



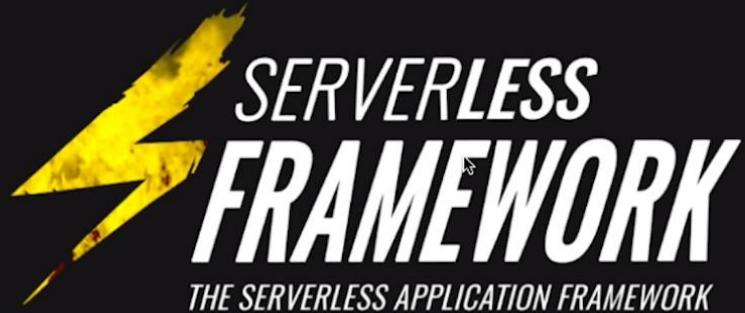
# Lambda Day

Serverless Framework

Austen Collins, Founder - Serverless Inc.

02/25/2016

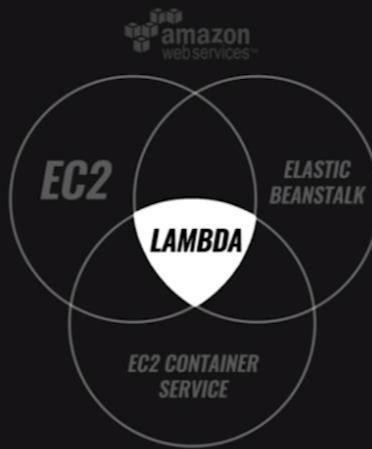
© 2015, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



It's An Arms Race  
Out There



And There Are A  
Lot Of Tools To  
Help You Win!



But Lambda has the  
Lowest Total Cost Of Ownership

## Building Entire Apps With Lambda Is Complex!

1. Lots of functions means managing lots of containers  
(ANARCHY!)
2. Multi-region, multi-stage, multi-developer
3. Resource management (Databases, File Storage, REST API)
4. Code sharing across functions/containers
5. Local development, testing
6. Security
7. Architectural Patterns
8. Optimization



Demo: Install & Create A Project

Serverless Framework – Build and maintain web, mobile and IoT applications running on AWS Lambda and API Gateway (formerly known as JAWS) – <http://www.serverless.com> — Edit

2,061 commits 5 branches 24 releases 41 contributors

Branch: master New pull request

ac360 Merge pull request #685 from MrRio/patch-1 ... Latest commit 8be78a1 a day ago

- bin Use universal /usr/bin/env path instead of /bin/env a month ago
- lib functionCreate: fix error message. Closes #553. 14 days ago
- other/img Readme: animate all the things 7 days ago
- tests Release v0.4.0 15 days ago
- .gitignore All: Rebrand 3 months ago
- .jsbeautifyrc jaws new: add cloudformation and swagger template creation 6 months ago
- .jscsrc deploy changes 4 months ago

This is a NodeJS application with a CLI as below

```
Austens-MacBook-Pro:s_demo austen$ npm install serverless -g
/usr/local/bin/serverless -> /usr/local/lib/node_modules/serverless/bin/serverless
/usr/local/bin/slss -> /usr/local/lib/node_modules/serverless/bin/serverless
/usr/local/bin/sls -> /usr/local/lib/node_modules/serverless/bin/serverless
/usr/local/bin/serverless-run-python-handler -> /usr/local/lib/node_modules/serverless/bin/serverless-run-python-handler
```

```
└─ rimraf@2.5.2 (glob@7.0.0)
└─ node-zip@1.1.1 (jszip@2.5.0)
└─ prompt@0.3.0 (revalidator@0.1.8, pkginfo@0.3.1, read@1.0.7, winston@0.8.3, utile@0.2.1)
└─ moment@2.11.2
└─ aws-sdk@2.2.38 (sax@1.1.5, xml2js@0.4.16, xmlbuilder@4.2.1)
└─ lodash@4.5.1
└─ download@4.4.3 (gulp-rename@1.2.2, object-assign@4.0.1, is-url@1.2.1, readable-stream@2.0.5, stream-combiner2@1.1.1, concat-stream@1.5.1, vinyl@1.1.1, filenamify@1.2.0, ware@1.3.0, each-asynchronous@1.1.1, read-all-stream@3.1.0, got@5.4.1, ca@1.2.0, vinyl-fs@2.3.1, gulp-decompress@1.2.0)
Austens-MacBook-Pro:s_demo austen$ |
```

Now you have **serverless** commands that you can use

The `serverless` or `sls` command above will give you a list of all the things that serverless can do

We can start our serverless project to get a scaffold on our local machine and also create an S3 bucket in our AWS account where it is going to store backups of your environment variables, your CF templates and lambda functions for this project. The CLI will prompt you to select a profile if you have one on your local machine, it will use this along with your credentials to create a ***dev*** stage for you.

This screenshot shows the GitHub repository page for the 'serverless/serverless' project. The repository has 2,061 commits, 5 branches, 24 releases, and 41 contributors. The master branch is selected. A recent pull request from ac360 is listed.

File	Description	Time Ago
bin	Use universal /usr/bin/env path instead of /bin/env	a month ago
lib	functionCreate: fix error message. Closes #553.	14 days ago
other/img	Readme: animate all the things	7 days ago
tests	Release v0.4.0	15 days ago
.gitignore	All: Rebrand	3 months ago
.jsbeautifyrc	jaws new: add cloudformation and swagger template creation	6 months ago
.jscsrc	deploy changes	4 months ago

Let us see the internal architecture of this serverless framework

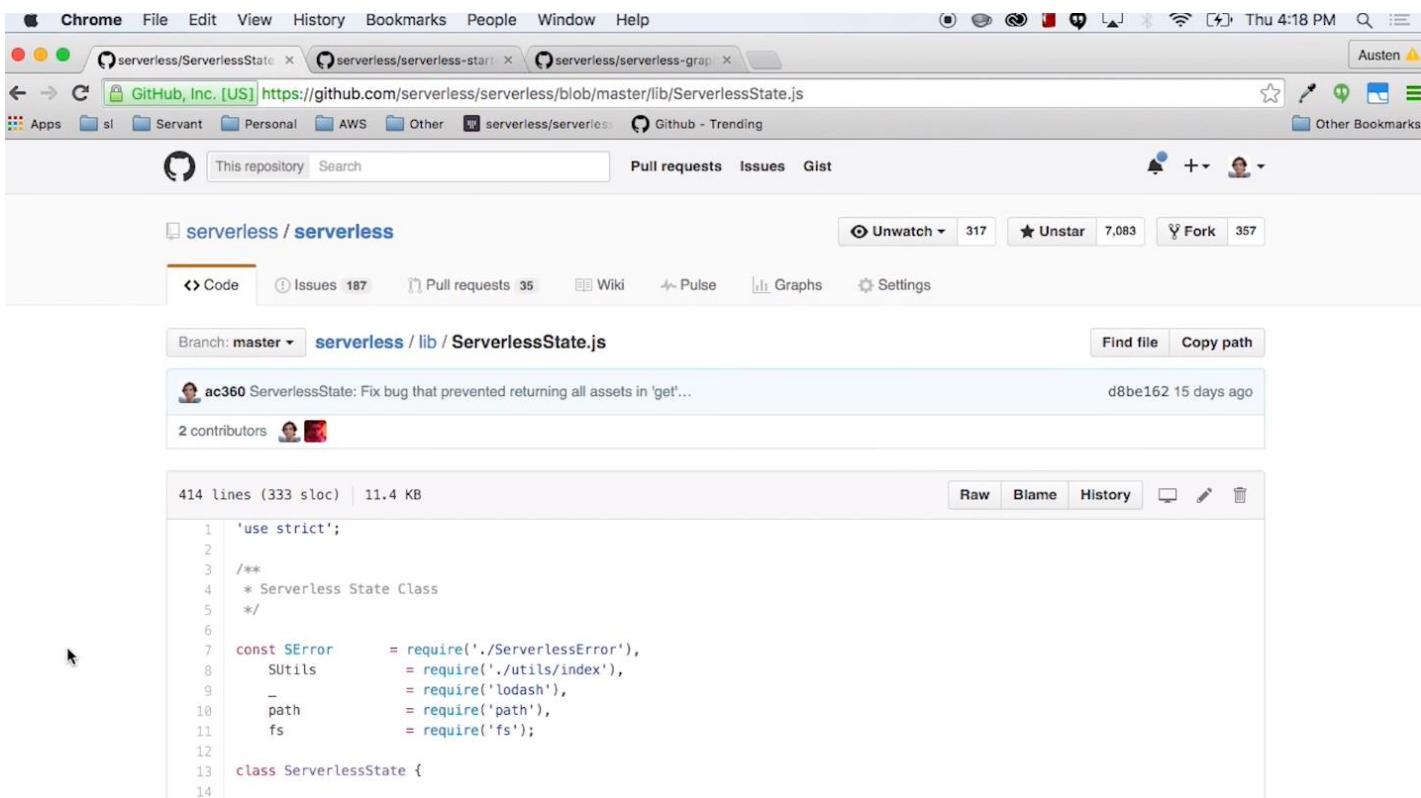
This screenshot shows the GitHub repository page for the 'serverless/serverless' project. The repository has 2,061 commits, 5 branches, 24 releases, and 41 contributors. The master branch is selected. A recent pull request from ac360 is listed, along with a list of files and their descriptions.

File	Description	Time Ago
bin	Use universal /usr/bin/env path instead of /bin/env	a month ago
lib	functionCreate: fix error message. Closes #553.	14 days ago
other/img	Readme: animate all the things	7 days ago
tests	Release v0.4.0	15 days ago
.gitignore	All: Rebrand	3 months ago
.jsbeautifyrc	jaws new: add cloudformation and swagger template creation	6 months ago
.jscsrc	deploy changes	4 months ago
.travis.yml	Add travisci for jscs and jshint	2 months ago
CONTRIBUTING.md	All: Rebrand	3 months ago
LICENSE.txt	readme:update	3 months ago
README.md	Add Parallax to the list of consultants	2 days ago
package.json	Release v0.4.2	14 days ago

		Latest commit bebedc4 14 days ago
	functionCreate: fix error message. Closes #553.	
..		
	functionCreate: fix error message. Closes #553.	14 days ago
	Component: bump version of serverless-helpers-js	15 days ago
	Replace JAWS references with serverless	15 days ago
	fixing events conflicts	17 days ago
	fixing events after merge	17 days ago
	Classes: fix errors	15 days ago
	fixed dashDeploy	18 days ago
	All: Rebrand	3 months ago
	ServerlessEvent: fix setting from ServerlessFunction.set(), Serverles...	15 days ago
	Classes: fix errors	15 days ago
	ServerlessMeta: skip over hidden files when loading #578	15 days ago
	FunctionCreate: start refactoring to only use sPath	18 days ago
	ResourcesDeploy: Merge resources correctly if using s-resources-cf.j...	14 days ago
	ServerlessState: Fix bug that prevented returning all assets in 'get'...	15 days ago

<https://github.com/serverless/serverless/tree/master/lib/actions>

There is a class file for each of our project assets (starting with Serverless\* above).



The screenshot shows a GitHub repository page for 'serverless / serverless'. The 'Code' tab is selected, showing the 'ServerlessState.js' file. The file has 414 lines (333 sloc) and is 11.4 KB in size. The code is as follows:

```

1  'use strict';
2
3  /**
4   * Serverless State Class
5   */
6
7  const SError      = require('../ServerError'),
8      SUtils        = require('../utils/index'),
9      _              = require('lodash'),
10     path          = require('path'),
11     fs             = require('fs');
12
13  class ServerlessState {
14

```

Each of the asset classes has a ton of useful methods inside them to do various things

```

60     * - Set data from a javascript object
61     */
62
63     set(data) {
64       this.meta = data.meta ? this.meta.set(data.meta) : this.meta;
65       this.project = data.project ? this.project.set(data.project, { deep: true }) : this.project;
66       return this;
67     }
68
69     /**
70      * Set Asset
71      * - Add or replace an asset to the state
72      * - Accepts a class instance of: Project, Component, Function, Endpoint
73      */
74
75     setAsset(data) {
76       if (data instanceof this._S.classes.Project) {
77         this.project = data;
78       } else if (data instanceof this._S.classes.Component) {
79         this.project.components[data.name] = data;
80       } else if (data instanceof this._S.classes.Function) {
81         this.project.components[data._config.sPath.split('/')[0]].functions[data._config.sPath] = data;
82       } else if (data instanceof this._S.classes.Endpoint) {
83         let func = this.project.components[data._config.sPath.split('/')[0]].functions[data._config.sPath.split('@')[0]];
84         let added = false;
85         for (let i = 0; i < func.endpoints.length; i++) {
86           if (func.endpoints[i].path === data.path && func.endpoints[i].method === data.method) {
87             func.endpoints[i] = data;
88             added = true;
89           }
90         }
91         if (!added) func.endpoints.push(data);
92       } else {
93       }
94     }

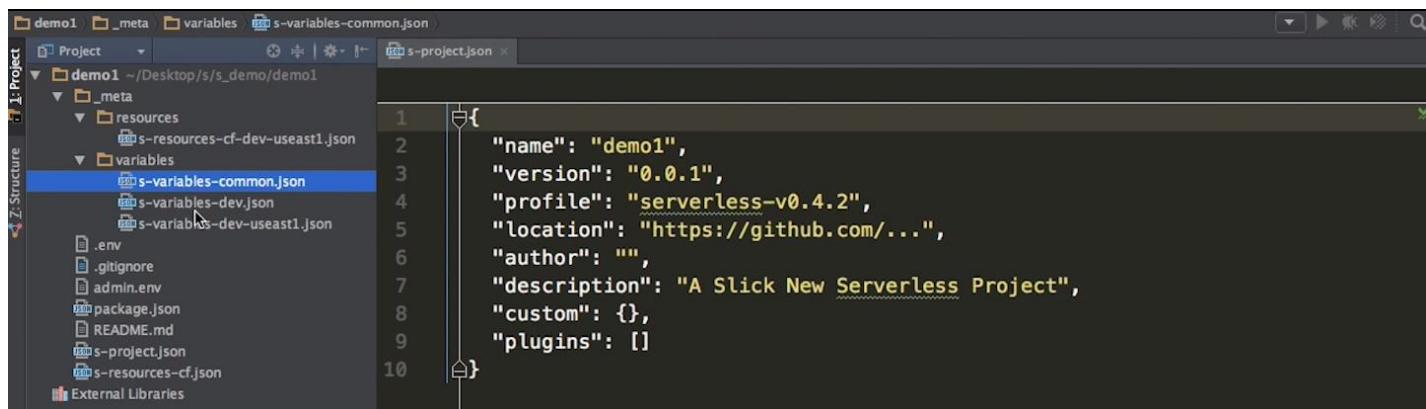
```

You can easily extend any of the methods defined in the files if needed.

..			
<a href="#">CodeDeployLambda.js</a>	ServerlessFunction: Add support for custom IAM Role		15 days ago
<a href="#">CodePackageLambda.js</a>	FunctionDeploy: finish, add templateName, update tests		18 days ago
<a href="#">ComponentCreate.js</a>	componentCreate: take path instead of name		18 days ago
<a href="#">DashDeploy.js</a>	DashDeploy: Allow API Gateway deployment descriptions		15 days ago
<a href="#">DashSummary.js</a>	componentCreate: take path instead of name		18 days ago
<a href="#">EndpointBuildApiGateway.js</a>	Improve a few minor text regions.		16 days ago
<a href="#">EndpointDeploy.js</a>	DashDeploy: Allow API Gateway deployment descriptions		15 days ago
<a href="#">EndpointDeployApiGateway.js</a>	Actions: fix all to pass evt object through, wherever possible, Tests...		2 months ago
<a href="#">EnvGet.js</a>	fixing all actions to use classes		a month ago
<a href="#">EnvList.js</a>	ServerlessState, ServleressMeta: add new helpful methods: getProject,...		a month ago
<a href="#">EnvSet.js</a>	fixing all actions to use classes		a month ago
<a href="#">EnvUnset.js</a>	some fixes for the interactive CLI		a month ago
<a href="#">EventDeploy.js</a>	fixing events after merge		17 days ago
<a href="#">EventDeployS3Lambda.js</a>	event sources: adding permissions		15 days ago
<a href="#">EventDeploySNSSLambda.js</a>	event sources: adding permissions		15 days ago
<a href="#">EventDeployScheduledLambda.js</a>	event sources: adding permissions		15 days ago
<a href="#">EventDeployStreamLambda.js</a>	Fixed Lambda function ARN for all event deploys		15 days ago

Also, within the lib folder, everything that the framework does is divided into actions inside the actions folder. Individual actions are isolated into single files

The deployment is finished and we have our initial resources. Let us take a look at the project structure and files below.



This is the scaffolding that serverless has just created, this file saves our core project configuration. Serverless has a concept of stages and variables, and you are able to specify certain variables for each stage as shown below



These are the variables that are automatically created for you when you create a project. You can take these variables and put them in your project configuration files so that when you create projects assets, then these would work across all your stages in your project.

The screenshot shows a code editor interface with a project structure on the left and code snippets on the right. The project structure includes a 'demo1' folder containing '\_meta' (with 'resources' and 'variables' subfolders), '.env', '.gitignore', 'admin.env', 'package.json', 'README.md', 's-project.json', and 's-resources-cf.json'. The 'variables' folder contains 's-variables-common.json', 's-variables-dev.json', and 's-variables-dev-useast1.json'. The code editor displays the contents of 's-variables-dev-useast1.json':

```
1 "region": "us-east-1",
2 "resourcesStackName": "demo1-dev-r",
3 "iamRoleArnLambda": "arn:aws:iam::149631484542:role/demo1-dev-r-IamRoleLambda"
4
5 }
```

There is also a resources folder, every resource file here is a CF template with the stage and region it is deployed into written in the file name.

The screenshot shows a code editor interface with a project structure on the left and code snippets on the right. The project structure includes a 'demo1' folder containing '\_meta' (with 'resources' and 'variables' subfolders), '.env', '.gitignore', 'admin.env', 'package.json', 'README.md', 's-project.json', and 's-resources-cf.json'. The 'resources' folder contains 's-resources-cf-dev-useast1.json'. The code editor displays the contents of 's-resources-cf-dev-useast1.json':

```
1 "AWSTemplateFormatVersion": "2010-09-09",
2 "Description": "The AWS CloudFormation template for this Serverless application",
3 "Resources": {
4     "IamRoleLambda": {
5         "Type": "AWS::IAM::Role",
6         "Properties": {
7             "AssumeRolePolicyDocument": {
8                 "Version": "2012-10-17",
9                 "Statement": [
10                     {
11                         "Effect": "Allow",
12                         "Principal": {
13                             "Service": [
14                                 "lambda.amazonaws.com"
15                             ]
16                         },
17                         "Action": [
18                             "sts:AssumeRole"
19                         ]
20                     }
21                 ],
22             },
23             "Path": "/"
24         }
25     }
26 }
```

Serverless creates this CF template file out of the box for you, it is an IAM Role with a simple Policy that allows your Lambda to do basic tasks.

```
22     ],
23     },
24     "Path": "/"
25   },
26 },
27 "IamPolicyLambda": {
28   "Type": "AWS::IAM::Policy",
29   "Properties": {
30     "PolicyName": "dev-demo1-lambda",
31     "PolicyDocument": {
32       "Version": "2012-10-17",
33       "Statement": [
34         {
35           "Effect": "Allow",
36           "Action": [
37             "logs:CreateLogGroup",
38             "logs:CreateLogStream",
39             "logs:PutLogEvents"
40           ],
41           "Resource": "arn:aws:logs:us-east-1:*:*
42         }
43       ],
44     }
45   },
46   "Roles": [
```

```
44   ],
45 },
46   "Roles": [
47     {
48       "Ref": "IamRoleLambda"
49     }
50   ],
51 },
52 },
53 "Outputs": {
54   "IamRoleArnLambda": {
55     "Description": "ARN of the lambda IAM role",
56     "Value": {
57       "Fn::GetAtt": [
58         "IamRoleLambda",
59         "Arn"
60       ]
61     }
62   }
63 }
```

If you want to edit your core project resources then there is an **s-resources-cf.json** file that allows you to do that shown below

The screenshot shows the AWS CloudFormation template editor interface. The left sidebar displays the project structure under 'demo1'. The main area shows the content of the 's-resources-cf.json' file:

```
1  {
2      "AWSTemplateFormatVersion": "2010-09-09",
3      "Description": "The AWS CloudFormation template for this Serverless application",
4      "Resources": {
5          "IamRoleLambda": {
6              "Type": "AWS::IAM::Role",
7              "Properties": {
8                  "AssumeRolePolicyDocument": {
9                      "Version": "2012-10-17",
10                 "Statement": [
11                     {
12                         "Effect": "Allow",
13                         "Principal": {
14                             "Service": [
15                                 "lambda.amazonaws.com"
16                             ]
17                         },
18                         "Action": [
19                             "sts:AssumeRole"
20                         ]
21                     }
22                 ],
23             },
24             "Path": "/"
25         }
26     }
27 }
```

This is also a CF template file with the difference that we can use our stage variables in here. There is a command called **serverlessResourcesDeploy** that you can use whenever you add something to your CF template file, this command will deploy the new configuration and save a copy of your CF into the resources folder and then also push a copy to your project's S3 bucket, then it will go deploy the new stack on AWS CF for you.

The screenshot shows the AWS CloudFormation template editor interface. The left sidebar displays the project structure under 'demo1'. The main area shows the content of the 's-resources-cf.json' file, which includes the previous code plus additional content from lines 27 to 32:

```
9      "Version": "2012-10-17",
10     "Statement": [
11         {
12             "Effect": "Allow",
13             "Principal": {
14                 "Service": [
15                     "lambda.amazonaws.com"
16                 ]
17             },
18             "Action": [
19                 "sts:AssumeRole"
20             ]
21         }
22     ],
23     "Path": "/"
24 },
25 },
26 "IamPolicyLambda": {
27     "Type": "AWS::IAM::Policy",
28     "Properties": {
29         "PolicyName": "${stage}-${project}-lambda",
30         "PolicyDocument": {
31             "Version": "2012-10-17".
32         }
33     }
34 }
```

```
29     "Properties": {
30         "PolicyName": "${stage}-${project}-lambda",
31         "PolicyDocument": {
32             "Version": "2012-10-17",
33             "Statement": [
34                 {
35                     "Effect": "Allow",
36                     "Action": [
37                         "logs:CreateLogGroup",
38                         "logs:CreateLogStream",
39                         "logs:PutLogEvents"
40                     ],
41                     "Resource": "arn:aws:logs:${region}::*"
42                 }
43             ],
44             "Roles": [
45                 {
46                     "Ref": "IamRoleLambda"
47                 }
48             ]
49         }
50     }
51 }
52 }
```

```
46         "Ref": "IamRoleLambda"
47     }
48 }
49 ]
50 }
51 }
52 },
53 "Outputs": {
54     "IamRoleArnLambda": {
55         "Description": "ARN of the lambda IAM role",
56         "Value": {
57             "Fn::GetAtt": [
58                 "IamRoleLambda",
59                 "Arn"
60             ]
61         }
62     }
63 }
64 }
```

The ***sls dash summary*** command shows a simple summary of your project.

```
austens-MacBook-Pro:demo1 austen$ sls component create restApi
serverless: Installing "serverless-helpers" for this component via NPM...
serverless: -----
serverless-helpers-js@0.1.0 node_modules/serverless-helpers-js
└── dotenv@1.2.0
serverless: -----
serverless: Successfully created new serverless component: restApi
austens-MacBook-Pro:demo1 austen$ |
```

In our project folder, all the lambda functions are kept in the root folder of the project. These lambda functions are called ***Components***, components define a runtime. Use the `sls component create restApi` command to create a ***component*** called `restAPI`.

The screenshot shows the AWS CloudFormation Designer interface. On the left, the project structure is displayed with a tree view. Under the 'restApi' folder, there is a 'lib' folder containing an 'index.js' file. The main workspace shows the JSON configuration for the 'lib' folder. The code defines an 'IamRoleLambda' resource with an 'AssumeRolePolicyDocument' that allows the lambda service to assume the role. It also defines an 'IamPolicyvLambda' resource.

```
4   "Resources": {
5     "IamRoleLambda": {
6       "Type": "AWS::IAM::Role",
7       "Properties": {
8         "AssumeRolePolicyDocument": {
9           "Version": "2012-10-17",
10          "Statement": [
11            {
12              "Effect": "Allow",
13              "Principal": {
14                "Service": [
15                  "lambda.amazonaws.com"
16                ]
17              },
18              "Action": [
19                "sts:AssumeRole"
12              ]
21            },
22            {
23              "Path": "/"
24            }
25          ],
26        },
27      "IamPolicyvLambda": {
```

In the restApi folder, there is a package.json file for getting dependencies for the component. The lib folder can be used to hold functions that you want to share across all the functions of your component.

The screenshot shows the AWS CloudFormation Designer interface. On the left, the project structure is displayed with a tree view. Under the 'restApi' folder, there is an 's-component.json' file. The main workspace shows the JSON configuration for the 's-component.json' file. It defines a component named 'restApi' with a runtime of 'nodejs' and a custom configuration object.

```
1  {
2   "name": "restApi",
3   "runtime": "nodejs",
4   "custom": {}
```

You also get a s-component.js file shown above

```

Serverless: -----
serverless-helpers-js@0.1.0 node_modules/serverless-helpers-js
└── dotenv@1.2.0
Serverless: -----
Serverless: Successfully created new serverless component: restApi
Austens-MacBook-Pro:demo1 austen$ sls dash summary
-----
|   _ .-----.-.----.----| .-----.-.----.
|  | | | -| | | | | -| | | | -| | | --| | |
|__| | | | | | \ | | | | | | | | | | | | | |
| | | |           The Serverless Application Framework
| | | |           serverless.com, v0.4.2
`-----'

Serverless: 1 stages -----
Serverless:   |_ dev (1 regions)
Serverless:     |_ us-east-1

Serverless: 1 components -----
Serverless:   |_ restApi (0 functions)

Serverless: SUMMARY -----
Serverless: stages      : 1
Serverless: regions     : 1
Serverless: components  : 1
Serverless: functions   : 0
Serverless: endpoints   : 0
Austens-MacBook-Pro:demo1 austen$ |

```

Now we have a component

```

Austens-MacBook-Pro:demo1 austen$ sls function create restApi/users/show
Serverless: Successfully created function: "restApi/users/show"
Austens-MacBook-Pro:demo1 austen$ |

```

We then create our first function called show within a component called users. You can also nest your functions in subfolders.

```

Austens-MacBook-Pro:demo1 austen$ sls dash summary
-----
|   _ .-----.-.----.----| .-----.-.----.
|  | | | -| | | | | -| | | | -| | | --| | |
|__| | | | | | \ | | | | | | | | | | | | |
| | | |           The Serverless Application Framework
| | | |           serverless.com, v0.4.2
`-----'

Serverless: 1 stages -----
Serverless:   |_ dev (1 regions)
Serverless:     |_ us-east-1

Serverless: 1 components -----
Serverless:   |_ restApi (1 functions)
Serverless:     |_ show (1 endpoints)
Serverless:       |_ GET - users/show

Serverless: SUMMARY -----
Serverless: stages      : 1
Serverless: regions     : 1
Serverless: components  : 1
Serverless: functions   : 1
Serverless: endpoints   : 1
Austens-MacBook-Pro:demo1 austen$ |

```

The summary now says we have 1 function and 1 endpoint.

The screenshot shows a code editor with a dark theme. On the left is a project tree for a 'demo1' project. The 'restApi' component contains a 'users' folder with a 'show' folder. Inside 'show' are 'event.json', 'handler.js', and 's-function.json'. Other files in the root include '.env', '.gitignore', 'admin.env', 'package.json', 's-component.json', 'README.md', 's-project.json', and 's-resources-cf.json'. The right pane shows the content of 'handler.js':

```
'use strict';

/**
 * Serverless Module: Lambda Handler
 * - Your lambda functions should be a thin wrapper around your own separate
 * modules, to keep your code testable, reusable and AWS independent
 * - 'serverless-helpers-js' module is required for Serverless ENV var support
 */

// Require Serverless ENV vars
var ServerlessHelpers = require('serverless-helpers-js').loadEnv();

// Require Logic
var lib = require('../lib');

// Lambda Handler
module.exports.handler = function(event, context) {
    lib.respond(event, function(error, response) {
        return context.done(error, response);
    });
};
```

Here is our project scaffolding once again, we now have a **users** folder within our **restApi component** that contains a **show** folder for our endpoint. The **handler.js file is our JS lambda function for the show endpoint folder.**

The screenshot shows a code editor with a dark theme. The project tree is identical to the previous one. The right pane shows the content of 'event.json':

```
{}
```

The **event.json file** contains an object can be used to hold data that we want to populate the handler.js function with when we want to run the function locally.

The screenshot shows a code editor interface with a project structure on the left and the content of the `s-function.json` file on the right. The project structure includes files like `.env`, `.gitignore`, `admin.env`, `package.json`, `README.md`, `s-project.json`, and `s-resources-cf.json`. The `s-function.json` file contains the following JSON configuration:

```
1 "name": "show",
2 "customName": false,
3 "customRole": false,
4 "handler": "users/show/handler.handler",
5 "timeout": 6,
6 "memorySize": 1024,
7 "custom": {
8     "excludePatterns": [],
9     "envVars": []
10 },
11 "endpoints": [
12     {
13         "path": "users/show",
14         "method": "GET",
15         "type": "AWS",
16         "authorizationType": "none",
17         "apiKeyRequired": false,
18         "requestParameters": {},
19         "requestTemplates": {
20             "application/json": ""
21         },
22         "responses": {
23             "400": {
24                 "statusCode": "400"
25             }
26         }
27     }
28 ],
29 "default": {
30     "statusCode": "200",
31     "responseParameters": {},
32     "responseModels": {}
33 }
```

The **`s-function.json` file** specifies our lambda function configuration. Note that the endpoints array object contains all the raw parameters that the AWS SDK takes in when you are working with lambda.

This screenshot is similar to the one above, but the `s-function.json` file now contains two entries in the `endpoints` array, indicating multiple endpoints have been added.

```
7     "memorySize": 1024,
8     "custom": {
9         "excludePatterns": [],
10        "envVars": []
11    },
12    "endpoints": [
13        {
14            "path": "users/show",
15            "method": "GET",
16            "type": "AWS",
17            "authorizationType": "none",
18            "apiKeyRequired": false,
19            "requestParameters": {},
20            "requestTemplates": {
21                "application/json": ""
22            },
23            "responses": {
24                "400": {
25                    "statusCode": "400"
26                }
27            }
28        },
29        {
30            "path": "users/show",
31            "method": "POST",
32            "type": "AWS",
33            "authorizationType": "none",
34            "apiKeyRequired": false,
35            "requestParameters": {},
36            "requestTemplates": {
37                "application/json": ""
38            },
39            "responses": {
40                "200": {
41                    "statusCode": "200"
42                }
43            }
44        }
45    ],
46    "default": {
47        "statusCode": "200",
48        "responseParameters": {},
49        "responseModels": {}
50    }
51 }
```

You can add multiple endpoints to your lambda function



```
    "application/json": ""},  
    "responses": {  
        "400": {  
            "statusCode": "400"},  
        "default": {  
            "statusCode": "200",  
            "responseParameters": {},  
            "responseModels": {},  
            "responseTemplates": {  
                "application/json": ""}},  
        }},  
    },  
    "events": []  
}
```

If you want your lambda function to be triggered by an event(s), you can add the list of events to the events array section of the s-function.json file above

```
Austens-MacBook-Pro:demo1 austen$ sls function run restApi/users/show
Serverless: Running restApi/users/show...
Serverless: -----
Serverless: Success! - This Response Was Returned:
Serverless: {"message": "Your Serverless function ran successfully!"}
Austens-MacBook-Pro:demo1 austen$ |
```

We can run our lambda function locally using the `sls function run restApi/users/show` command as above

```
Austens-MacBook-Pro:demo1 austen$ sls dash deploy
-----
|   .-----.-.-----.-.-----.-.-----.
|   | _ | -_ | _ | | _ | _ | | -_ | _ --| _ --| |
|   | _ | _ | _ | \_ /| _ | _ | | _ | _ | _ | _ |
|   |   |   |   |   |   |   |   |   |   |   |   |
|   |       The Serverless Application Framework
|   |           serverless.com, v0.4.2
`-----'

Use the <up>, <down>, <pageup>, <pagedown>, <home>, and <end> keys to navigate.
Press <enter> to select/deselect, or <space> to select/deselect and move down.
Press <ctrl> + a to select all, and <ctrl> + d to deselect all.
Press <ctrl> + f to select all functions, and <ctrl> + e to select all endpoints.
Press <ctrl> + <enter> to immediately deploy selected.
Press <escape> to cancel.
```

```
Serverless: Select the assets you wish to deploy:  
  restApi/users/show  
>  function - restApi/users/show  
    endpoint - restApi/users/show@users/show~GET  
- - - - -  
Deploy  
Cancel
```

We have a separate dashboard for whenever we want to deploy our project assets, we can use the `sls dash deploy` command as above. This will give another summary of all the assets in your project.

```
Use the <up>, <down>, <pageup>, <pagedown>, <home>, and <end> keys to navigate.  
Press <enter> to select/deselect, or <space> to select/deselect and move down.  
Press <ctrl> + a to select all, and <ctrl> + d to deselect all.  
Press <ctrl> + f to select all functions, and <ctrl> + e to select all endpoints.  
Press <ctrl> + <enter> to immediately deploy selected.  
Press <escape> to cancel.
```

```
Serverless: Select the assets you wish to deploy:
```

```
restApi/users/show  
  function - restApi/users/show  
  endpoint - restApi/users/show@users/show~GET  
- - - - -  
> Deploy  
Cancel
```

```
Serverless: Deploying functions in "dev" to the following regions: us-east-1
```

```
Serverless: -----
```

```
Serverless: Successfully deployed functions in "dev" to the following regions:
```

```
Serverless: us-east-1 -----
```

```
Serverless:   restApi/users/show (demo1-restApi-show): arn:aws:lambda:us-east-1:149631484542:function:demo1-restApi-show:dev
```

```
Serverless: Deploying endpoints in "dev" to the following regions: us-east-1
```

```
Serverless: Successfully deployed endpoints in "dev" to the following regions:
```

```
Serverless: us-east-1 -----
```

```
Serverless:   GET - users/show - https://95ds8qgyj3.execute-api.us-east-1.amazonaws.com/dev/users/show
```

```
Austens-MacBook-Pro:demo1 austen$ |
```

You can **toggle through the options shown in the CLI** and hit the **Deploy** command to have that asset deployed individually to your AWS account as above. You can use **serverless function deploy-all** command to deploy all the functions for you, and **serverless endpoint deploy-all** command to deploy all your endpoints to AWS. You then get back the **arn** for the functions or endpoints that were deployed, **serverless then attaches all the endpoints to their respective functions and gives back a URL that you can use to trigger all the lambda functions.**



The browser URL runs our lambda functions as above

Chrome File Edit View History Bookmarks People Window Help

Thu 4:31 PM

GitHub, Inc. [US] https://github.com/serverless/serverless/blob/master/lib/ServerlessState.js

Unwatch 317 ★ Unstar 7,083 Fork 357

Code Issues 187 Pull requests 35 Wiki Pulse Graphs Settings

Branch: master serverless / lib / ServerlessState.js

ac360 ServerlessState: Fix bug that prevented returning all assets in 'get'... d8be162 15 days ago

2 contributors

414 lines (333 sloc) | 11.4 KB

```

1 'use strict';
2
3 /**
4  * Serverless State Class
5 */
6
7 const SError      = require('./ServerError'),
8     SUtils        = require('./utils/index'),
9     _              = require('lodash'),
10    path          = require('path'),
11    fs             = require('fs');
12
13 class ServerlessState {
14

```

Chrome File Edit View History Bookmarks People Window Help

Thu 4:31 PM

GitHub, Inc. [US] https://github.com/serverless/serverless

Unwatch 317 ★ Unstar 7,084 Fork 357

Code Issues 187 Pull requests 35 Wiki Pulse Graphs Settings

Serverless Framework – Build and maintain web, mobile and IoT applications running on AWS Lambda and API Gateway (formerly known as JAWS) – <http://www.serverless.com> — Edit

2,061 commits 5 branches 24 releases 41 contributors

New pull request New file Upload files Find file HTTPS https://github.com/serv Download ZIP

ac360 Merge pull request #685 from MrRio/patch-1 ... Latest commit 8be78a1 a day ago

	bin	Use universal /usr/bin/env path instead of /bin/env	a month ago
		functionCreate: fix error message. Closes #553.	14 days ago
	other/img	Readme: animate all the things	7 days ago
	tests	Release v0.4.0	15 days ago
	.gitignore	All: Rebrand	3 months ago
	.jsbeautifyrc	jaws new: add cloudformation and swagger template creation	6 months ago
	.jscsrc	deploy changes	4 months ago

Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless/tree/master/lib/actions

Apps Servant Personal AWS Other serverless/serverless Github - Trending

This repository Search Pull requests Issues Gist

serverless / serverless

Code Issues Pull requests Wiki Pulse Graphs Settings

Branch: master serverless / lib /

New file Upload files Find file History

 eahefnawy functionCreate: fix error message. Closes #553. Latest commit bebedc4 14 days ago

..

 actions functionCreate: fix error message. Closes #553. 14 days ago

 templates Component: bump version of serverless-helpers-js 15 days ago

 utils Replace JAWS references with serverless 15 days ago

 Actions.json fixing events conflicts 17 days ago

 Serverless.js fixing events after merge 17 days ago

 ServerlessComponent.js Classes: fix errors 15 days ago

 ServerlessEndpoint.js fixed dashDeploy 18 days ago

 ServerlessError.js All: Rebrand 3 months ago

 ServerlessEvent.js ServerlessEvent: fix setting from ServerlessFunction.set(), Serverles... 15 days ago

Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless/tree/master/lib/actions

Apps Servant Personal AWS Other serverless/serverless Github - Trending

This repository Search Pull requests Issues Gist

serverless / serverless

Code Issues Pull requests Wiki Pulse Graphs Settings

Branch: master serverless / lib / actions /

New file Upload files Find file History

 eahefnawy functionCreate: fix error message. Closes #553. Latest commit bebedc4 14 days ago

..

 CodeDeployLambda.js ServerlessFunction: Add support for custom IAM Role 15 days ago

 CodePackageLambda.js FunctionDeploy: finish, add templateName, update tests 18 days ago

 ComponentCreate.js componentCreate: take path instead of name 18 days ago

 DashDeploy.js DashDeploy: Allow API Gateway deployment descriptions 15 days ago

 DashSummary.js componentCreate: take path instead of name 18 days ago

 EndpointBuildApiGateway.js Improve a few minor text regions. 16 days ago

 EndpointDeploy.js DashDeploy: Allow API Gateway deployment descriptions 15 days ago

 EndpointDeployApiGateway.js Actions: fix all to pass evt object through, wherever possible, Tests... 2 months ago

 EnvGet.js fixing all actions to use classes a month ago

Anyone can write a function for serverless that replaces one of the core actions and have the new implementation of the action do something custom to their logic or need. You can also add hooks to an action, a hook is a logic that fires before or after an action is fired.

The screenshot shows a Chrome browser window with several tabs open. The active tab is 'serverless/serverless' at https://github.com/serverless/serverless. Other tabs include 'serverless/serverless-start', 'serverless/serverless-graph', 'https://95ds8qgyj3.execute...', and 'serverless/serverless'. The address bar shows 'GitHub, Inc. [US] https://github.com/serverless/serverless'. The page content is about 'Plugins'.

## Plugins

Serverless is composed of Plugins. A group of default Plugins ship with the Framework, and here are some others you can add to improve/help your workflow:

- [Plugin Boilerplate](#) - Make a Serverless Plugin with this simple boilerplate.
- [Serve](#) - Simulate API Gateway locally, so all function calls can be run via localhost.
- [Offline](#) - An alternative to the Serve plugin.
- [Alerting](#) - This Plugin adds Cloudwatch Alarms with SNS notifications for your Lambda functions.
- [Optimizer](#) - Optimizes your code for performance in Lambda.
- [CORS](#) - Adds support for CORS (Cross-origin resource sharing).
- [CloudFormation Validator](#) - Adds support for validating your CloudFormation template.
- [Prune](#) - Delete old versions of AWS lambdas from your account so that you don't exceed the code storage limit.
- [Base-Path](#) - Sets a base path for all API Gateway endpoints in a Component.
- [Test](#) - A Simple Integration Test Framework for Serverless.
- [SNS Subscribe](#) - This plugin easily subscribes your lambda functions to SNS notifications.
- [JSHint](#) - Detect errors and potential problems in your Lambda functions.
- [Webpack](#) - Use Webpack to optimize your Serverless Node.js Functions
- [Serverless Client](#) - Deploy and config a web client for your Serverless project to S3.

There are now many custom plugins for several actions written by the community. Many of the plugins have custom steps that fires before the action fires. The optimizer plugin is for NodeJS and it browsifies your JS code and extracts the code you really need and minifies it into a ZIP folder before it uploads it to the S3 bucket.

## Demo: Components, Functions & Endpoints

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with 'Project' and 'Structure' sections. Under 'Project', there's a tree view with 'demo1' expanded, showing 'restApi' selected. Under 'Structure', there are files like '.env', '.gitignore', 'admin.env', 'package.json', 'README.md', 's-project.json', 's-resources-cf.json', and 'External Libraries'. The main area shows the code for the 'restApi' component. The code is as follows:

```

1  'use strict';
2
3  /**
4   * Serverless Module: Lambda Handler
5   * - Your lambda functions should be a thin wrapper around your own separate
6   * modules, to keep your code testable, reusable and AWS independent
7   * - 'serverless-helpers-js' module is required for Serverless ENV var support
8  */
9
10 // Require Serverless ENV vars
11 var ServerlessHelpers = require('serverless-helpers-js').loadEnv();
12
13 // Require Logic
14 var lib = require('../lib');
15
16 // Lambda Handler
17 module.exports.handler = function(event, context) {
18
19   lib.respond(event, function(error, response) {
20     return context.done(error, response);
21   });
22};

```

Recall that you can have 1 or more components like the restApi component above in your project. Serverless now supports events that are triggered from lambdas, S3 bucket event, CloudWatch log events, DynamoDB streams, etc.

# Ideal Application Architecture

- RestAPI
- Events
- Utils

It is recommended that your serverless project should have a few components, one for your RestAPI or backend code, another component that just handles Events on your AWS account (like image resizing when an image is uploaded to an S3 bucket, etc), you then need another component for your utility functions for your different applications (i.e lambdas that filter logs or do security audits, etc)

## Possible Code Architectures

### **Nanoservices**

- A Lambda function for each individual REST API endpoint.

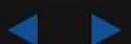
### **Microservices**

- A Lambda function for each resource (e.g., Users) available via multiple endpoints for CRUD.

### **The New Monolithic**

- GraphQL + Lambda

### **Mix Them!**



Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog

Apps Sl Servant Personal AWS Other serverless/serverless... Github - Trending Other Bookmarks

This repository Search Pull requests Issues Gist

serverless / serverless-graphql-blog

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

A Serverless Blog leveraging GraphQL to offer a REST API with only 1 endpoint — Edit

50 commits 2 branches 0 releases 2 contributors

New pull request New file Upload files Find file HTTPS https://github.com/serve...

Branch: master

kevinold add npm install for project level to README; clarify Serverless 0.3.x... 321ea9a 18 days ago

blog move babel-preset-es2015 dependency to project level 27 days ago

.gitignore initial commit a month ago

README.md add npm install for project level to README; clarify Serverless 0.3.x... 18 days ago

package.json update package.json 24 days ago

s-project.json remove mutation param; add introspection query example a month ago

serverless\_graphql\_blog.png README: update 24 days ago

video\_serverless\_blog\_graphql.png add video walkthrough graphic linked to the youtube video 23 days ago

Waiting for raw.githubusercontent.com...

This is a Serverless Blog that uses 3 DynamoDB tables for Authors, Posts, and Comments.

Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog/tree/master/blog

Apps Sl Servant Personal AWS Other serverless/serverless... Github - Trending Other Bookmarks

This repository Search Pull requests Issues Gist

serverless / serverless-graphql-blog

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master serverless-graphql-blog / blog /

New file Upload files Find file History

kevinold move babel-preset-es2015 dependency to project level f5a4845 27 days ago

..

lib add validation via custom graphql types to createPost mutation 28 days ago

resource remove mutation param; add introspection query example a month ago

package.json move babel-preset-es2015 dependency to project level 27 days ago

s-component.json wip - get basic post schema working a month ago



This screenshot shows a GitHub repository page for 'serverless/serverless-graphql-blog'. The repository name is 'serverless / serverless-graphql-blog'. The current branch is 'master'. The path shown is 'serverless-graphql-blog / blog / resource /'. There are 4 issues and 0 pull requests. The commit history shows three commits from 'kevinold' and one from 'graphql'. All commits are dated 'a month ago'. The commits are:

- remove mutation param; add introspection query example
- create blog component, resource module, graphql function/endpoint
- create blog component, resource module, graphql function/endpoint

The latest commit is '3eb672d' on Jan 26.

This screenshot shows the same GitHub repository page, but the path is now 'serverless-graphql-blog / blog / resource / graphql /'. The commit history shows three commits from 'kevinold' and one from 'event.json'. All commits are dated 'a month ago'. The commits are:

- remove mutation param; add introspection query example
- create blog component, resource module, graphql function/endpoint
- update to use ES6 for graphql
- remove mutation param; add introspection query example

The latest commit is '3eb672d' on Jan 26.

It also has just 1 Lambda function that has 1 endpoint ***resource/graphql***

Chrome File Edit View History Bookmarks People Window Help

serverless-graphql-blog/s serverless/serverless-start serverless/serverless-graph https://95ds8qgy3.execute... Asten

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog/blob/master/blog/resource/graphql/s-function.json Apps Servant Personal AWS Other serverless/serverless... Github - Trending Other Bookmarks

serverless / serverless-graphql-blog

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master serverless-graphql-blog / blog / resource / graphql / s-function.json Find file Copy path

kevinold remove mutation param; add introspection query example 3eb672d on Jan 26

1 contributor

36 lines (35 sloc) | 772 Bytes Raw Blame History

```
1 {
2   "name": "graphql",
3   "handler": "resource/graphql/handler.handler",
4   "runtime": "nodejs",
5   "timeout": 6,
6   "memorySize": 1024,
7   "custom": {
8     "excludePatterns": [],
9     "envVars": []
10 },
11   "endpoints": [
12     {
13       "path": "resource/graphql",
14       "method": "POST",
15       "authorizationType": "none",
16       "apiKeyRequired": false
17     }
18   ]
19 }
```

Chrome File Edit View History Bookmarks People Window Help

serverless-graphql-blog/s serverless/serverless-start serverless/serverless-graph https://95ds8qgy3.execute... Asten

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog/blob/master/blog/resource/graphql/s-function.json Apps Servant Personal AWS Other serverless/serverless... Github - Trending Other Bookmarks

```
2 {
3   "name": "graphql",
4   "handler": "resource/graphql/handler.handler",
5   "runtime": "nodejs",
6   "timeout": 6,
7   "memorySize": 1024,
8   "custom": {
9     "excludePatterns": [],
10    "envVars": []
11  },
12  "endpoints": [
13    {
14      "path": "resource/graphql",
15      "method": "POST",
16      "authorizationType": "none",
17      "apiKeyRequired": false,
18      "requestParameters": {},
19      "requestTemplates": {
20        "application/json": "{\"query\": $input.json(\"$\")}"
21      },
22      "responses": {
23        "400": {
24          "statusCode": "400"
25        },
26        "default": {
27          "statusCode": "200",
28          "responseParameters": {},
29          "responseModels": {},
30          "responseTemplates": {},
31          "application/json": ""
32        }
33      }
34    }
35  ]
36 }
```

You can send a complex data shape to this API as a single request.

A screenshot of a Chrome browser window. The address bar shows the URL <https://github.com/serverless/serverless-graphql-blog/blob/master/blog/lib/index.js>. The page content is a GitHub pull request for the repository 'serverless-graphql-blog'. The pull request details are as follows:

- Author: kevinold
- Title: add GraphQL/GraphQL introspectionQuery support
- Date: 45afff3c 29 days ago
- Contributors: 1 contributor
- File: lib/index.js (20 lines, 14 sloc, 466 Bytes)
- Description: The code handles GraphQL queries by patching the event object to allow queries from GraphiQL and then executing the query using graphql().then(). The code is as follows:

```
1 import { graphql } from 'graphql';
2 import Schema from './schema';
3
4 export function runGraphQL(event, cb) {
5
6   let query = event.query;
7
8   // patch to allow queries from GraphiQL
9   // like the initial introspectionQuery
10  if (event.query && event.query.hasOwnProperty('query')) {
11    query = event.query.query.replace("\n", ' ', "g");
12  }
13
14  graphql(Schema, query).then( function(result) {
15    //console.log('RESULT: ', result);
16    return cb(null, result);
17  });
18}
19}
```

**GraphQL sits in your lambda function** and it splits up the requests into 1 or more parallel requests to the 3 different DynamoDB tables and then aggregates the responses and sends it back out to your client in the shape/format it is expecting the data to be in.

A screenshot of a Chrome browser window. The address bar shows the URL <https://github.com/serverless/serverless-graphql-blog>. The page content is the GitHub repository page for 'serverless / serverless-graphql-blog'. The repository details are as follows:

- Owner: serverless
- Name: serverless-graphql-blog
- Branch: master
- Issues: 4
- Pull requests: 0
- Forks: 3
- Commits: 3 (latest commit d20d2a9 28 days ago)

The commit history shows the following recent commits:

- kevinold add validation via custom graphql types to createPost mutation (28 days ago)
- fix project name variable (a month ago)
- add GraphQL/GraphQL introspectionQuery support (29 days ago)

Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog/blob/master/blog/lib/schema.js

Raw Blame History

```
117 lines (107 sloc) | 2.56 KB
```

```
1 import {
2   GraphQLObjectType,
3   GraphQLSchema,
4   GraphQLList,
5   GraphQLString,
6   GraphQLNonNull
7 } from 'graphql';
8
9 import {
10   GraphQLLimitedString
11 } from 'graphql-custom-types';
12
13 import { getPosts, getAuthor, getAuthors, getComments, createPost } from './dynamo';
14
15 const Author = new GraphQLObjectType({
16   name: "Author",
17   description: "Author of the blog post",
18   fields: () => ({
19     id: {type: GraphQLString},
20     name: {type: GraphQLString}
21   })
22 });
23
24 const Comment = new GraphQLObjectType({
25   name: "Comment",
26   description: "Comment on the blog post",
27   fields: () => ({
28     id: {type: GraphQLString},
29     content: {type: GraphQLString},
30   })
31 });
32
33 const Post = new GraphQLObjectType({
34   name: "Post",
35   description: "A blog post",
36   fields: () => ({
37     id: {type: GraphQLString},
38     title: {type: GraphQLString},
39     author: {type: Author},
40     comments: {type: GraphQLList(Comment)},
41     created: {type: GraphQLString}
42   })
43 });
44
45 const Mutation = new GraphQLObjectType({
46   name: "Mutation",
47   description: "Mutations for creating and deleting posts",
48   fields: () => ({
49     createPost: {type: Post, args: {post: {type: PostInput}}},
50     deletePost: {type: Boolean, args: {id: {type: GraphQLString}}}
51   })
52 });
53
54 const schema = new GraphQLSchema({
55   query: Post,
56   mutation: Mutation
57 });
```

All the schemas are stored in this **blog/lib/schema.js** file

Chrome File Edit View History Bookmarks People Window Help

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-graphql-blog/blob/master/blog/lib/dynamo.js

Raw Blame History

```
kevinoid fix project name variable 4db37cd on Jan 26
```

1 contributor

```
106 lines (90 sloc) | 2.2 KB
```

```
1 import Promise from 'bluebird';
2 import AWS from 'aws-sdk';
3 const dynamoConfig = {
4   sessionToken: process.env.AWS_SESSION_TOKEN,
5   region: process.env.AWS_REGION
6 };
7 const docClient = new AWS.DynamoDB.DocumentClient(dynamoConfig);
8 const stage = process.env.SERVERLESS_STAGE;
9 const projectName = process.env.SERVERLESS_PROJECT_NAME;
10 const postsTable = projectName + '-posts-' + stage;
11 const authorsTable = projectName + '-authors-' + stage;
12 const commentsTable = projectName + '-comments-' + stage;
13
14 export function createPost(post) {
15   return new Promise(function(resolve, reject) {
16     var params = {
17       TableName: postsTable,
18       Item: post
19     };
20
21     docClient.put(params, function(err, data) {
22       if (err) return reject(err);
23       return resolve(post);
24     });
25   });
26 }
```

Here is our DynamoDB logic in the **blog/lib/dynamo.js** file,

# Community

## Plugins

## Projects

V0.5

## Serverless-App

A screenshot of a web browser window showing a Gitter chat interface for the 'serverless/serverless' channel. The browser's address bar shows the URL <https://gitter.im/serverless/serverless>. The sidebar on the left lists several users: Austen, Joost Farla, Erik Erikson, Jeremy McJunkin, Jared Short, Kamil Burzynski, Philipp Muens, Ryan Brown, Kurt Freytag, satyanarayana-1, Alexandre Saiz Verdaguer, and Nick. A green button at the bottom of the sidebar says 'CREATE A ROOM'. The main chat area shows messages from users:

- @staceymoore awesome, thanks! (15:16)
- boushley @staceymoore great idea. And my bad for mixing up the only two terms 😊 I meant event not context (15:16)
- staceymoore @boushley @dijitalmunku\_ The other thing I find handy is adding an event.meta object when the method starts running. I set that up by deep cloning a plain json template and I use it to store any vars that need to be accessible when calling different lib and utility functions. I can usually just call `doSomethingUseful(event)` and keep the code more DRY and readable. Works for me, lol. As you know, everyone has a different approach. (15:25)
- boushley Everyone definitely has different approaches, but it's great to see what everyone does. Gives us ideas of how to improve our own approach 😊 (15:35)
- staceymoore @boushley I've gotten a lot of great ideas and resources in this chat. There are many talented and helpful people in the Serverless community. (15:46)

The interface also includes sections for 'People' (listing 616 members) and 'Activity' (listing recent comments). A 'Gitter Support' button is located at the bottom right.

Chrome File Edit View History Bookmarks People Window Help

Thu 4:52 PM

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-starter

serverless/serverless-starter

serverless/serverless-graph

https://95ds8qgyj3.execute... serverless/serverless -

Apps Servant Personal AWS Other serverless/serverless GitHub - Trending Other Bookmarks

This repository Search Pull requests Issues Gist

serverless / serverless-starter

Unwatch 4 ★ Unstar 40 Fork 8

Code Issues 2 Pull requests 1 Wiki Pulse Graphs Settings

A boilerplate for new Serverless Projects — Edit

23 commits 1 branch 8 releases 2 contributors

Branch: master New pull request New file Upload files Find file HTTPS https://github.com/serv... Download ZIP

Austen Collins Release v0.2.0 Latest commit 00c6596 14 days ago

restApi updates to reflect Serverless v.0.4.0 changes 15 days ago

.gitignore temp a month ago

README.md Release v0.2.0 14 days ago

package.json Release v0.1.0 14 days ago

s-project.json updates to reflect Serverless v.0.4.0 changes 15 days ago

s-resources-cf.json updates to reflect Serverless v.0.4.0 changes 15 days ago

s-templates.json Updates: Better template usage, give an example of a path param in an... a month ago

Chrome File Edit View History Bookmarks People Window Help

Thu 4:52 PM

Austen

GitHub, Inc. [US] https://github.com/serverless/serverless-serverless-graphql-blog/blob/master/blog/lib/dynamo.js

serverless/serverless-graphql-blog

serverless/serverless-graph

https://95ds8qgyj3.execute... serverless/serverless -

Apps Servant Personal AWS Other serverless/serverless GitHub - Trending Other Bookmarks

This repository Search Pull requests Issues Gist

serverless / serverless-graphql-blog

Unwatch 12 ★ Unstar 200 Fork 3

Code Issues 4 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master serverless-graphql-blog / blog / lib / dynamo.js Find file Copy path

kevinold fix project name variable 4db37cd on Jan 26

1 contributor

106 lines (90 sloc) | 2.2 KB Raw Blame History

```
1 import Promise from 'bluebird';
2 import AWS from 'aws-sdk';
3 const dynamoConfig = {
4   sessionToken: process.env.AWS_SESSION_TOKEN,
5   region: process.env.AWS_REGION
6 };
7 const docClient = new AWS.DynamoDB.DocumentClient(dynamoConfig);
8 const stage = process.env.SERVERLESS_STAGE;
9 const projectName = process.env.SERVERLESS_PROJECT_NAME;
10 const postsTable = projectName + '-posts-' + stage;
11 const authorsTable = projectName + '-authors-' + stage;
12 const commentsTable = projectName + '-comments-' + stage;
13
14 export function createPost(post) {
```

A screenshot of a web browser displaying the GitHub repository for the "serverless" framework. The repository has 7,084 stars and 357 forks. It features a lightning bolt icon and a bar chart.

A screenshot of a web browser displaying the "Plugins" section of the Serverless documentation. It lists various third-party plugins:

- [Plugin Boilerplate](#) - Make a Serverless Plugin with this simple boilerplate.
- [Serve](#) - Simulate API Gateway locally, so all function calls can be run via localhost.
- [Offline](#) - An alternative to the Serve plugin.
- [Alerting](#) - This Plugin adds Cloudwatch Alarms with SNS notifications for your Lambda functions.
- [Optimizer](#) - Optimizes your code for performance in Lambda.
- [CORS](#) - Adds support for CORS (Cross-origin resource sharing).
- [CloudFormation Validator](#) - Adds support for validating your CloudFormation template.
- [Prune](#) - Delete old versions of AWS lambdas from your account so that you don't exceed the code storage limit.
- [Base-Path](#) - Sets a base path for all API Gateway endpoints in a Component.
- [Test](#) - A Simple Integration Test Framework for Serverless.
- [SNS Subscribe](#) - This plugin easily subscribes your lambda functions to SNS notifications.
- [JSHint](#) - Detect errors and potential problems in your Lambda functions.
- [Webpack](#) - Use Webpack to optimize your Serverless Node.js Functions
- [Serverless Client](#) - Deploy and config a web client for your Serverless project to S3.

## Contributing

The ***Serve*** plugin runs a local serverless server so that you can run and test your serverless functions locally before deploying to AWS

The screenshot shows a Google Chrome browser window with the following details:

- Address Bar:** https://github.com/serverless/serverless
- Toolbar:** Shows icons for Back, Forward, Stop, Refresh, and a search bar.
- Header:** Chrome, File, Edit, View, History, Bookmarks, People, Window, Help. A user icon for "Austen" is also present.
- Bookmarks Bar:** Contains links to "serverless/serverless: Serv", "serverless/serverless-star", "serverless/serverless-graph", "https://95ds8qgyj3.execute..., serverless/serverless", and "GitHub - Trending".
- Left Sidebar:** A sidebar with a "Facebook" link and a "Twitter" link.
- Content Area:**
  - ## Projects

Serverless Projects are shareable and installable. You can publish them to npm and install them via the Serverless Framework CLI by using `$ serverless project install <project-name>`

    - [serverless-starter](#) - A simple boilerplate for new projects with a few architectural options.
    - [serverless-graphql-blog](#) - A blog boilerplate that leverages GraphQL in front of DynamoDB to offer a minimal REST API featuring only 1 endpoint.
  - ## Plugins

Serverless is composed of Plugins. A group of default Plugins ship with the Framework, and here are some others you can add to improve/help your workflow:

    - [Plugin Boilerplate](#) - Make a Serverless Plugin with this simple boilerplate.
    - [Serve](#) - Simulate API Gateway locally, so all function calls can be run via localhost.
    - [Offline](#) - An alternative to the Serve plugin.
    - [Alerting](#) - This Plugin adds Cloudwatch Alarms with SNS notifications for your Lambda functions.
    - [Optimizer](#) - Optimizes your code for performance in Lambda.

We have also made project files shareable using NPM as a delivery mechanism so that you can use the **`$ serverless project install <project-name>`** to install a serverless project.