bst Documentation

Release

John Kelvie, Bela Vizy, Michael Myers

August 31, 2016

Table of Contents:

CHAPTER 1

Getting Started

1.1 Installation

Make sure you have NPM and node installed:

```
$ node --version && npm --version
```

We support node version `4.x.x` and above. For help installing, see How To Install NPM

Next, install the bespoken tools command line tool (bst):

```
$ npm install bespoken-tools -g
```

Note: If you are on MacOS and the command fails, it is probably because you need to run it with sudo, like this:

```
$ sudo npm install bespoken-tools -g
```

Verify the installation by typing:

```
$ bst
```

1.2 Updating

To update bst:

```
$ npm update bespoken-tools -g
```

CHAPTER 2

Configuring your Skill for bst proxy

In order to leverage the bst proxy functionality, you must configure your skill from the Amazon Developer Console to point to the bst proxy server.

2.1 Configure your Skill Endpoint

When you run the bst proxy command, you will see a URL output:

```
$ bst proxy lambda index.js
BST: v0.6.5  Node: v4.3.2

Your URL for Alexa Skill configuration:
https://proxy.bespoken.tools?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
```

From your Skill's list, select an existing skill to edit or create a new one. Navigate to the Configuration step and update the HTTPS endpoint to the one provided by the proxy command.

Make sure Account linking is set to "No" and on the SSL Certificate step select "My development endpoint is a subdomain of a domain that has a wildcard certificate from a certificate authority".

__ Note__ To help generate your configuration URL, you can also use $ bst proxy urlgen

2.2 Path Component for proxy http

When using bst proxy http for local servers running on a specific port, you need to make sure your path components are correct.

For example, when you run the bst proxy http command:

```
$ bst proxy http 9999
BST: v0.6.5  Node: v4.3.2

Your URL for Alexa Skill configuration:
https://proxy.bespoken.tools/YOUR/SKILL/PATH?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
(Be sure to put in your real path and other query string parameters!)
```

And your local server listens on path `/alexa-skill`, you need to update the URL with your path before configuring your Skill.

```
https://proxy.bespoken.tools/alexa-skill?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
```

CHAPTER 3

bst proxy

3.1 Overview

The proxy command allows you to interact with a local service running on your machine via an Alexa device. Using it, you can make changes to code running on your machine and immediately make them available via an Echo or the Alexa simulator.

The proxy tool works either directly with Node/JavaScript lambda code - **proxy lambda**. Or it can proxy any http service using **proxy http**.

The two commands are described below, as well as the urlgen helper command.

3.2 bst proxy lambda

Overview

The proxy lambda command allows you to run a Lambda as a local service your machine.

Note

- The command currently only supports Node/JavaScript Lambdas.

Usage

To use it, invoke it with the Lambda file to run. The proxy will automatically use the latest code in your working directory.

Syntax:

```
$ bst proxy lambda <PATH_TO_LAMBDA>
```

Example:

```
$ bst proxy lambda index.js
```

You can learn more here at our Node.js Tutorial:

3.3 bst proxy http

Overview

Proxy http allows you to interact with a local service running on your machine (on a port) via an Alexa device.

Usage

The proxy http command takes one command, the that your local Alexa service is listening on. All traffic coming from Alexa will be forwarded to it.

Syntax:

```
$ bst proxy http <PORT>
```

Example:

```
$ bst proxy http 9999
```

You can learn more here at our Java Tutorial

3.4 bst proxy urlgen

Overview

Your skill must be setup to point at our server. For example, if the URL for your skill is normally:

```
https://myskill.example.com/skillA
```

It should instead be configured to point at the bst proxy server, like so:

```
https://proxy.bespoken.tools/skillA?node-id=1b84270f-5b58-4176-a8b6-7b5b1c03a308
```

Note the Node ID passed in the query string. This is a UUID that ties off your local proxy with our server.

The rest of the URL path and query string should be unchanged.

For more information on configuring your Skill see Configuring your Skill for bst proxy.

Usage

The proxy urlgen command can help generate the endpoint.

Syntax:

```
$ bst proxy urlgen <URL>
```

Example:

```
$ bst proxy urlgen https://myskill.example.com/skillA
```

The above example command will then provide you with HTTPS endpoint that is required during the configuration step when you setup your Alexa Skill.

CHAPTER 4

bst speak

4.1 Overview

The speak command generates intent requests for your service as if they were coming from Alexa itself.It works in a manner very similar to the Alexa simulator available via the Alexa developer console.

To start using it, you will need a local file that contains your Intent Schema and Sample Utterances.By default, we have adopted the pattern used by the Alexa Skills Sample projects (see here).

That is, we look for the Interaction Model files inside a folder called speechAssets located off the source root.

You can specify an alternative location via options to the command-line.

4.2 Speaking

To invoke the speak command, simply type:

```
$ bst speak <UTTERANCE>
```

For example:

```
$ bst speak Hello World
```

The speak command will return the full request and response of the interaction with your Skill service.

By default, the system will:

- Use the Intent Model and Sample Utterances in the speechAssets folder under the current working directory
- Use the service currently running via the `bst proxy` command

If no service is currently running via bst proxy, and HTTP endpoint can be specified with the `--url` option:

```
$ bst speak Hello World --url https://my.skill.com/skill/path
```

4.3 Speech Asset Format and Location

If your speech assets (Intent Model and Sample Utterances) are not stored under ./speechAssets, you can use an option to specify another location.

By default, we look for:

- `./speechAssets/IntentSchema.json`
- `./speechAssets/SampleUtterances.txt`

Example:

```
$ bst speak https://my.skill.com/skill/path Hello World -i interactions/IntentSchema.json -s interact
```

The format of these files is the same as they are entered in the Alexa Skill configuration.

The Intent Schema is a JSON file. Samples utterances is a space-delimited text file.

An example of these files can be found here.

4.4 Working With Slots

Slot handling is a bit tricky. To send an utterance that uses slots, surround the slot variables like so:

```
{MySlot}
```

For example, if the sample utterance was defined as:

```
HelloWorld Hello world, my name is {Name}
```

Then the speak command would be:

```
$ bst speak Hello World, my name is {John}
```

The value `John` will then be automatically placed in the Name slot for the utterance on the request.

4.5 Other Notes

Default Utterances

We currently use a simple algorithm to pick out a default Intent if none of the sample utterances match the supplied utterance.

This algorithm is - the first utterance of the first intent defined in the sample file.

This is meant to loosely mimic the behavior of the Alexa Simulator, which also seems to randomly select an intent/utterance when none matches.

CHAPTER 5

Local Java Server

This tutorial shows you how to get started developing for Alexa with Java and Maven.

5.1 Prerequisites

- bespoken tools (bst)
  - `$ npm install bespoken-tools -g`
  - Installation Instructions
- maven
  - OSX with homebrew: `$ brew install maven`
  - Installation Instructions
- Amazon Developer Account
  - Amazon Developer

5.2 Getting Started

Clone the bst project:

```
$ git clone https://github.com/bespoken/bst
```

Go to the sample java skill

```
$ cd bst/samples/java
```

5.3 Run the Sample Java Skill

From within bst/samples/java directory, compile the example with this command:

```
$ mvn compile
```

Run the server with this command:

```
$ mvn exec:java -Dexec.executable="java" -DdisableRequestSignatureCheck=true -Dexec.args=$@
```

The service will listen on port 9999 by default.

5.4 Start bst proxy

Open a new terminal and start the proxy for port 9999:

```
$ bst proxy http 9999
```

5.5 Configure your Skill

From the Alexa Skills Kit list within the Amazon Developer's Console:

Choose "Add a New Skill"

Fill out the Information tab

- Give your skill a name and invocation phrase, 'bst java sample' and 'greeter' for example

Fill out the Interaction Model

- Copy and paste the Intent Schema from here
- Copy and paste the Sample Utterances from here

Configure the Endpoint

When you started the proxy, bst printed out a URL that you need to configure your skill:

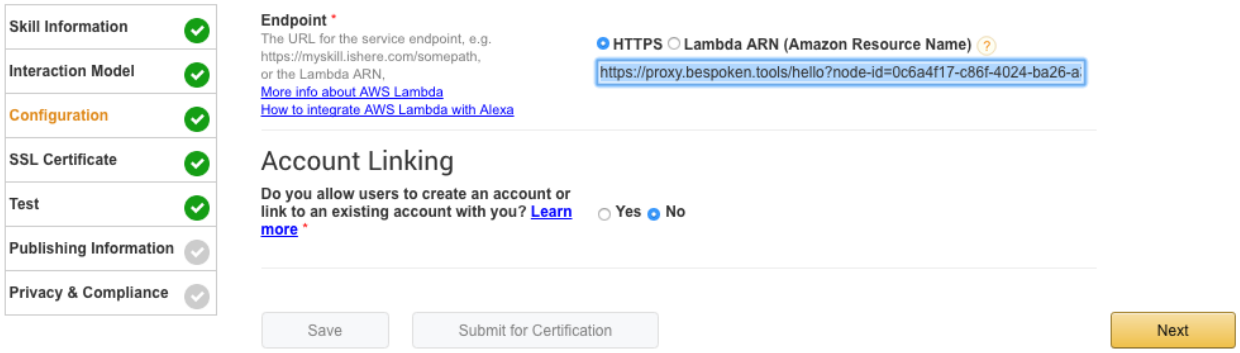
```
$ bst proxy http 9999
BST: v0.6.5  Node: v4.3.2

Your URL for Alexa Skill configuration:
https://proxy.bespoken.tools/YOUR/SKILL/PATH?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
(Be sure to put in your real path and other query string parameters!)
```

Alternatively, you can create this URL via the `proxy urlgen` command.

You first need to modify it to the path that your server is listening on, in this case it is `/hello`.

```
https://proxy.bespoken.tools/hello?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
```

Copy and paste this URL as your endpoint:



Also make sure you select “HTTPS” and account linking to “NO”.

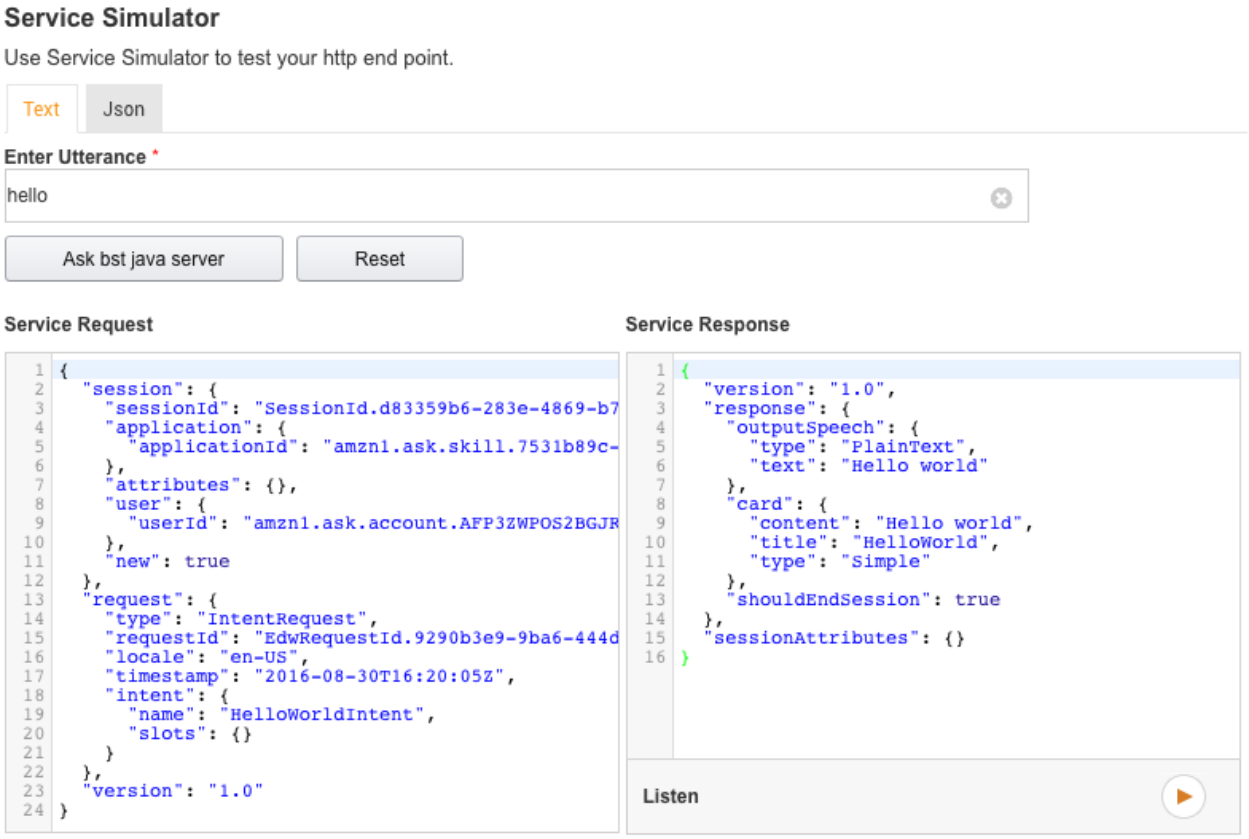
Configure SSL

On the SSL Certificate page, select the middle radio button,”My development endpoint is a subdomain of a domain that has a wildcard certificate from a certificate authority”

5.6 Test

Go to the service simulator, and type: “hello” and hit “Ask <Your Skill Name>”.

You should get a valid JSON in reply:



5.7 Next Steps

You can now start adding functionality to your skill. To learn more about coding Alexa Skills, see the official documentation

You can also try it out on an Alexa device like an Echo, as long as it is registered with your account. Just say “Open <Your Invocation Name>” to use it.

CHAPTER 6

Nodejs Lambda

This tutorial shows you how to get started developing for Alexa Skills Kit using a Nodejs Lambda.

6.1 Prerequisites

- bespoken tools (bst)
  - `$ npm install bespoken-tools -g`
  - Installation Instructions
- Amazon Developer Account
  - Amazon Developer

6.2 Getting Started

Clone the Amazon Alexa Skills Kit for JavaScript repo:

```
$ git clone https://github.com/amzn/alexa-skills-kit-js
```

Go to the root level of the sample:

```
$ cd alexa-skills-kit-js/
```

6.3 Start bst proxy

For Nodejs Lambdas, bst proxy command, in addition to setting up the proxy, will run your lambda for you and even reload it on changes.

This will start the helloWorld lambda:

```
$ bst proxy lambda samples/helloWorld/src/index.js
```

6.4 Configure your Skill

From the Alexa Skills Kit list within the Amazon Developer's Console:

Choose "Add a New Skill"

Fill out the Information tab

- Give your skill a name and invocation phrase, 'bst nodejs sample' and 'greeter' for example

Fill out the Interaction Model

- Copy and paste the Intent Schema from here
- Copy and paste the Sample Utterances from here

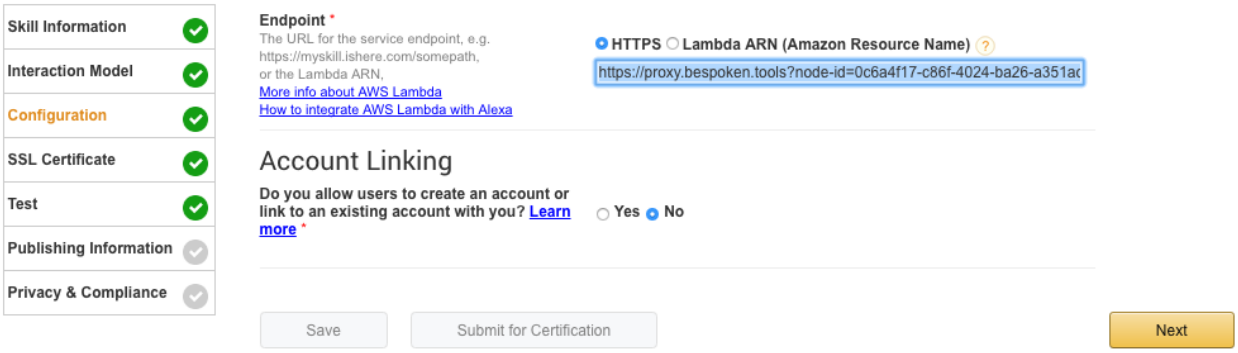
Configure the Endpoint

When you started the proxy, bst printed out a URL that you need to configure your skill:

```
$ bst proxy lambda samples/helloWorld/src/index.js
BST: v0.6.5  Node: v4.3.2

Your URL for Alexa Skill configuration:
https://proxy.bespoken.tools?node-id=0c6a4f17-c86f-4024-ba26-a351ac319431
```

Alternatively, you can create this URL via the `proxy urlgen` command.

Copy and paste this URL as your endpoint:



Also make sure you select "HTTPS" and account linking to "NO".

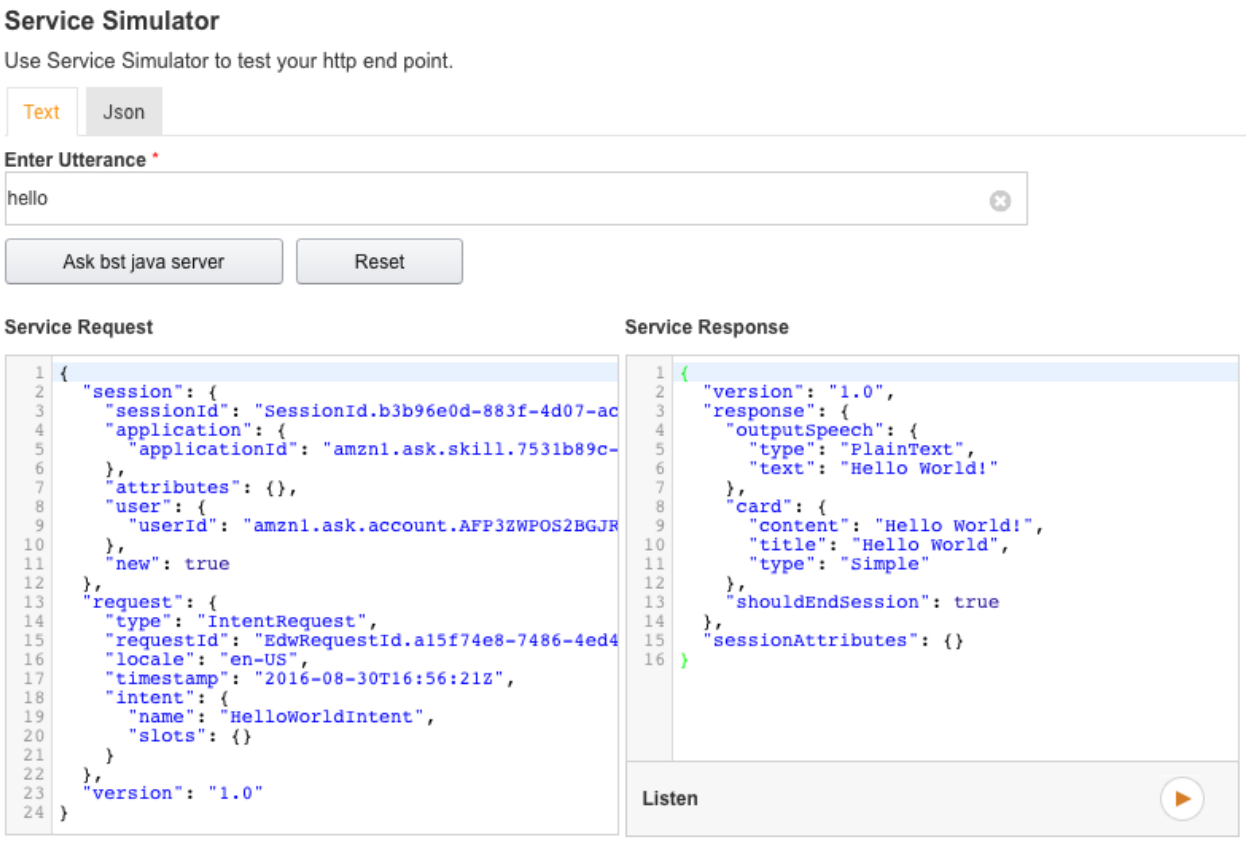
Configure SSL

On the SSL Certificate page, select the middle radio button,"My development endpoint is a subdomain of a domain that has a wildcard certificate from a certificate authority"

6.5 Test

Go to the service simulator, and type: "hello" and hit "Ask <Your Skill Name>".

You should get a valid JSON in reply:



6.6 Next Steps

You can now start adding functionality to your skill. To learn more about coding Alexa Skills, see the official documentation

You can also try it out on an Alexa device like an Echo, as long as it is registered with your account. Just say "Open <Your Invocation Name>" to use it.