

# Building Serverless Web Applications with OpenWhisk

[bluemix.net/openwhisk](http://bluemix.net/openwhisk)



Niklas Heidloff, IBM  
Developer Advocate  
[@nheidloff](https://twitter.com/nheidloff)

July 2017

# What is ‘Serverless’ and ‘Function as a Service’ ?

- Existence of servers is hidden from developers
- Code is executed on-demand in an auto-scaled, managed environment through event driven triggers
- Encourages and simplifies developing microservice-oriented solutions via functions

# Why Serverless for Web Applications ?

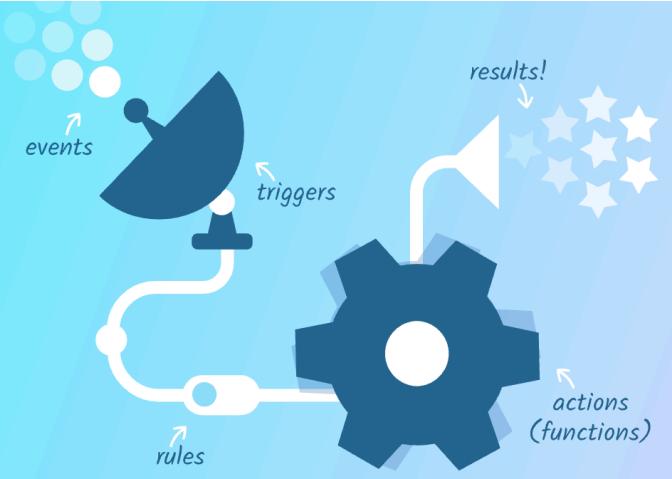
- Often cheaper than other server-side compute options since you only pay for the exact duration your code is executing
- Easier to operate than other compute options
  - Example: Auto-scalability
  - Example: Built-in API management
- Serverless functions can be built with JavaScript skills which web developers have

# Serverless

# Apache OpenWhisk

Apache OpenWhisk (Incubating) is a serverless, **open source** cloud platform that executes functions in **response to events** at any scale.

[Get Started](#)



# Demos

- Creation of simple JavaScript actions
- Web actions for anonymous access
- API management, including OAuth access
- Angular application accessing protected action

Annotations in orange

# Create JavaScript Action

Demo

IBM Bluemix OpenWhisk

Getting Started ▾

MY ACTIONS Create an Action Run this Action View Action Details Browse Public Packages

## Create an Action

By creating an OpenWhisk action, you automatically get a REST endpoint

 niklas-demo-action

You are creating an OpenWhisk [action](#). Choose an execution runtime, and, optionally, a sample from which to start. You can also choose to change the default quota settings.

MY ACTION

Choose an Execution Runtime

JS Node.js 6

Select a sample (or blank) to start with

Hello World in Javascript

Memory Quota In megabytes. [Learn more](#)

256

Time Limit In seconds. [Learn more](#)

60

\$0.000017 USD/GB-SECOND [Learn more](#)

[Cancel](#) [Create Action](#)

# Demo

The screenshot shows the IBM Bluemix OpenWhisk interface. On the left, there's a sidebar with navigation links: Getting Started, Manage, **Develop**, Monitor, and APIs. The Develop section is currently selected. The main area has tabs for 'MY ACTIONS', 'MY SEQUENCES', 'MY RULES', and 'MY TRIGGERS'. Under 'MY ACTIONS', the 'niklas-demo-action' is selected, and its code is displayed:

```
1  /**
2   *
3   * main() will be invoked when you Run This Action.
4   *
5   * @param OpenWhisk actions accept a single parameter,
6   * which must be a JSON object.
7   *
8   * In this case, the params variable will look like:
9   * {
10    *   "message": "xxxx"
11   *
12   * @return which must be a JSON object.
13   * It will be the output of this action.
14   */
15  function main(params) {
16    return { "message": "you sent me " + params.message };
17 }
```

Below the code, it says 'This Code Is Live! [View REST Endpoint](#)'. At the bottom, there are buttons for 'Link into a Sequence' and 'Automate this Action'. To the right of the code, the text 'Synchronous Action' is displayed in large red letters.

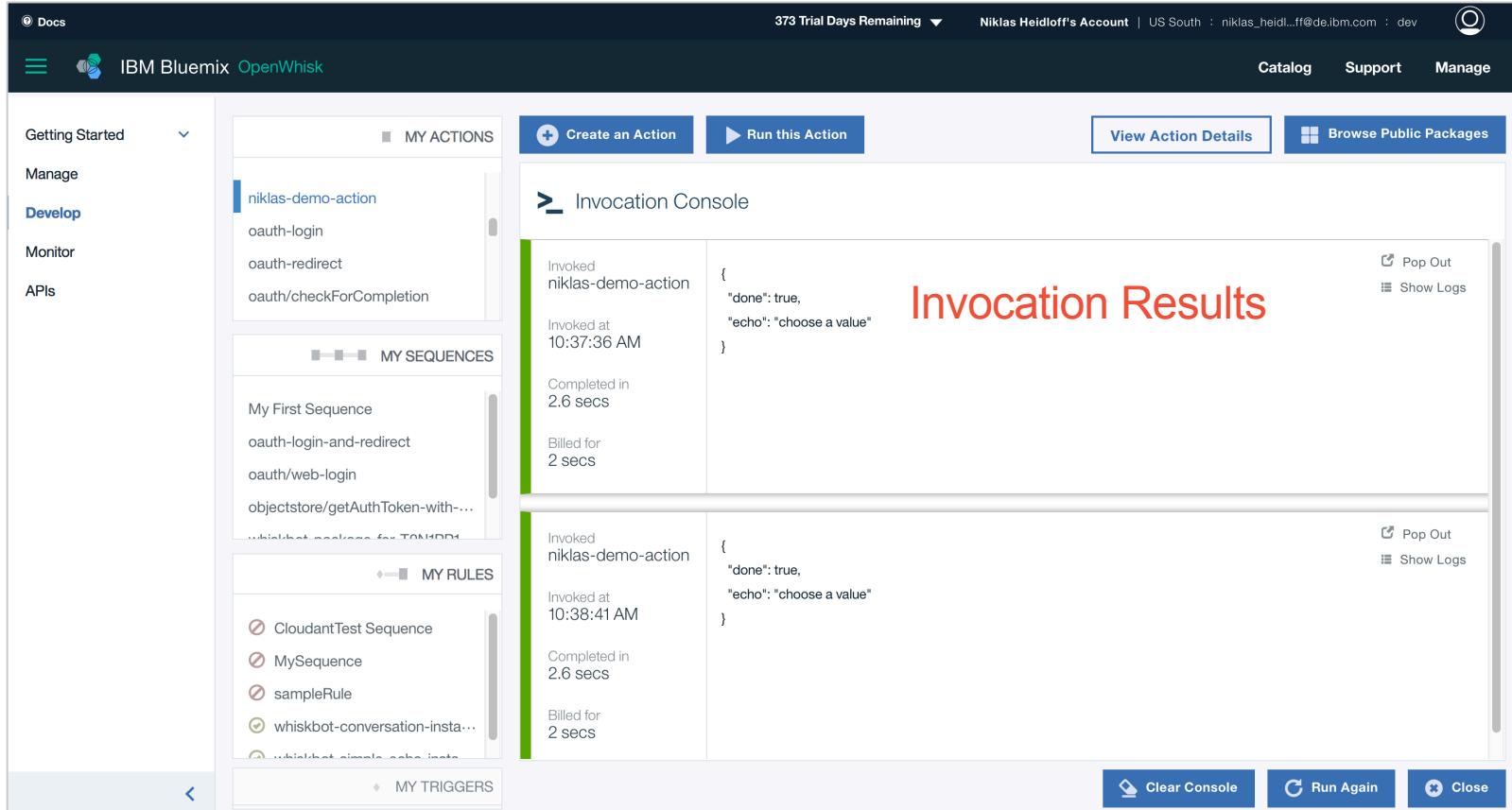
# Demo

The screenshot shows the IBM Bluemix OpenWhisk interface. On the left, there's a sidebar with navigation links: Getting Started, Manage, **Develop**, Monitor, and APIs. The Develop link is currently selected. The main area has tabs for MY ACTIONS, MY SEQUENCES, and MY RULES. Under MY ACTIONS, there's a list with 'niklas-demo-action' expanded, showing three sub-actions: oauth-login, oauth-redirect, and oauth/checkForCompletion. Under MY SEQUENCES, there are four sequences: My First Sequence, oauth-login-and-redirect, oauth/web-login, and objectstore/getAuthToken-with-... . Under MY RULES, there are four rules: CloudantTest Sequence, MySequence, sampleRule, and whiskbot-conversation-inst... . At the top right, there are buttons for 'Create an Action', 'Run this Action', 'View Action Details' (which is highlighted), and 'Browse Public Packages'. Below the code editor, there's a large red text 'Asynchronous Action'. The code editor shows the following Node.js code:

```
1
2  function main(params) {
3    return new Promise(function(resolve, reject) {
4      setTimeout(function() {
5        resolve({ done: true, echo: params.message });
6      }, 2000);
7    })
8  }
9 |
```

At the bottom right, it says 'This Code Is Live! [View REST Endpoint](#)'. At the very bottom, there are buttons for 'Link into a Sequence' and 'Automate this Action'.

# Demo



The screenshot shows the IBM Bluemix OpenWhisk interface. On the left, a sidebar menu includes 'Getting Started', 'Manage', 'Develop' (selected), 'Monitor', and 'APIs'. The main area displays the 'Invocation Console' for the 'niklas-demo-action'. Two invocation logs are shown:

**Invocation 1:**  
Invoked niklas-demo-action  
Invoked at 10:37:36 AM  
Completed in 2.6 secs  
Billed for 2 secs

```
{  
  "done": true,  
  "echo": "choose a value"  
}
```

**Invocation 2:**  
Invoked niklas-demo-action  
Invoked at 10:38:41 AM  
Completed in 2.6 secs  
Billed for 2 secs

```
{  
  "done": true,  
  "echo": "choose a value"  
}
```

Buttons at the bottom right include 'Clear Console', 'Run Again', and 'Close'.

# Demo

IBM Bluemix OpenWhisk

Catalog Support Manage

Back to Getting Started

## CLI

To use the OpenWhisk CLI (Command Line Interface), follow these steps.

- 1.** [Download the MacOS CLI](#) This gives you the OpenWhisk command line interface (CLI) for your Mac OS X system.
- 2.** Set your OpenWhisk **Namespace** and **Action**. You can do this by running the command `wsk property set --apihost openwhisk.ng.bluemix.net` in your terminal.

**\*\*IMPORTANT NOTICE\*\*** OpenWhisk assumes your namespace and action will change as you switch your org or project.

```
wsk property set --apihost openwhisk.ng.bluemix.net
```
- 3.** Verify your setup. Here, we perform a blocking action invoke.

```
wsk action invoke /whisk.system/utils/echo
```

```
{ "message": "hello" }
```

Niklass-MacBook-Pro:openwhisk-serverless-webapp nheidloff\$ wsk

```
openwhisk-serverless-webapp — bash — 80x38
```

```
Usage:
  wsk [command]
```

Available Commands:

bluemix	bluemix integration
action	work with actions
activation	work with activations
package	work with packages
rule	work with rules
trigger	work with triggers
sdk	work with the sdk
property	work with whisk properties
namespace	work with namespaces
list	list entities in the current namespace
api-experimental	work with APIs (experimental)
api	work with APIs

Flags:

--apihost HOST	whisk API HOST
--apiversion VERSION	whisk API VERSION
-u, --auth KEY	authorization KEY
-d, --debug	debug level output
-h, --help	help for wsk
-i, --insecure	bypass certificate checking
-v, --verbose	verbose output

Use "wsk [command] --help" for more information about a command.

```
Niklass-MacBook-Pro:openwhisk-serverless-webapp nheidloff$
```

## Local Development and CLI

# Monitoring and Logs

IBM Bluemix OpenWhisk

Getting Started ▾

Manage

Develop

**Monitor**

APIs

Activity Summary Invocation counts per action or rule

Action	Activations	Avg Time
niklas-demo-action	3	2.04s on average
oauth-login	9 activations	731.70ms
oauth-login-and-redirect	6	1.13s
oauth-redirect	12 activations	38.75ms
QuickDemo	19 activations	17.84ms

Activity Log

Action	ID	Timestamp	Time
niklas-demo-action	57fc7fdbd82943b884f8609b13c26b7e	7/6/2017 10:45:26 AM	2.06s
	{ "done": true, "echo": "" }		
niklas-demo-action	420aeacdd4cd4404a26da5c490d3c5e1	7/6/2017 10:38:42 AM	2s
	{ "done": true, "echo": "choose a value" }		
niklas-demo-action	101765dc5850488aa8e8cc914154d0d1	7/6/2017 10:37:37 AM	2.06s
	{ "done": true, "echo": "choose a value" }		
QuickDemo	63fa43de43344fe8897b5188161ba519	7/6/2017 9:15:58 AM	5ms
	{ "message": "Hello World Niklas" }		

Activity Timeline Invocation count over time

Filtering Options

Time Frame: most recent 50 ▾ Limit to: All Actions ▾  Exclude triggers from the views?

@nheidloff

Demo

The screenshot shows the IBM Bluemix OpenWhisk interface. At the top, there's a dark header with the IBM logo, 'IBM Bluemix OpenWhisk', and account information ('Niklas Heidloff's Account | US South : niklas\_heidloff@de.ibm.com : dev'). Below the header, there are links for 'Catalog', 'Support', and 'Manage'. A sidebar on the left has links for 'Code', 'Triggers', 'Default Parameters', 'Runtime Limits', and 'Additional Details', with 'Additional Details' being the active tab. The main content area has a title 'Additional Details' and a sub-section 'Fully Qualified Name' containing the value '/niklas\_heidloff@de.ibm.com\_dev/niklas-demo-action'. Another sub-section 'Runtime' shows 'Node.js 6' as the selected environment. The 'REST API' section lists a POST method with the URL 'https://openwhisk.ng.bluemix.net/api/v1/namespaces/niklas\_heidloff%40de.ibm.com\_dev/actions/niklas-demo-action'. The 'Web Action' section contains a checked checkbox for 'Enable as Web Action' and an unchecked checkbox for 'Raw HTTP handling'. It also displays a 'Web Action URL' at 'https://openwhisk.ng.bluemix.net/api/v1/web/niklas\_heidloff%40de.ibm.com\_dev/default/niklas-demo-action.json'. A red banner on the right side of the page reads 'Web Actions for anonymous Access'.

Additional Details

Fully Qualified Name

/niklas\_heidloff@de.ibm.com\_dev/niklas-demo-action

Runtime

The environment in which your code will be executed.

Node.js 6

REST API

Actions can be invoked from outside of OpenWhisk via a standard API call.

HTTP METHOD	URL
POST	https://openwhisk.ng.bluemix.net/api/v1/namespaces/niklas_heidloff%40de.ibm.com_dev/actions/niklas-demo-action

Web Action

Allow your OpenWhisk actions to handle HTTP events. [Learn more about Web Actions.](#)

Enable as Web Action    Raw HTTP handling

Web Action URL

[https://openwhisk.ng.bluemix.net/api/v1/web/niklas\\_heidloff%40de.ibm.com\\_dev/default/niklas-demo-action.json](https://openwhisk.ng.bluemix.net/api/v1/web/niklas_heidloff%40de.ibm.com_dev/default/niklas-demo-action.json)

Web Actions for anonymous Access

@nheidloff

Demo

IBM Bluemix OpenWhisk

373 Trial Days Remaining ▾ Niklas Heidloff's Account | US South : niklas\_heidloff@de.ibm.com : dev

Catalog Support Manage

All OpenWhisk APIs

Summary

Definition

Sharing

API Explorer

# Built-in API Management

MyAPI

Expose Managed API

API definition file

Optional import a YAML or JSON format OpenAPI definition file. Imported settings will replace all existing settings.

API definition file ▾

API name \*

Specify a descriptive name for this API.

MyAPI

Base path for API \*

Specify a base path for this API.

/niklas

Operations \*

Create API operations that invoke OpenWhisk actions.

Create operation +

PATH	VERB	PACKAGE	ACTION
/action	GET	default	QuickDemo
			⋮

Demo

Docs

373 Trial Days Remaining ▾ Niklas Heidloff's Account | US South : niklas\_heidloff@de.ibm.com : dev

IBM Bluemix OpenWhisk Catalog Support Manage

All OpenWhisk APIs

Application authentication

You can require consuming applications to authenticate using API key and secret or API key alone.

Method

API key only

Location of API key and secret

Header

Parameter name of API key

X-IBM-Client-ID

Parameter name of API secret

X-IBM-Client-Secret

Rate limiting

When rate limiting is enabled, API calls falling outside of the limit will be rejected and response code 429 will be returned. Given that rate limiting is on a per-key basis, application authentication must be enabled.

The leaky bucket algorithm is used to prevent sudden bursts of invocations of your API. For example, if you set your limit as 10 calls per minute, users will be restricted to 1 call every 6 seconds ( $60/10 = 6$ ).

User authentication

You can control access to your APIs through the OAuth authorization standard. OAuth is a token-based authorization protocol that allows third-party websites or applications to access user data without requiring the user to share personal information.

CORS

Enabling cross-origin resource sharing (CORS) will allow embedded

Require applications to authenticate via API key

Method

API key only

Location of API key and secret

Header

Parameter name of API key

X-IBM-Client-ID

Parameter name of API secret

X-IBM-Client-Secret

Limit API call rate on a per-key basis

Maximum calls

1000

Unit of time

Second

## Require User Authentication

Require users to authenticate via OAuth social login

Provider

Google

Enable CORS so that browser-based applications can call this API

@nheidloff

Demo

IBM Bluemix OpenWhisk

373 Trial Days Remaining ▾ Niklas Heidloff's Account | US South : niklas\_heidloff@de.ibm.com : dev

Catalog Support Manage

All OpenWhisk APIs

Summary MyAPI Analytics for APIs

Expose Managed API ?

Definition Sharing API Explorer

API name MyAPI Operations 1 Route <https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/a172f687d4c7fac2da3546903...>

Rate Limit None Security OAuth social login (Google) CORS enabled

Sharing Not shared with Bluemix organization

Analytics and Logging Last Updated: 10:53 AM Refresh

3 RESPONSES

47ms AVG RESPONSE TIME

API CALLS PER MINUTE

Time	API Calls Per Minute
9:53	0
9:58	0
10:04	0
10:09	0
10:15	0
10:20	0
10:26	0
10:31	0
10:37	0
10:42	0
10:47	0
10:53	3

# API Explorer

Demo

The screenshot shows the IBM Bluemix OpenWhisk API Explorer interface. At the top, there's a navigation bar with 'Docs', '373 Trial Days Remaining', 'Niklas Heidloff's Account | US South : niklas\_heidl...ff@de.ibm.com : dev', and a user icon. Below the navigation is a header with 'IBM Bluemix OpenWhisk' and links for 'Catalog', 'Support', and 'Manage'. On the left, a sidebar titled 'All OpenWhisk APIs' lists 'Summary', 'Definition', 'Sharing', and 'API Explorer' (which is selected). The main area is titled 'MyAPI' and shows 'MyAPI 1.0'. It includes a search bar, an 'Operations' dropdown set to 'by name', and a list of actions: 'getAction'. Below the actions is a 'Security' section with 'google', 'oauth2 application', and 'google' listed. To the right, there's a 'REST' tab, a 'GET' request URL, 'Examples' and 'Try it' buttons, an 'Example request' section with a 'curl' dropdown, and a 'Responses' section showing a successful 200 response. A blue button labeled 'Expose Managed API' is also visible.

API Explorer

373 Trial Days Remaining Niklas Heidloff's Account | US South : niklas\_heidl...ff@de.ibm.com : dev

IBM Bluemix OpenWhisk Catalog Support Manage

All OpenWhisk APIs

Summary Definition Sharing API Explorer

MyAPI

Expose Managed API ?

Operations by name

getAction

Security

google oauth2 application google

REST

GET https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/ai72f687d4c7fac2da3546903009090a18a9643313d2d6e47ff43dd8ede5fa3a/niklas/action

Examples Try it

Example request

curl

```
curl --request GET \
--url https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/ai72f687d4c7fac2da3546903009090a18a9643313d2d6e47ff43dd8ede5fa3a/niklas/action
```

Responses

200 A successful invocation response

200

Google APIs OpenWhiskDemo

API Manager Client ID for Web application [DOWNLOAD JSON](#) [RESET SECRET](#) [DELETE](#)

Dashboard	Client ID	7770290 [REDACTED] apps.googleusercontent.com
Library	Client secret	AzA [REDACTED]
Credentials	Creation date	Jul 4, 2017, 10:55:20 AM

Name

Restrictions  
Enter JavaScript origins, redirect URIs, or both

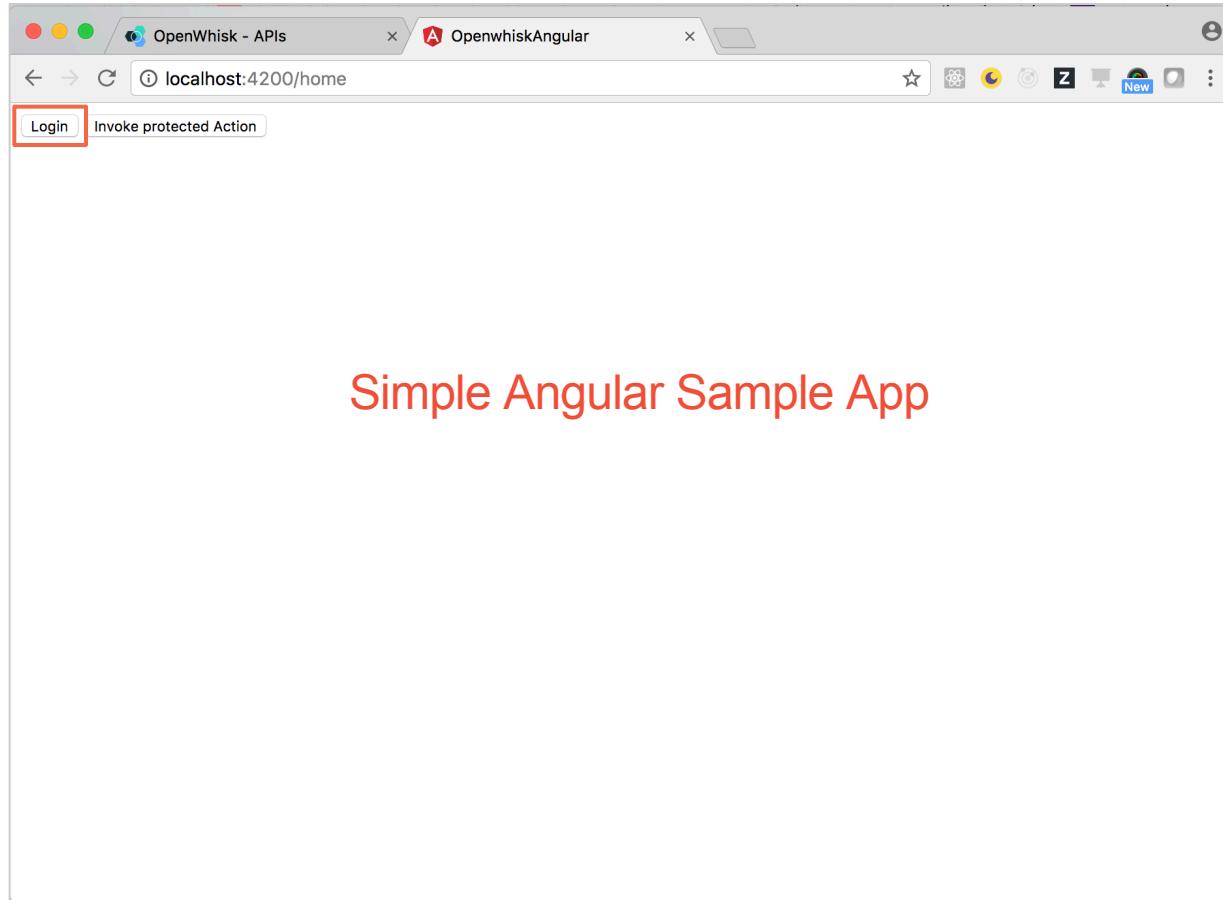
Authorized JavaScript origins  
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (`http://*.example.com`) or a path (`http://example.com/subdir`). If you're using a nonstandard port, you must include it in the origin URI.

Authorized redirect URIs  
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

`https://openwhisk.ng.bluemix.net/api/v1/web/niklas_heidloff@de.ibm.com_dev/default/oauth-login-and-redirect`

## Creation of Google Application

# Demo



# Demo

home.component.ts — openwhisk-serverless-webapp

EXPLORER

OPEN EDITORS

OPENWHISK-SERVERLESS-WEBAPP

angular

- e2e
- node\_modules
- src
  - app
    - # app.component.css
    - app.component.html
    - TS app.component.spec.ts
    - TS app.component.ts
    - TS app.module.ts
    - TS app.routing.ts
    - home.component.html
    - TS home.component.ts
  - assets
  - .gitkeep
  - { providers-template.json }
  - { providers.json }
  - environments
  - favicon.ico
  - index.html
  - TS main.ts
  - nginx.conf
  - TS polyfills.ts
  - # styles.css
  - TS test.ts
  - { tsconfig.app.json }
  - { tsconfig.spec.json }

home.component.ts

```
1 import { Component } from '@angular/core';
2 import { ActivatedRoute, Params, Router } from '@angular/router';
3 import { Http, Headers, URLSearchParams, RequestOptions } from '@angular/http';
4 import * as providers from '../assets/providers.json';
5 import { Observable } from 'rxjs/Observable';
6 import 'rxjs/Rx';

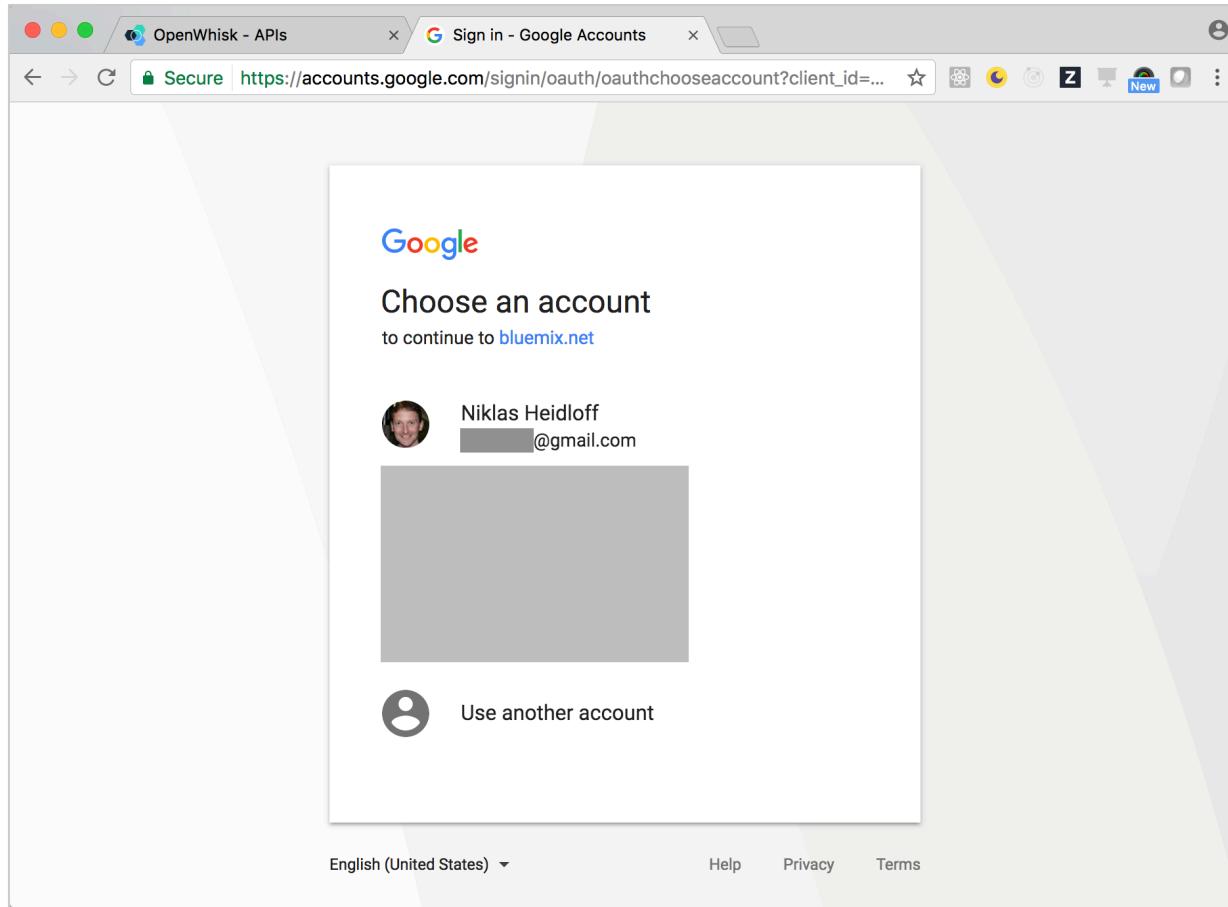
7
8 @Component({
9   selector: 'app-root',
10  templateUrl: './home.component.html'
11 })
12
13 export class HomeComponent {
14
15   private initialized: boolean = false;
16   private redirectUrl: string;
17   private authorizationUrl: string;
18   private clientId: string;
19
20   private accessToken: string;
21   private refreshToken: string;
22   private expiresIn: string;
23   private userName: string;
24
25   private resultOfProtectedAPI;
26
27   onButtonLoginClicked(): void {
28     if (this.initialized == true) {
29       let url = this.authorizationUrl + "?scope=email";
30       url = url + "&response_type=" + "code";
31       url = url + "&client_id=" + this.clientId;
32       url = url + "&access_type=" + "offline";
33       url = url + "&redirect_uri=" + this.redirectUrl;
34       window.location.href = url;
35     }
36   }
37 }
```

Angular Code to invoke the login

@heidloff

VS master 0 0 1 Ln 6, Col 18 Spaces: 2 UTF-8 LF TypeScript 2.3.4

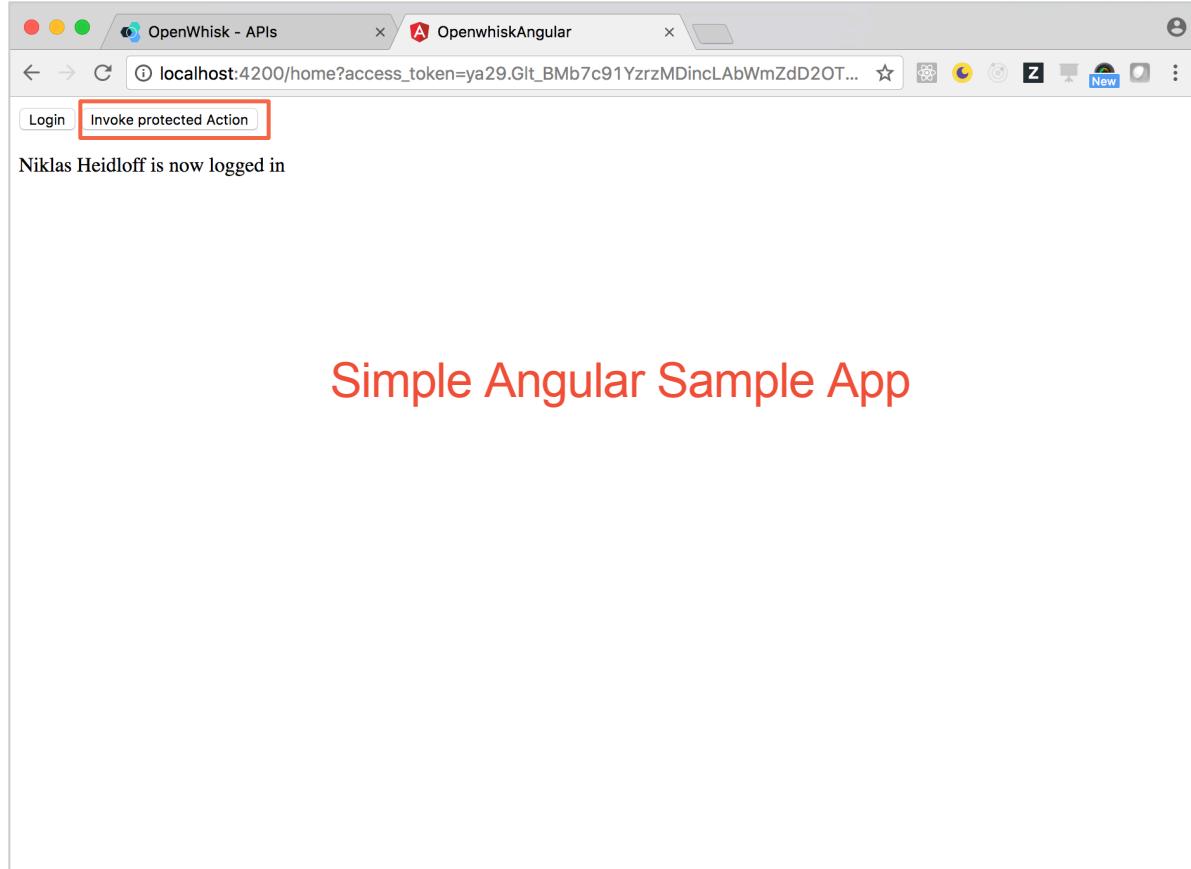
# Google Authentication OAuth Dance



Demo

@nheidloff

# Demo



# Demo

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure. The file `home.component.ts` is selected in the "OPEN EDITORS" section.
- Editor View:** Displays the code for `home.component.ts`. A red box highlights the following code block:

```
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
```

```
/*
curl --request GET \
--url https://service.us.apiconnect.ibmcloud.com/gws/apigateway/api/a172f687d4c7
--header 'accept: application/json' \
--header 'authorization: Bearer REPLACE_BEARER_TOKEN'
*/
```

```
onButtonInvokeActionClicked(): void {
  let url: string = "https://service.us.apiconnect.ibmcloud.com/gws/apigateway/a172f687d4c7";

  let headers = new Headers({
    'Content-Type': 'application/json',
    'authorization': 'Bearer ' + this.accessToken
  });

  let options = new RequestOptions({ headers: headers });

  this.http.get(url, options)
    .map(res => res.json())
    .subscribe(
      result => {
        console.log(result)
        this.resultOfProtectedAPI = JSON.stringify(result, null, 2);
      },
      err => {
        console.error(err);
        this.resultOfProtectedAPI = JSON.stringify(err._body, null, 2);
      });
}
```

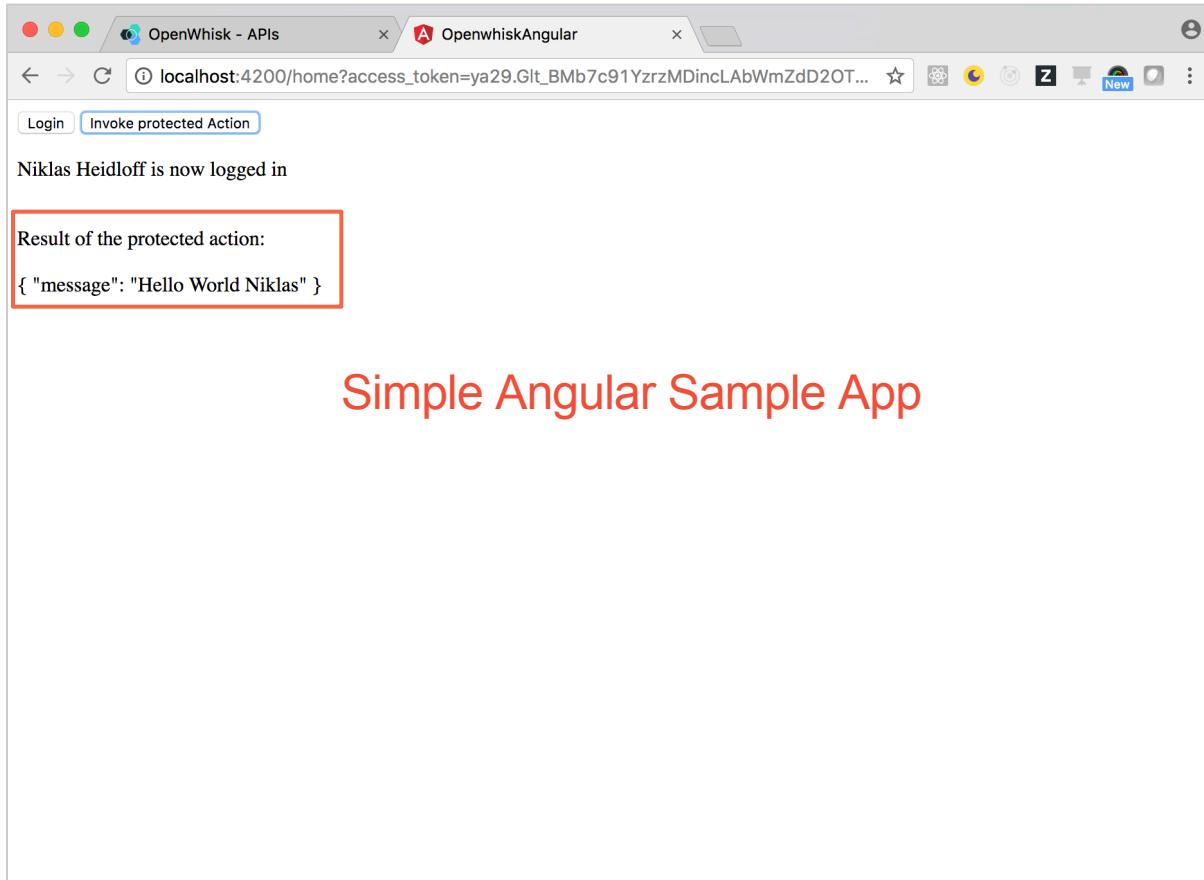
```
constructor(
  private http: Http,
  private route: ActivatedRoute,
  private router: Router) {
```

```
ngOnInit(): void {
```

- Status Bar:** Shows the current file is `home.component.ts`, the editor version is `openwhisk-serverless-webapp`, and the status bar includes "Ln 6, Col 18", "Spaces: 2", "UTF-8", "LF", "TypeScript", "2.3.4", and a smiley face icon.

**Angular Code to invoke Action**

# Demo



# Discussion

[github.com/nheidloff/openwhisk-serverless-webapp](https://github.com/nheidloff/openwhisk-serverless-webapp)  
[heidloff.net](http://heidloff.net)  
[bluemix.net/openwhisk](http://bluemix.net/openwhisk)



Niklas Heidloff, IBM  
Developer Advocate  
[@nheidloff](https://twitter.com/nheidloff)

July 2017