

---

## Tutorial: Create a simple custom skill

---

The Watson Assistant provides a boilerplate in Node.js for creating a simple hello world skill. Using this skill, when you say hello to your assistant, your assistant responds with hello world.

The high-level steps in this tutorial are as follows:

1. Clone the git repository of the hello world skill to your local system. Start the skill on your local system.
  2. Test that the skill is working. Simulate a conversation with the hello world skill. Use the Converse REST API that is running on your local system.
  3. Register your locally-hosted skill with your Watson Assistant instance in IBM Cloud.
  4. Add your hello world skill to the skill set of your Watson Assistant instance.
  5. Test that the skill is working. Simulate a conversation with the hello world skill using the converse REST API of your Watson Application on IBM Cloud.
  6. Deploy your hello world skill to IBM Cloud and start it.
  7. Update the Watson Assistant application to reference the hello world skill on IBM Cloud.
  8. Test that the skill is working. Simulate a conversation with your hello world skill on IBM Cloud using the converse REST API that your Watson application is running.
- 

### Before you begin

1. Set up access to your Watson Assistant application instance on IBM Cloud. For instructions, see [Access your Watson Assistant instance](#).
  2. Install [Git](#).
  3. Register for an [IBM Cloud account](#).
  4. Install the [IBM Cloud CLI tool](#).
  5. Install [NGROK](#) to create a secure tunnel from your skill to your Watson Assistant instance on IBM Cloud.  
Start the secure tunnel. From the command-line, enter `ngrok http 10011`
  6. (Optional) Add the regex browser plugin to your web browser.
- 

The detailed steps in this tutorial are as follows:

### Step 1: Clone the hello world skill

1. Copy the hello world skill boilerplate to your local system.
  - i. Click Fork to take a copy of the repository.
  - ii. Click Clone or download. Copy the git url.
  - iii. Open the git bash command-line utility and enter `git clone git_url`.
2. Install the node dependencies for the skill.
  - i. Enter `cd SkillBoilerplate`.
  - ii. Enter `npm install`.
3. Specify regex as the NLU engine to use in the skill. The skill is configured to use two NLU engines; `wcs` (Watson Conversation Service) and `regex`. Remove `wcs` for this tutorial.
  - i. Go to the `res/assets` directory of your local skill.
  - ii. Find 

```
"nlu": [ "regex", "wcs", "skill"]
```
  - iii. Remove `"wcs"`.
  - iv. Save the file.
4. Start the skill. From the command line, enter: `npm run start`

### Step 2: Test your skill from your local system.

1. Start a web browser and open the converse REST API in the Swagger UI. Enter:  
`http://localhost:10011`
2. Go to Converse.
3. Click /converse.
4. Click Try it out.
5. Click Execute.  
The converse REST API responds with "Hello world". The response is included the text attribute of the following JSON data.

**Note:** In this tutorial, you sent both the intent and the input directly to a single skill. In reality, the Watson Assistant performs some additional steps. It determines which skills can process an intent and it determines which skill is best placed to handle the intent.

### Step 3: Register your skill with your Watson Assistant.

1. Start a secure NGROK tunnel between your skill and your Watson Assistant service on IBM Cloud. Go to the directory where you install NGROK and enter:  
`ngrok http 10011`

2. Using the Skills REST API of your Watson Assistant, register your skill with your assistant. Enter:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "name": "paste-your-skill_name_here", "url": "https://paste-your-ngrok-key-here.ngrok.io" }' 'https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/skills?api_key=paste-your-api-key-here'
```

A skill added successfully message is displayed.

## Step 4: Add your skill to the skill set of your Watson Assistant.

1. Your assistant typically has multiple skills associated with it. Using the skillset REST API of your assistant, create a skillset. Enter:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "name": "paste_skillset_name_here" }' 'https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/skillSets?api_key=paste-your-api-key-here'
```

A skillSet added successfully message is displayed.

2. Add your new skill to your skillset. Enter:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "name": "paste_skillset_name_here" }' 'https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/skillSets?api_key=paste-your-api-key-here'
```

A link between myCollection and myHelloWorld added message is displayed.

## Step 5: Test your skill from your Watson Assistant.

1. Use the converse REST API to converse with your skill. Enter:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "text": "Hello", "language": "en-US", "userID": "application-14c", "deviceType": "phone", "additionalInformation": { "context": {} } }' 'https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/skillSets/paste_skillset_name_here/converse?api_key=paste-your-api-key-here'
```

The JSON data that is returned includes the following text:

```
"speech": { "text": "Hello world" }
```

## Step 6: Deploy your skill to IBM Cloud.

Deploy your skill to IBM Cloud to make your skill available to other users.

1. Stop your local skill. Enter `Ctrl C` to stop the skill.
2. Update the host name and skill name in the manifest.yml file of your skill.
3. Log in to IBM Cloud.

```
bx login --sso -a api.ng.bluemix.net -o paste-your-IBMid-here -s dev
```

4. Push your skill to IBM Cloud. Enter:

```
bx app push *skill_name*
```

An App started message is returned.

5. Verify that your skill is running and reachable on IBM Cloud using the /healthcheck API. Enter:

```
curl -X GET --verbose --header 'Accept: application/json' https://paste_your-name-hello-world-skill_name_here.mybluemix.net/v1/api/healthcheck
```

If your skill is running and accessible, a 200 OK response is returned. For example:

```
```> Connected to carloshelloskill.mybluemix.net (158.85.156.19) port 80 (#0)
```

```
GET /v1/api/healthcheck HTTP/1.1
```

```
Host: sampleskill.mybluemix.net
```

```
User-Agent: curl/7.51.0
```

```
Accept: application/json
```

```
< HTTP/1.1 200 OK < X-Backside-Transport: OK OK < Connection: Keep-Alive < Transfer-Encoding: chunked < Access-Control-Allow-Origin: * < Content-Type: application/json < Date: Tue, 27 Jun 2017 01:05:06 GMT < X-Powered-By: Express < X-Global-Transaction-ID: 1919343303``
```

If the skill is not accessible, complete these steps

- Check that the URL you specified matches the host name of the skill on IBM Cloud.
- Use the skill URL to access the skill to verify that it is running.
- From a web browser, check if you can open the Swagger documentation for the skill. For example: enter the URL <https://sampleskill.mybluemix.net/docs>.

## Step 7: Update the Watson Assistant application to reference the hello world skill on IBM Cloud

Use the skills REST API of your Watson Assistant to add the skill that is running on IBM Cloud with your assistant. Enter:

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "name": "myHelloWorld", "url": "https://your-name-hello-world-skill.mybluemix.net" }' 'https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/skills/myHelloWorld?api_key=paste-your-api-key-here'
```

A `skill updated successfully` message is displayed.

## Step 8: Test your externally-deployed skill from your Watson Assistant.

Use the converse REST API to converse with your skill. Enter:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "text": "Hello", "language": "en-US", "userID":  
"application-14c", "deviceType": "phone", "additionalInformation": { "context": {} } }' 'https://watson-personal-assistant-  
toolkit.mybluemix.net/v2/api/skillSets/mySet/converse?api_key=paste-your-api-key-here'
```

The JSON data that is returned includes the following text:

```
"speech": { "text": "Hello world" }
```

You have completed the tutorial for creating a simple skill.