multimedia-on-ip

Jean-Michel REMEUR & Thomas LACAZE - EFREI M1 APP LSI

Code partie I et II en Typescript

```
require("dotenv").config();
import { SessionsClient } from '@google-cloud/dialogflow';
import { v4 } from 'uuid';
const RainbowSDK = require('rainbow-node-sdk');

const options = {
  rainbow: {
    host: "sandbox"
  },
  credentials: {
    login: process.env.LOGIN,
    password: process.env.PASSWORD
  },
  application: {
    appID: process.env.ID,
    appSecret: process.env.APP_SECRET
  },
  logs: {
    enableConsoleLogs: true,
    enableFileLogs: false,
    file: {
      path: '/var/tmp/rainbowsdk/',
      level: 'debug'
    }
  },
  im: {
    sendReadReceipt: true
  }
}

const projectId = process.env.PROJECT_ID;
const credentials_file_path = './small_talk.json';
const sessionId = v4();

const rainbowSDK = new RainbowSDK(options);

rainbowSDK.start().then( () => {
  console.log("Started...");
});

rainbowSDK.events.on('rainbow_onmessagereceived', async (msg) => {

  // Partie 1
  // console.log(msg);
  // rainbowSDK.im.sendMessageToJid("Message bien reçu", msg.fromJid);
  // rainbowSDK.im.sendMessageToJid(msg.content, msg.fromJid);

  // Partie 2
  const response = await runSample(projectId, msg.content);
  rainbowSDK.im.sendMessageToJid(response, msg.fromJid);
});

async function runSample(projectId: string, query: string) {
  const languageCode = "en-US";

  const sessionClient = new SessionsClient({
    projectId,
    keyFilename: credentials_file_path,
  });

  const sessionPath = sessionClient.projectAgentSessionPath(projectId,
sessionId);

  const request = {
    session: sessionPath,
    queryInput: {
      text: {
        text: query,
        languageCode: languageCode,
      },
    },
  };

  const responses = await sessionClient.detectIntent(request);

  const result = responses[0].queryResult;

  console.log(`Query: ${result.queryText}`);
  console.log(`Response: ${result.fulfillmentText}`);

  return result.fulfillmentText;
}
```

Package.json

```
{
  "name": "multimedia-on-ip",
  "version": "1.0.0",
  "description": "",
  "main": "src/index.ts",
  "scripts": {
    "start": "tsc && node dist/index.js",
    "build": "tsc"
  },
  "author": "JM REMEUR, THOMAS LACAZE",
  "license": "MIT",
  "dependencies": {
    "@google-cloud/dialogflow": "^3.3.0",
    "@types/uuid": "^8.3.0",
    "dotenv": "^8.2.0",
    "rainbow-node-sdk": "^1.84.0",
    "uuid": "^8.3.2"
  },
  "devDependencies": {
    "@types/node": "^14.14.31",
    "typescript": "^4.2.2"
  }
}
```

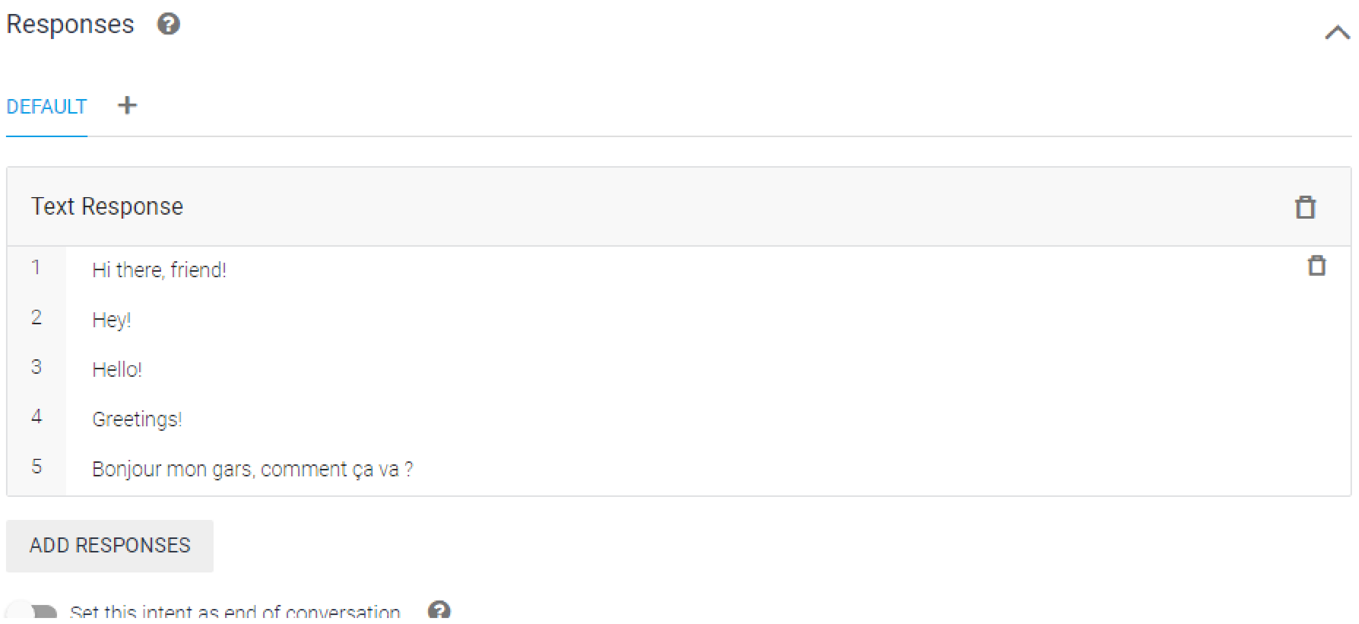
tsconfig.json

```
{
  "compilerOptions": {
    "module": "commonjs",
    "target": "es2017",
    "noImplicitAny": false,
    "sourceMap": true,
    "outDir": "dist",
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "importHelpers": true,
    "forceConsistentCasingInFileNames": true,
    "lib": [
      "es2017",
      "esnext.asynciterable"
    ],
    "moduleResolution": "node",
    "types": [ "node" ],
  },
  "exclude": [
    "node_modules"
    ]
}
```

.env

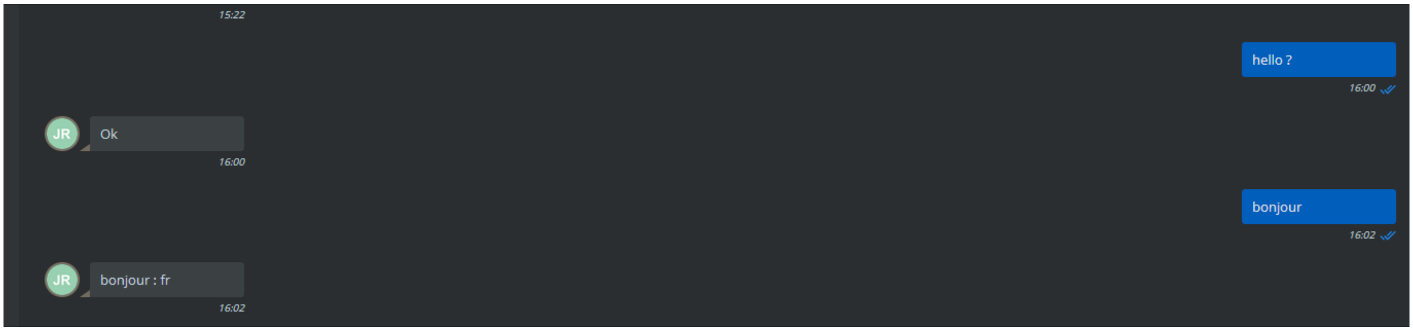
```
APP_SECRET=
ID=
LOGIN=
PASSWORD=
PROJECT_ID=
```

Dialogflow configuration



1 Hi there, friend!
2 Hey!
3 Hello!
4 Greetings!
5 Bonjour mon gars, comment ça va ?

Responses



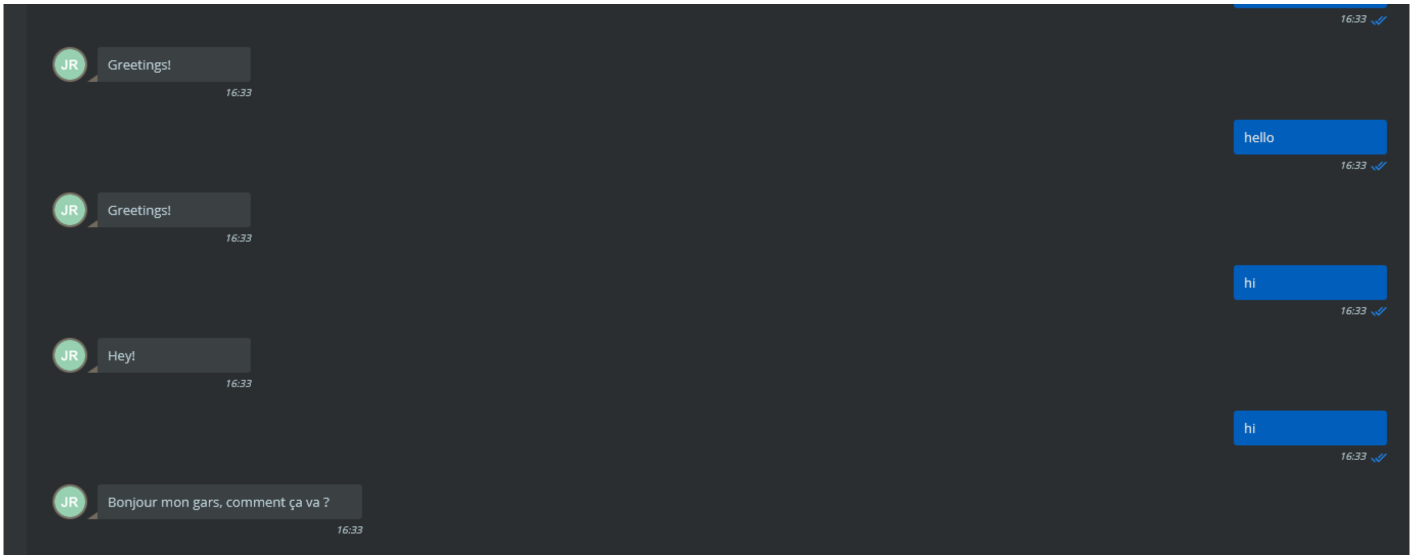
15:22

hello ?
16:00

JR Ok
16:00

bonjour
16:02

JR bonjour : fr
16:02



16:33

JR Greetings!
16:33

hello
16:33

JR Greetings!
16:33

hi
16:33

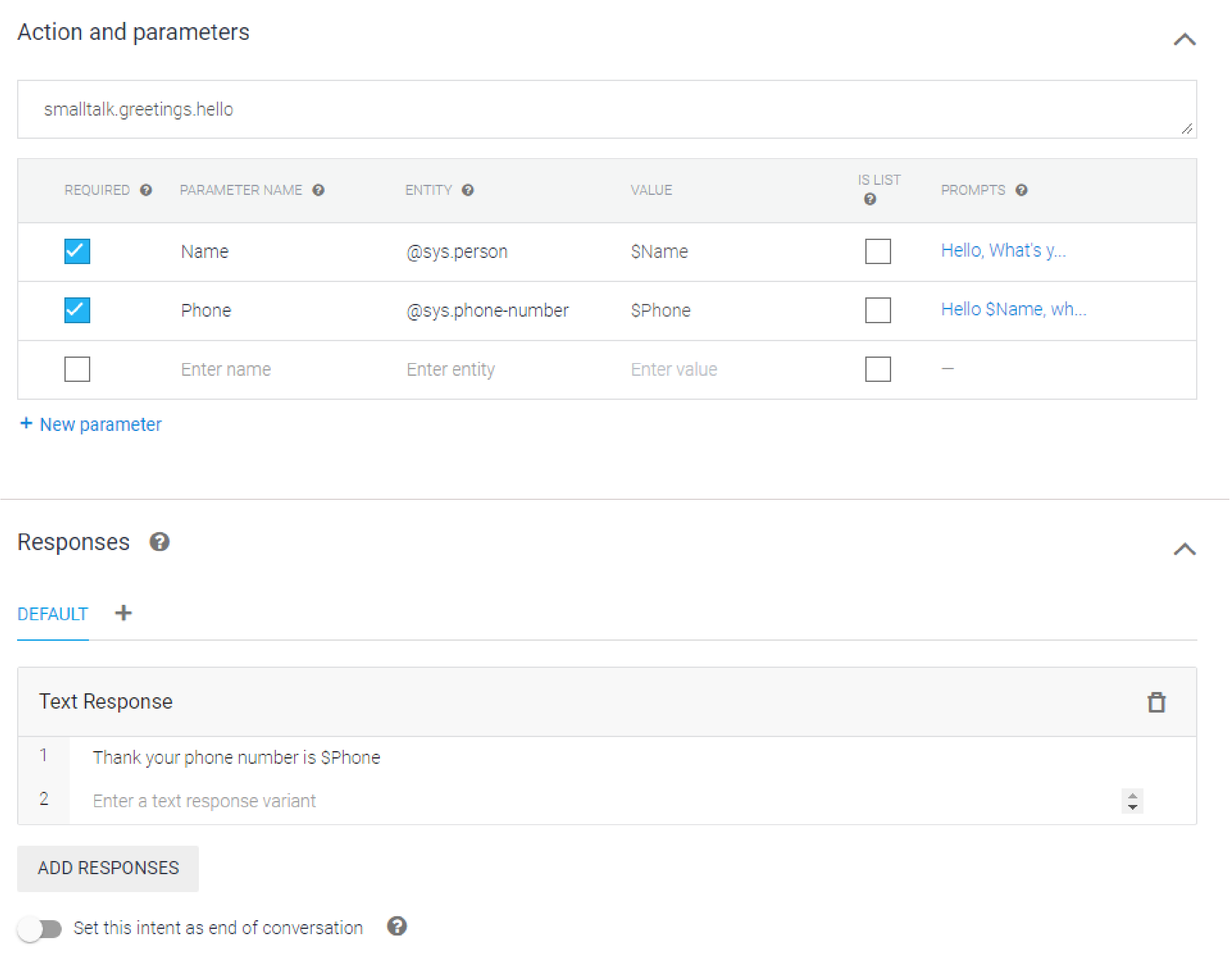
JR Hey!
16:33

hi
16:33

JR Bonjour mon gars, comment ça va ?
16:33

Ask name and phone number

Dialogflow configuration



Action and parameters

smalltalk.greetings.hello

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	Name	@sys.person	$Name	□	Hello, What's y...
☑	Phone	@sys.phone-number	$Phone	□	Hello $Name, wh...
□	Enter name	Enter entity	Enter value	□	–

Responses

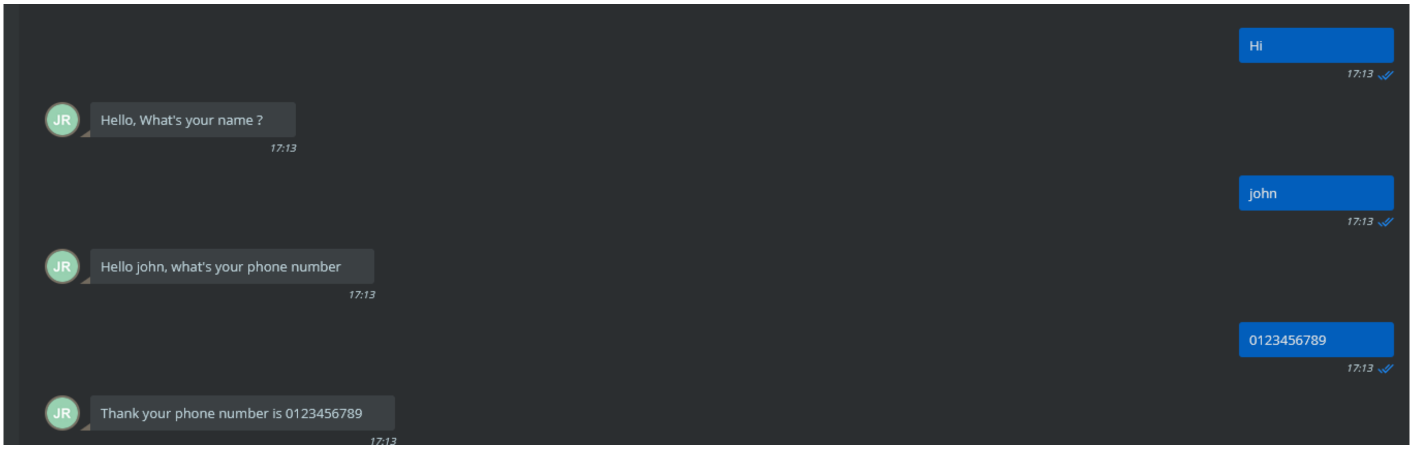
Text Response

1 Thank your phone number is $Phone

2 Enter a text response variant

Set this intent as end of conversation

Responses



Hi 17:13

JR Hello, What's your name ? 17:13

john 17:13

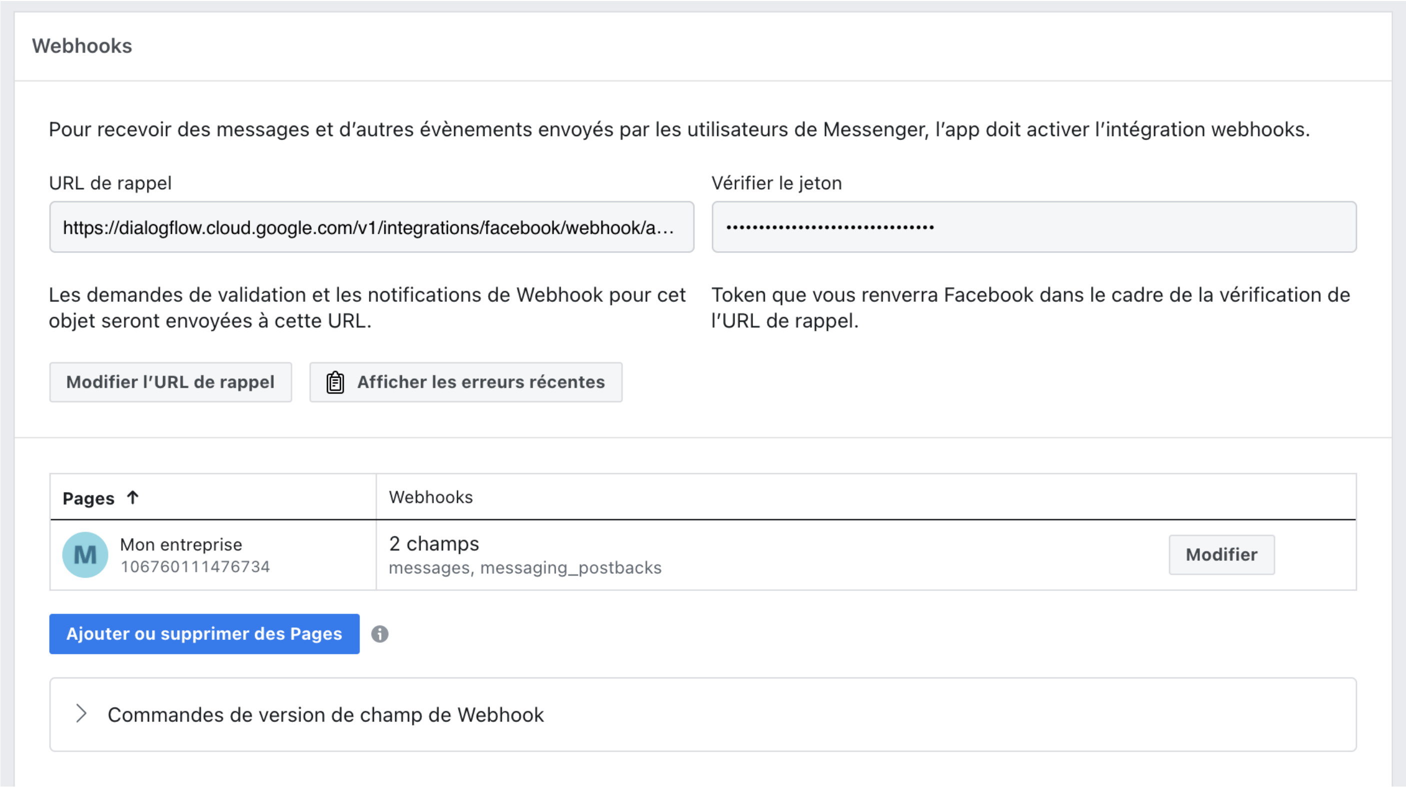
JR Hello john, what's your phone number 17:13

0123456789 17:13

JR Thank your phone number is 0123456789 17:13

Messenger

Configuration



Webhooks

Pour recevoir des messages et d’autres évènements envoyés par les utilisateurs de Messenger, l’app doit activer l’intégration webhooks.

URL de rappel: https://dialogflow.cloud.google.com/v1/integrations/facebook/webhook/a…

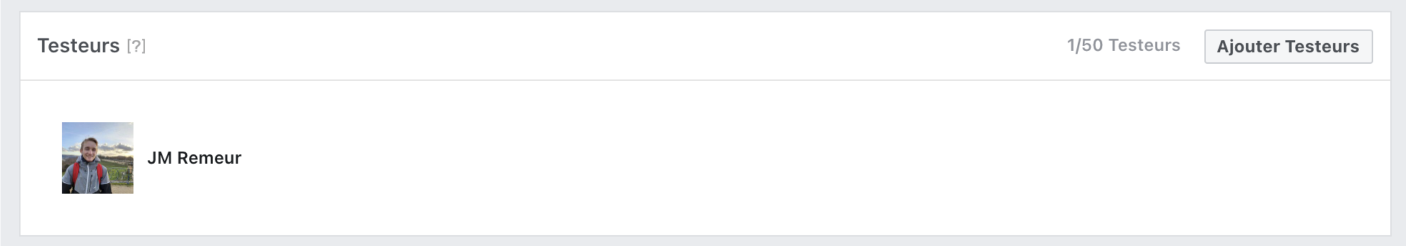
Vérifier le jeton: ••••••••••••••••••••••••••••••••

Les demandes de validation et les notifications de Webhook pour cet objet seront envoyées à cette URL.

Token que vous renverra Facebook dans le cadre de la vérification de l’URL de rappel.

Pages ↑	Webhooks	
M Mon entreprise 106760111476734	2 champs messages, messaging_postbacks	Modifier

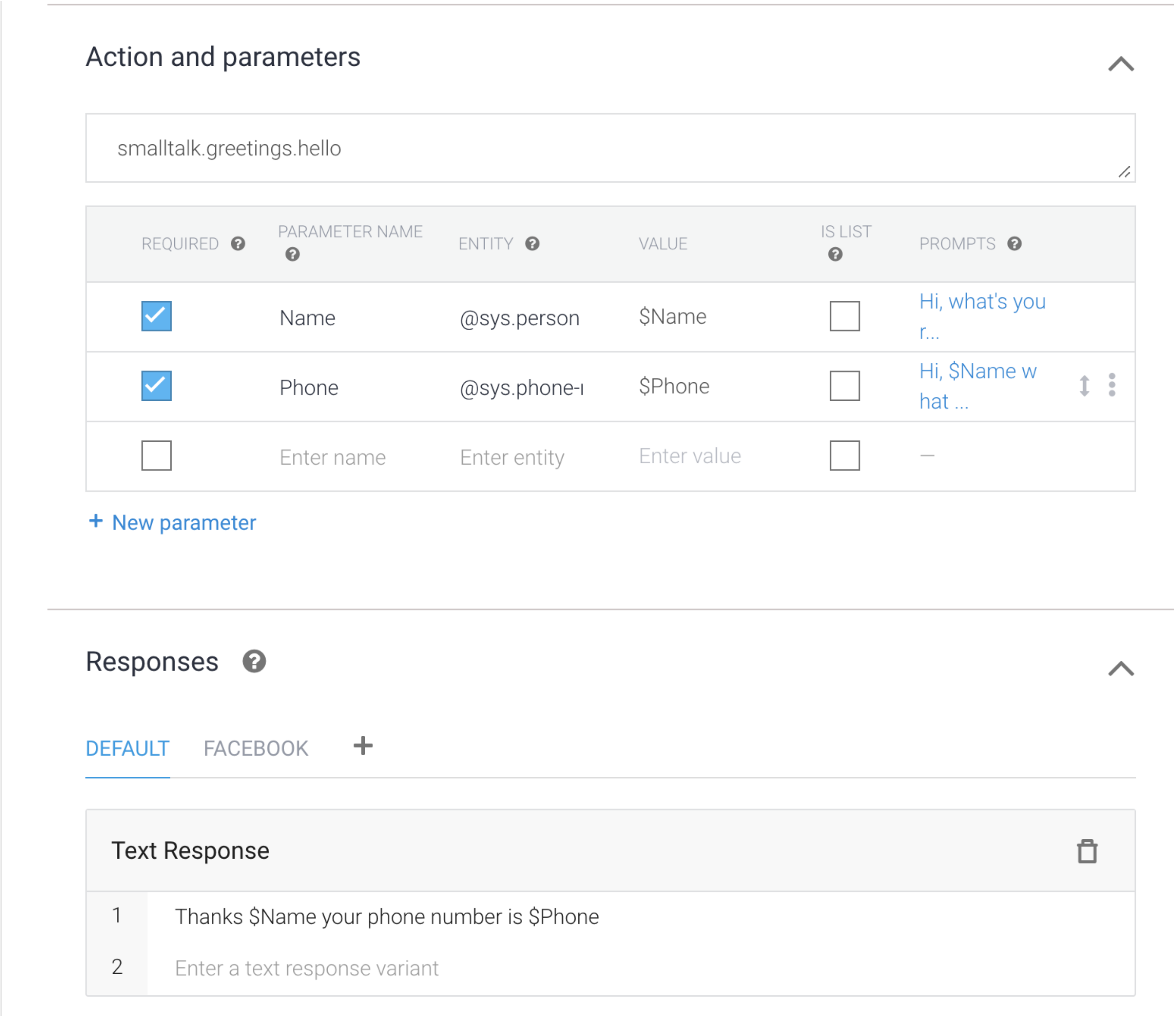
Ajouter ou supprimer des Pages
Commandes de version de champ de Webhook



Testeurs [?]
1/50 Testeurs
Ajouter Testeurs

JM Remeur

Dialogflow configuration



Action and parameters

smalltalk.greetings.hello

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	Name	@sys.person	$Name	□	Hi, what's you r...
☑	Phone	@sys.phone-ı	$Phone	□	Hi, $Name w hat ...
□	Enter name	Enter entity	Enter value	□	—

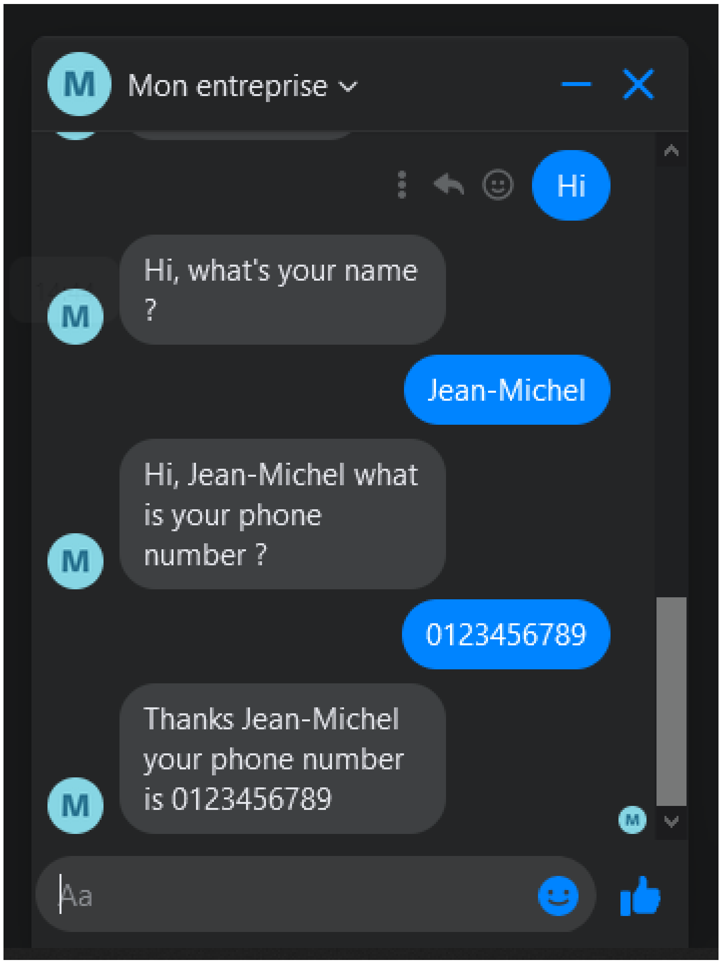
Responses

Text Response

1 Thanks $Name your phone number is $Phone

2 Enter a text response variant

Responses



Hi

Hi, what's your name ?

Jean-Michel

Hi, Jean-Michel what is your phone number ?

0123456789

Thanks Jean-Michel your phone number is 0123456789