AWS Serverless Websites

AngularJS - API Gateway - Lambda - DynamoDB

Create a new S3 Bucket, enable static hosting

Static documents

Route 53 for DNS

Amazon CloudFront for CDN

Bucket: firlumberrally
Region: Oregon
Creation Date: Wed Jan 13 00:01:11 GMT-600 2016
Owner: davidknaack

Permissions

Static Website Hosting

You can host your static website entirely on Amazon S3. Once you enable your bucket for static website hos
your content is accessible to web browsers via the Amazon S3 website endpoint for your bucket.

Endpoint: firlumberrally.s3-website-us-west-2.amazonaws.com

Each bucket serves a website namespace (e.g. "www.example.com"). Requests for your host name (e.g. "example.com" or "www.example.com") can be routed to the contents in your bucket. You can also redirect another host name (e.g. redirect "example.com" to "www.example.com"). See our walkthrough for how to s
Amazon S3 static website with your host name.

Do not enable website hosting

Enable website hosting

Index Document: flr.html

Error Document: err.html

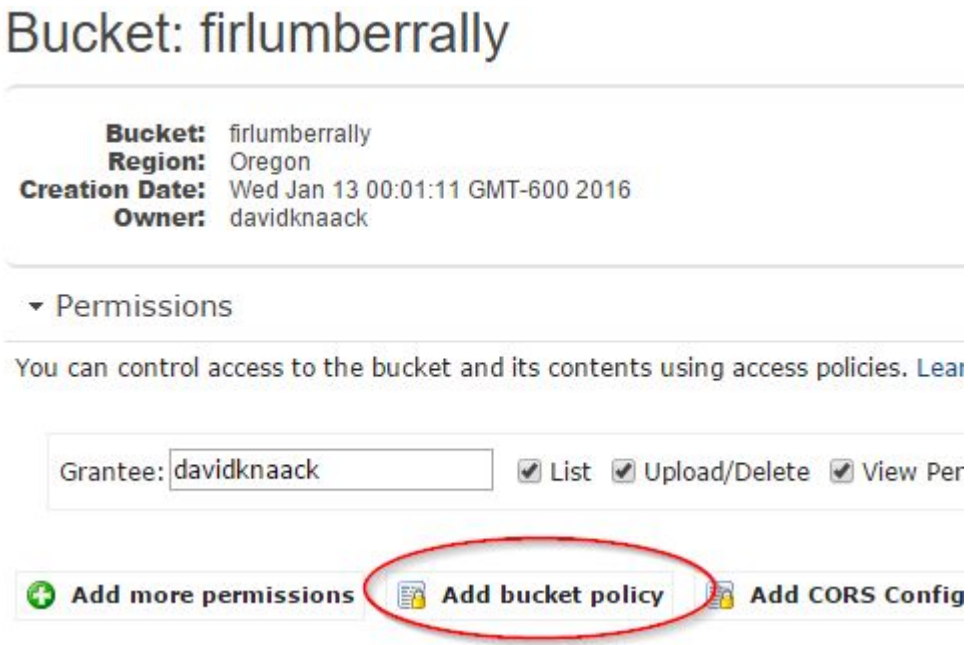
Edit Redirection Rules: You can set custom rules to automatically redirect web page requests for specific content.

Allow Public Access to Bucket

Bucket Properties, Permissions:

Add a bucket policy:

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "Allow Public Access to All Objects",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::firlumberrally/*"
  }
 ]
}
```



Deploy Static Content

Hosting is now enabled, upload some content.

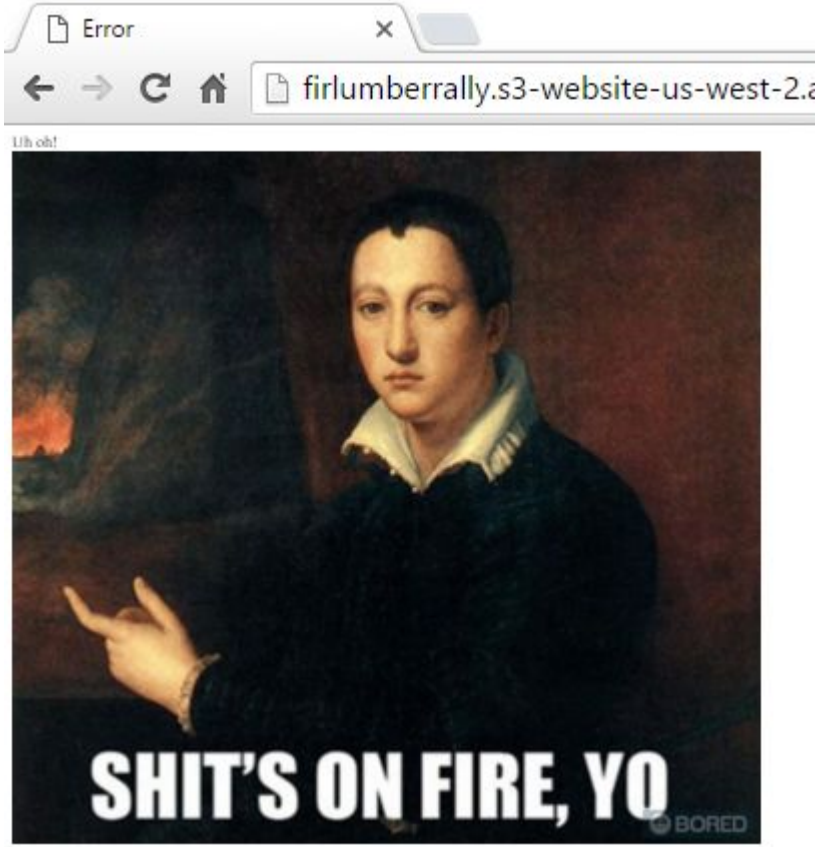
This can be done via the AWS console or AWS cli:

```
<html>
<head><title>Placeholder</title>
<body>Placeholder</body>
</html>
```

And an error page

Just in case anything goes wrong, add a useful error message and upload the static files:

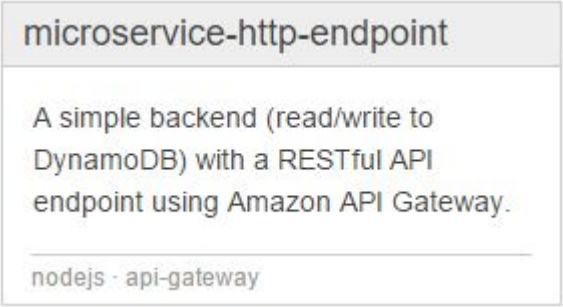
```
<html>
<head><title>Error</title>
<body>Uh oh!<br><img src="shits-on-fire-yo__big_no.jpeg"></body>
</html>
```



Create AWS Lambda RESTful API endpoint

Lambda blueprints make setting up common configurations easy.

Create a new Lambda function using the microservice-http-endpoint blueprint.



Configure the function

No coding required.

DynamoDB-backed CRUD operations are part of the blueprint function.

Function has cases for multiple operations, but each function only uses one.

Configure function

A Lambda function consists of the custom code you want to execute. Learn m

Name* cars-list

Description Return list of cars in the database

Runtime* Node.js

Lambda function code

Provide the code for your function. Use the editor if your code does not requir
libraries, you can upload your code and libraries as a .ZIP file. Learn more ab

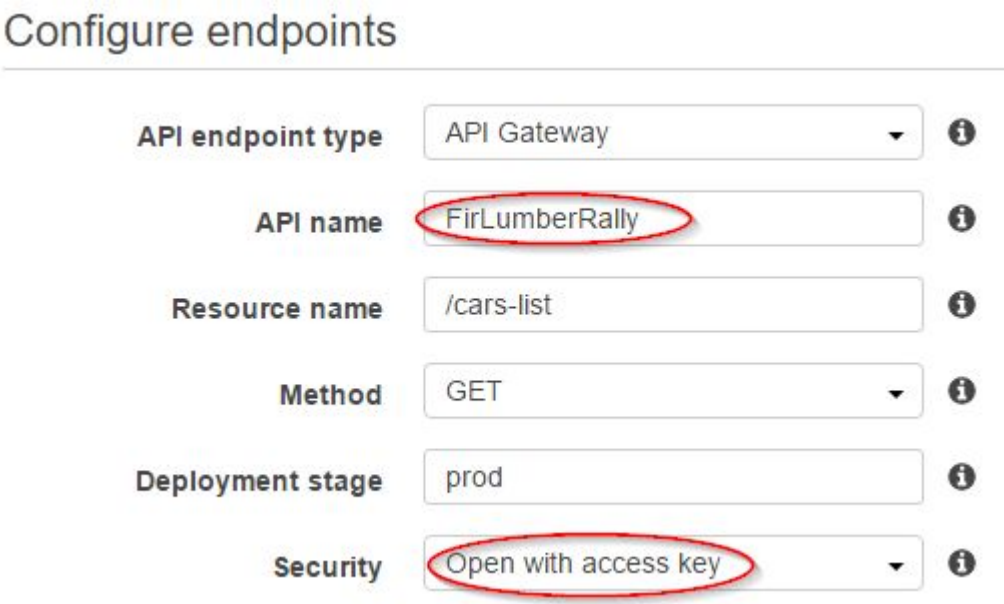
Code entry type Edit code inline Upload a .ZIP file

```
console.log('Loading function');

var doc = require('dynamodb-doc');
var dynamo = new doc.DynamoDB();

/**
 * Provide an event that contains the following keys:
 *
```

Configure Endpoint

Set or choose the API name and the security settings, if desired.

Add an access key in the API Gateway console.



API endpoint is now live and can be called with the API key in an x-api-key header, but the result is an error “Unrecognized operation “undefined””.

The Lambda function expects an ‘event’ parameter containing fields describing what to do with the DynomoDB.

```
6  /**
7   * Provide an event that contains the following keys:
8   *
9   *   - operation: one of the operations in the switch statement below
10  *   - tableName: required for operations that interact with DynamoDB
11  *   - payload: a parameter to pass to the operation being performed
12  */
13 exports.handler = function(event, context) {
14     //console.log('Received event:', JSON.stringify(event, null, 2));
15
16     var operation = event.operation;
17
```

GET https://oumi48qov7.execute-api.us-east-1.amazonaws.com/prod/cars-list

Authorization | Headers (1) | Body | Pre-request script | Tests

x-api-key Ti7aKnwQYw5admyhxJMEFaM

Header Value

Body | Cookies | Headers (8) | Tests (0/0) | Status 200 OK | Time 1026 ms

Pretty | Raw | Preview | JSON

```
1 {
2   "errorMessage": "Unrecognized operation \"undefined\"",
3   "errorType": "Error",
4   "stackTrace": [
5     "exports.handler (/var/task/index.js:45:26)"
6   ]
7 }
```

Adding Mapping

In API Gateway console, select Resources, /cars-list, GET, Integration Request and add a Mapping Template.

No parameters required, so no mapping is performed, static template:

Mapping Templates

Content-Type	
application/json	

Add mapping template

application/json Mapping template

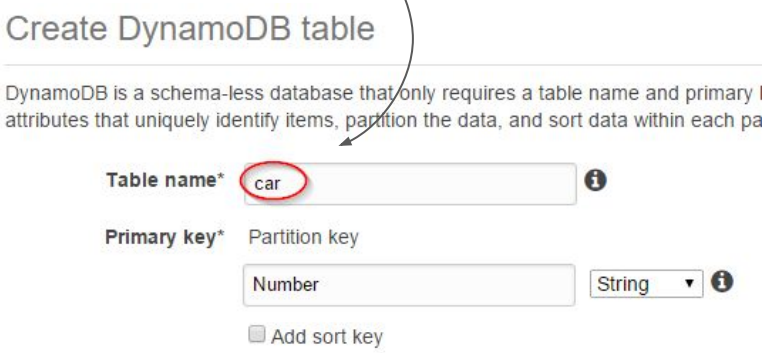
Template

```
1 {
2     "operation" : "list",
3     "tableName" : "car",
4     "payload"   : {}
5 }
```

Now the Lambda function works, but the error indicates that the requested table does not exist.

```
{
  "errorMessage": "Requested resource not found",
  "errorType": "ResourceNotFoundException",
  "stackTrace": [
```

In the DynamoDB console, create a ‘car’ table

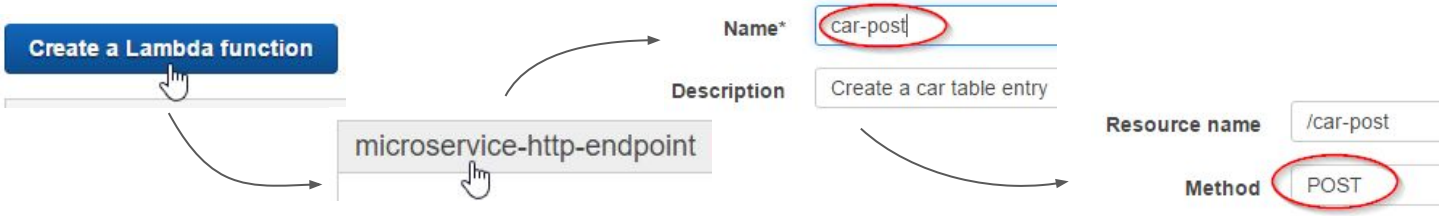


Retest the API endpoint, and now we’re getting JSON data back

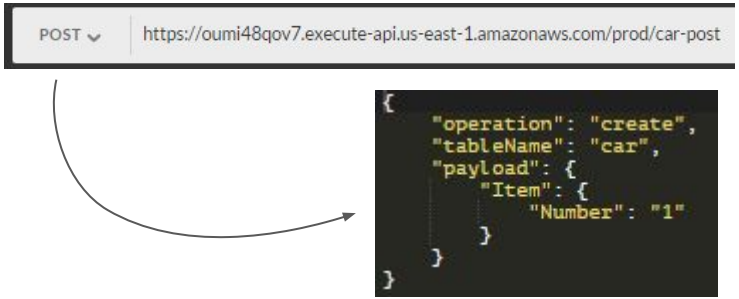

```
{
  "Items": [],
  "Count": 0,
  "ScannedCount": 0
}
```

Do it all again to add a car-post endpoint.

Create the Lambda function, POST this time.

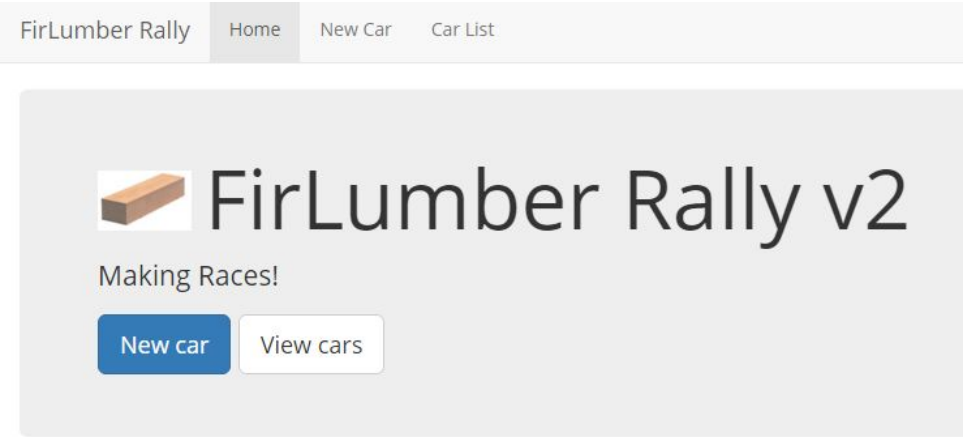


No template required, POST body directly supplies Lambda parameters.



Add a quick AngularJS website

editcar.html	Standard	729 bytes
err.html	Standard	109 bytes
firblock.jpg	Standard	6 KB
flr.html	Standard	2.4 KB
listcars.html	Standard	367 bytes
main.css	Standard	1.8 KB
main.js	Standard	2.2 KB
root.html	Standard	335 bytes
shits-on-fire-yo__big_no.jpeg	Standard	76.5 KB



Controller calls the API gateway, but with the defaults these are different domains (use Route 53).

Use the API Gateway console to add the CORS header:

```
var ENDPOINT = 'https://oumi48qov7.execute-api.us-east-1.amazonaws.com/prod';

.controller('CarListController', ['$scope', '$location', '$http', function($scope, $location, $http) {
    $scope.carREST('get', ENDPOINT+'/cars-list')
        .then(function(response) {
            $scope.cars = response.data.Items;
        });
}])
```

Try it out:

FirLumber Rally

FirLumber Rally v2

Making Races!

New car

View cars

FirLumber Rally

New Car

Car Number

3

Owner

Dave

Submit

How about that, it worked!

FirLumber Rally

All Cars

Number	Owner
1	Sam
2	Bob
3	Dave

http://firlumberrally.s3-website-us-west-2.amazonaws.com/#/

https://github.com/davidknaack/FirLumberRallyII