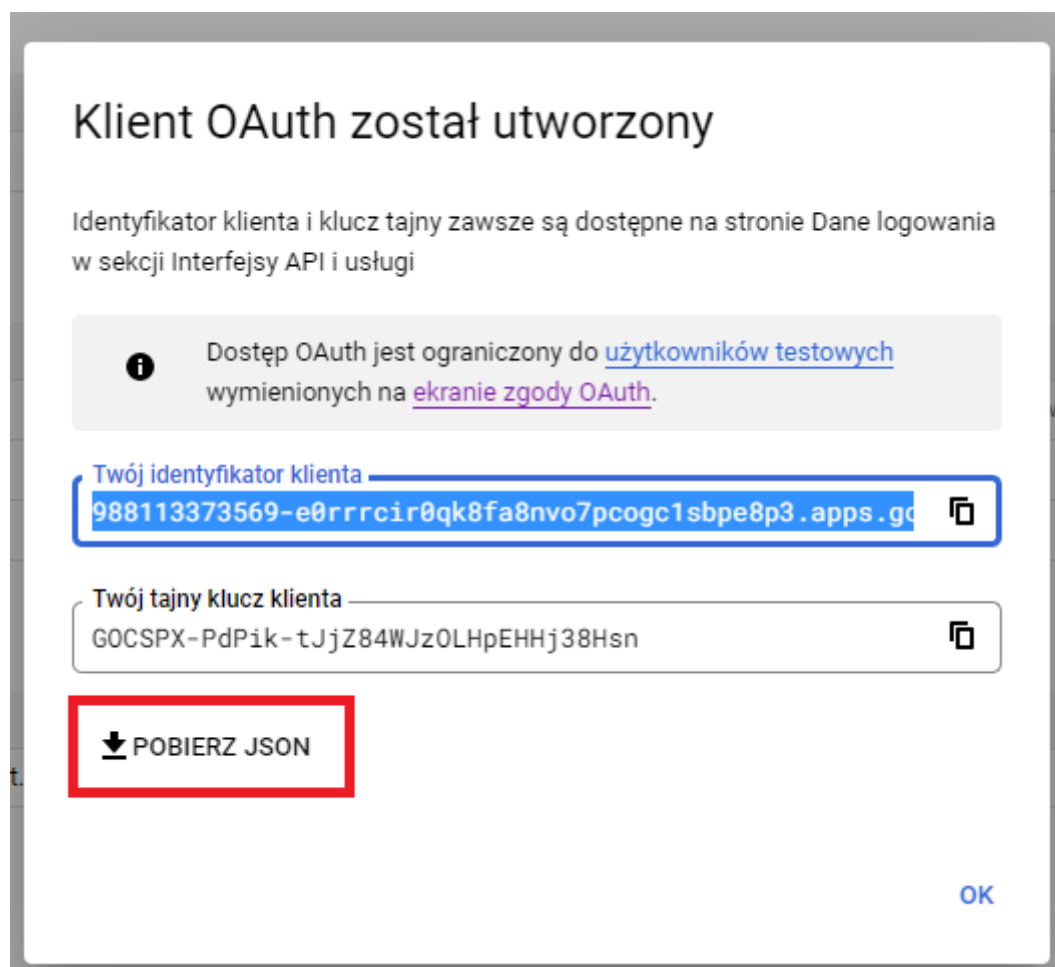
Informacje o użytkownikach

Brak wierszy do wyświetlenia

Po dodaniu użytkownika testowego należy przejść do zakładki `Dane logowania` i wybrać `+ UTWÓRZ` `DANE LOGOWANIA` (Identyfikator klienta OAuth). Jako `Typ aplikacji` należy wybrać `TV i urządzenia z ograniczoną możliwością wpisywania` oraz nazwać identyfikator, np. `Home Assistant`, i wybrać `UTWÓRZ`.



Po utworzeniu identyfikatora należy pobrać plik `client_secret_XXX.json`.



Po pobraniu pliku należy wrócić do Home Assistant, otworzyć ponownie panel dodatku pod adresem `http://homeassistant.local:3000/` i wybrać `Next`, następnie uzupełnić pole `Users Name` oraz załadować pobrany plik `client_secret_XXX.json` i wybrać `Next`.

Setting up your Project — 2 Adding Client Secrets — 3 Submit Token

### Setting up your account

With your client secrets, you can import them into Assistant Relay below

\* Users Name:

@ client\_secret\_988113373569-e0rrcir0qk8fa8nvo7pcogc1sbpe8p3.apps.googleusercontent.com.json

Pojawi się okno, w którym należy wybrać swoje konto Google oraz kontynuować akceptowanie aplikacji. W ostatniej części pojawi się kod autoryzacji, który należy skopiować i wkleić w polu `Submit Token` na stronie dodatku i wybrać `Finish`.

Setting up your Project — Adding Client Secrets — 3 Submit Token

### Setting up your accounts

Now that you have the client secrets downloaded, you can import them into Assistant Relay below

\* Auth Code:

W tym momencie, jeśli dla danego konta przypisany jest Głośnik Google, będzie można usłyszeć komunikat o połączeniu. Aby przetestować wysłanie polecenia, można przejść do zakładki `Sandbox`, wpisać wybraną wcześniej nazwę użytkownika, zaznaczyć `Broadcast = Yes`, przykładowo wpisać komendę `Hello` i zatwierdzić.

Command Successful

Home Configuration Add New User Remove User **Sandbox** About

User:

Preset:

Command:

Broadcast: ☒ Yes ☐ No

Converse: ☐ Yes ☒ No

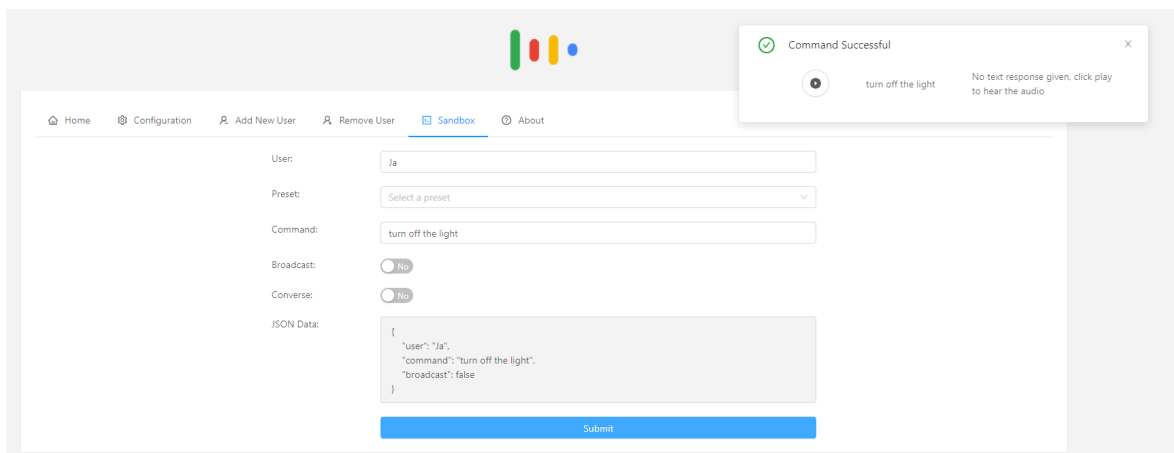
JSON Data: 

```
{
  "user": "Ja",
  "command": "Hello",
  "broadcast": true
}
```

Jeśli połączenie będzie prawidłowe, pojawi się komunikat `Command Successful` a na głośniku Google usłyszymy przychodzący komunikat.

Jeśli w tym miejscu wystąpią problemy z połączeniem, należy uruchomić ponownie Home Assistant. Jeśli problem nadal występuje, należy przejść do dodatku `Assistant Relay` i sprawdzić zawartość logów.

Jeśli wszystko działa poprawnie, możliwe jest wysłanie dowolnej możliwej komendy dostępnej w Asystencie Google. Aby przetestować wywołanie komendy należy uzupełnić nazwę użytkownika, wpisać dowolną komendę oraz odznaczyć `Broadcast`.



The screenshot shows the Assistant Relay web interface. At the top, there is a navigation bar with links: Home, Configuration, Add New User, Remove User, Sandbox (active), and About. The main form has the following fields: User (text input with 'Ja'), Preset (dropdown menu with 'Select a preset'), Command (text input with 'turn off the light'), Broadcast (radio button with 'No' selected), Converse (radio button with 'No' selected), and JSON Data (text area with a JSON object: { "user": "Ja", "command": "turn off the light", "broadcast": false }). A blue 'Submit' button is at the bottom. A toast notification in the top right corner says 'Command Successful' with a play button icon and the text 'turn off the light' and 'No text response given, click play to hear the audio'.

Po wywołaniu komendy, Asystent Google zgasił światło na urządzeniach dodanych do Google Home.

### 3. Konfiguracja w Grenton

Na Gate Http należy utworzyć obiekt wirtualny `HttpRequest` oraz skonfigurować go w następujący sposób:

- `Host` - ustawić adres dla serwera Home Assistant wraz z portem 3000,

Należy ustawić port ustawiony dla dodatku Assistant Relay, domyślnie 3000. Przykład url:  
`http://192.168.0.114:3000`

- `Path` - `"/assistant"`,
- `Method` - `POST`,
- `RequestType` , `ResponseType` - `JSON`.

### Właściwości obiektu

Nazwa: 
Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
Host	<input type="text" value="http://192.168.0.114:3000"/>	<input type="text" value="http://192.168.0.114:3000"/>	string	
Path	<input type="text" value="/assistant"/>	<input type="text" value="/assistant"/>	string	
QueryStringParams	<input type="text" value="-"/>	<input type="text" value="\z"/>	string	
Method	<input type="text" value="POST"/>	<input type="text" value="POST"/>	string	
Timeout	<input type="text" value="5"/>	<input type="text" value="5"/>	s	[1-255]
RequestType	<input type="text" value="2"/>	<input type="text" value="JSON"/>	-	0,1,2,3,4,5
ResponseType	<input type="text" value="2"/>	<input type="text" value="JSON"/>	-	0,1,2,3,4,5
RequestHeaders	<input type="text" value="-"/>	<input type="text" value="\z"/>	string	
RequestBody	<input type="text" value="-"/>	<input type="text" value="\z"/>	string	
ResponseBody	<input type="text" value="-"/>	<input type="text" value="\z"/>	string	
StatusCode	<input type="text" value="200"/>		-	

☒ Auto odświeżanie

Następnie należy utworzyć skrypt inicjujący wysłanie komendy do Asystenta Google:

```

local ga_command = "zgaś wszystkie światła" -- dowolna komenda wywoływana w
Asystencie Google

local eventJson = {
    user = "Ja",
    command = ga_command
}

GATE->GoogleAssistantCommand->SetRequestBody(eventJson)
GATE->GoogleAssistantCommand->SendRequest()

```

Parametr `user` zawiera ustawioną wcześniej nazwę użytkownika, natomiast `command` zawiera dowolną komendę, jaką chcemy wywołać w Asystencie Google.

GOTOWE!

W tym miejscu po wysłaniu konfiguracji, wywołanie skryptu spowoduje wywołanie żądanej komendy w Asystencie Google.

Przykładowe komendy:

- `Uruchom telewizor w Salonie`



- Zgaś światło w Sypialni

Google nie zezwala na działanie poleceń multimedialnych, więc komenda odtworzenia muzyki nie będzie działać.

Przykładowa odpowiedź po wysłaniu komendy wygląda następująco:

```
{
  "response": "",
  "audio": "/server/audio?v=1645697841613",
  "success": true
}
```

Dźwięk odpowiedzi Asystenta Google można odtworzyć wywołując przykładowo `http://192.168.0.114:3000/server/audio?v=1645697841613`.

Jeśli pojawi się odpowiedź tekstowa z asystenta, zostanie wyświetlona w pozycji `response`.

Możliwe jest również odtworzenie wiadomości głosowej na wszystkich lub wybranych urządzeniach Google Home lub Nest w całym domu. W tym celu komendę należy poprzedzić:

- Broadcast <message>
- Tell everyone, <message>
- Broadcast to the living room, it's time to start homework.

Przykład:

```
local ga_command = "tell evryone, Dinner time" -- wiadomość odtworzona na
głośnikach Google

local eventJson = {
  user = "Ja",
  command = ga_command
}

GATE->GoogleAssistantCommand->SetRequestBody(eventJson)
GATE->GoogleAssistantCommand->SendRequest()
```

Aktualnie możliwe jest wywołanie dowolnych lub zaprogramowanych komunikatów wyłącznie w języku angielskim.

Listę zaprogramowanych komunikatów można znaleźć pod linkiem: <https://support.google.com/googlenest/answer/7531913?hl=en&co=GENIE.Platform%3DAndroid>

## 4. rest\_command po stronie Home Assistant

Przykład implementacji komendy rest w Home Assistant w pliku `configuration.yaml`:

```
rest_command:
  assistant_relay:
    url: http://192.168.0.114:3000/assistant
    method: POST
    content_type: "application/json"
    payload: '{"command": "{{ google_command }}", "user": "Ja"}'
```

Należy uzupełnić `url` o właściwy adres ip, oraz `user` o prawidłową nazwę użytkownika.