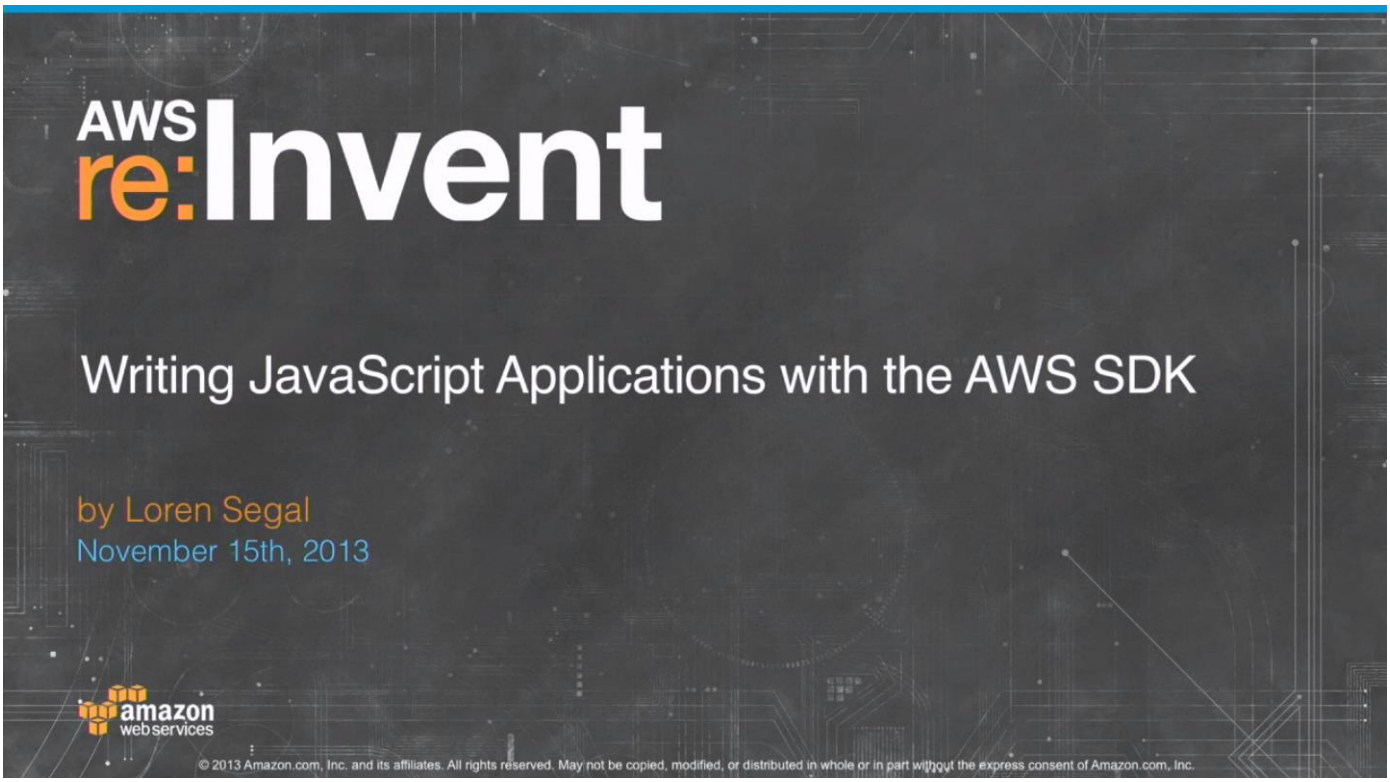
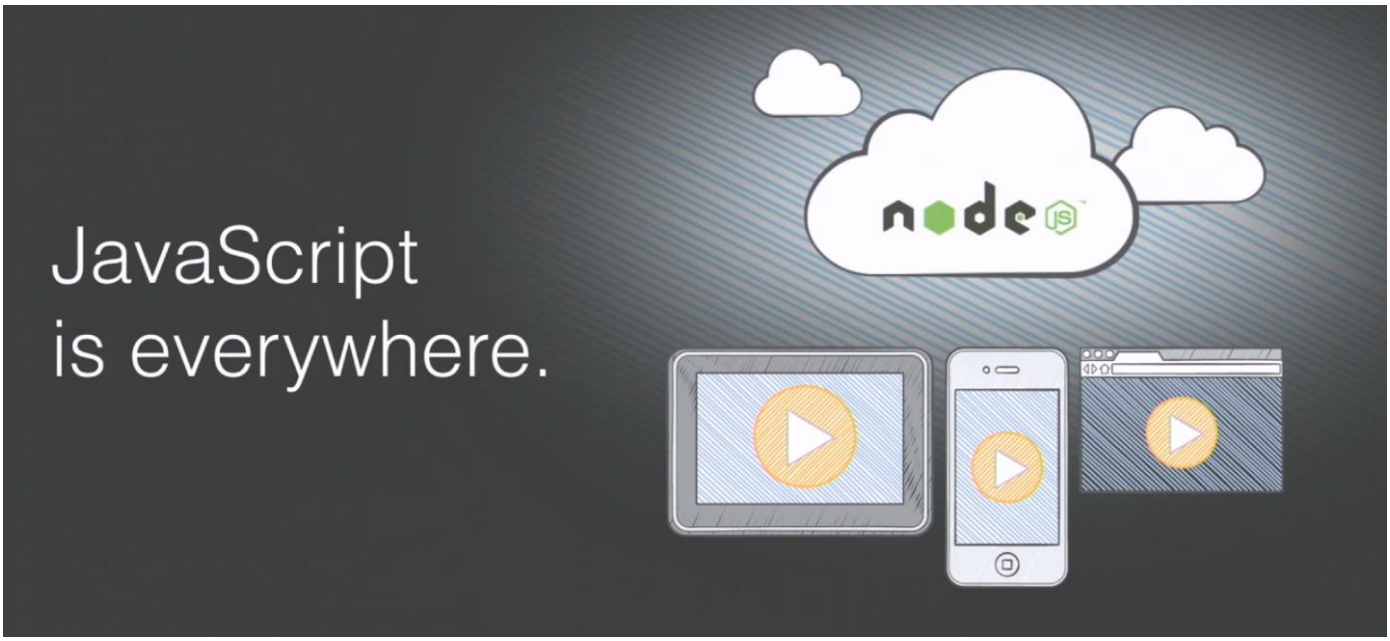
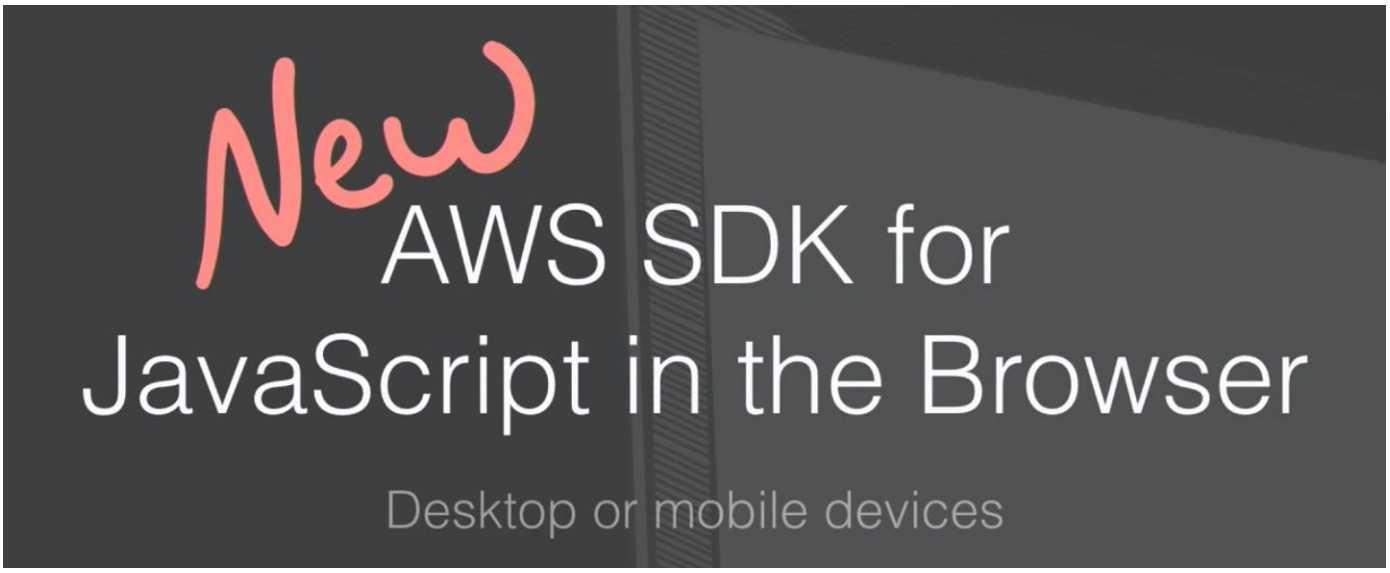


We give a guided tour of using the AWS SDK for JavaScript to create powerful web applications. Learn the best practices for configuration, credential management, streaming requests, as well as how to use some of the higher level features in the SDK. We also show a live demonstration of using AWS services with the SDK to build a real-world JavaScript application.



AWS SDK for JavaScript in Node.js

Almost a year old!



Goals

1. Learn about AWS SDK for Node.js
2. Introduce AWS SDK for JavaScript in the Browser
3. Write a two-tiered web application using JavaScript, HTML, and CSS

AWS SDK for Node.js

Full Service Coverage

Support for over 30 AWS services

Extensible Clients

Customize any part of the request cycle

Standard Node.js Idioms

Streams, EventEmitter, Domains

Open Source

Apache License, Version 2.0

http://github.com/aws/aws-sdk-js

Getting Started

AWS SDK for Node.js

Installing

Bash

```
$ npm install aws-sdk
```

Loading

JS

```
var AWS = require('aws-sdk');
```

Configuring the SDK

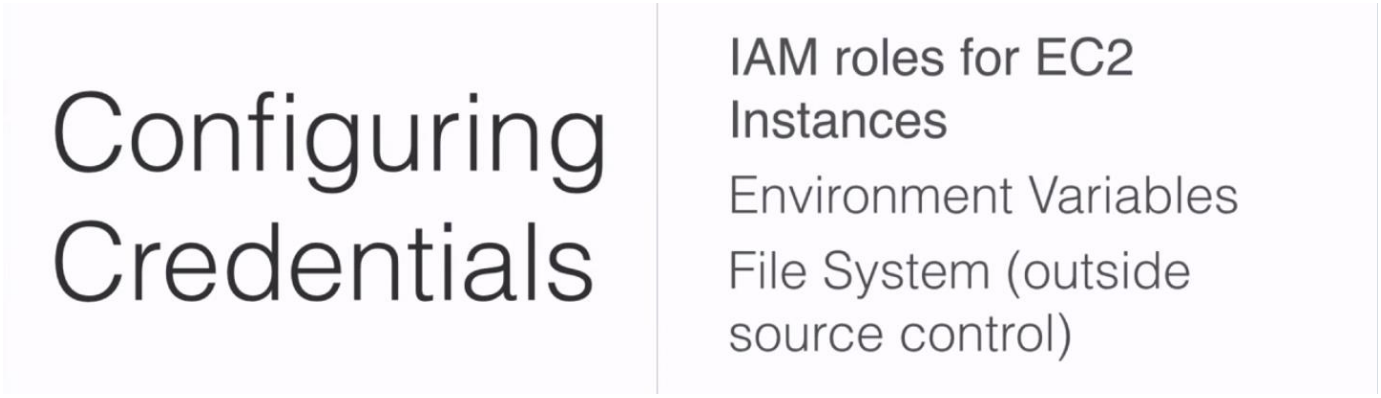
AWS.config

Credentials *

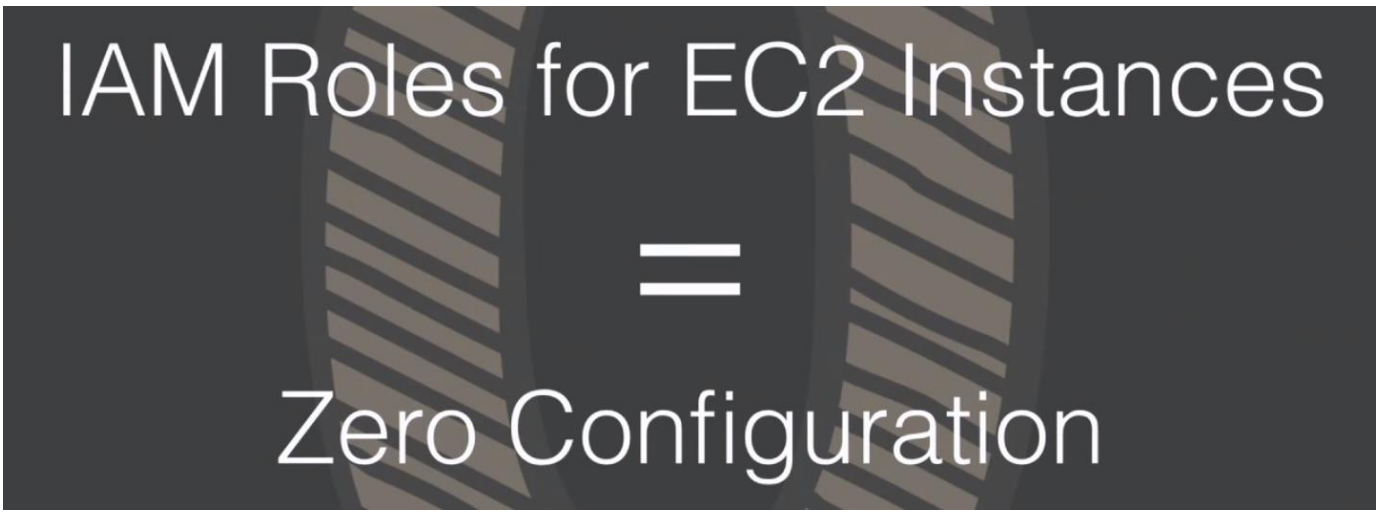
Region *

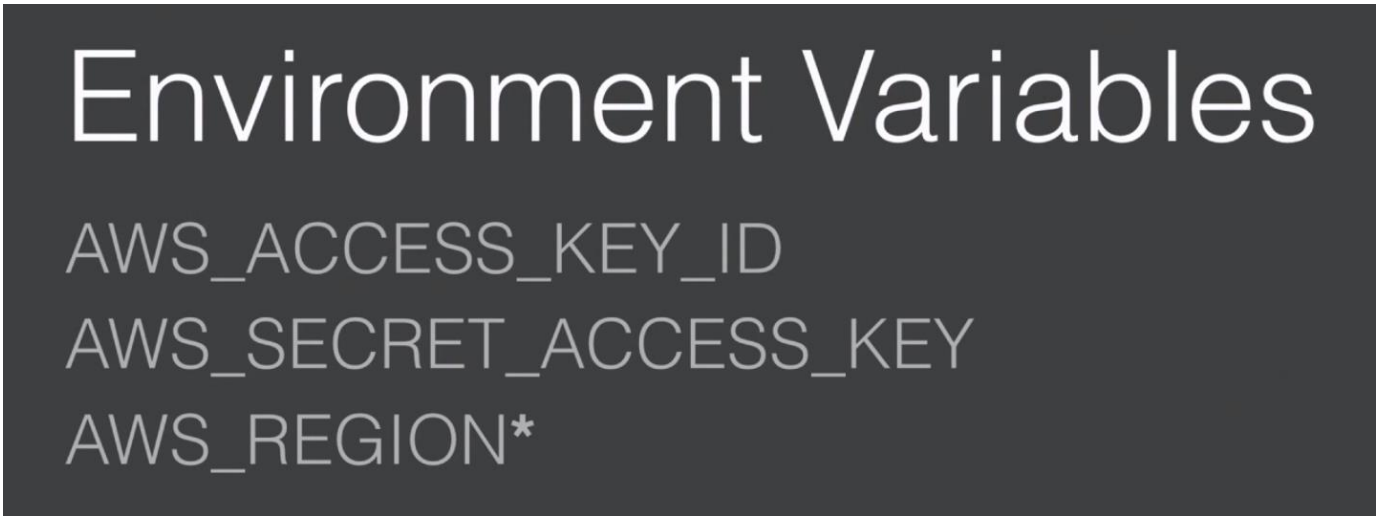
Extras

* Required by the SDK



You can use one of the above options





If you are going to be using environment variables, these are some of the environment variables that you can use to set up the SDK.

JS

```
AWS.config.update({
  region: 'us-west-2', // AWS_REGION
  maxRetries: 10,      // default: 3
  logger: process.stdout,
  // ... more options ...
});
```

This is how you can configure the SDK programmatically,

Config From a File

```JS
AWS.config.loadFromPath('./config.json');
```

If this file contains credentials, keep it out of source control

Working with Services

Service Objects

AWS.S3
AWS.EC2
AWS.DynamoDB
AWS.SQS
AWS.SNS
...

Constructing a Service Object

```JS
var ec2 = new AWS.EC2([config]);
```

You can also pass in an optional configuration as above to have that service be confined to that particular configuration object.

Calling an Operation

```JS
ec2.describeInstances(params, callback);
```

You can call an operation once you have the service object you want to use as above

The Callback

```JS
function (err, data) { ... }
```

The Callback is a standard NodeJS function with the error and data that needs to be sent back if call is successful.

Getting a Request Object

```JS
var req = ec2.describeInstances(params);
```

Sending the Request Object

```JS
var resp = req.send(callback);
```

Adding Listeners to the Request Object

```JS
req.on('complete', function(resp) { ... });
```

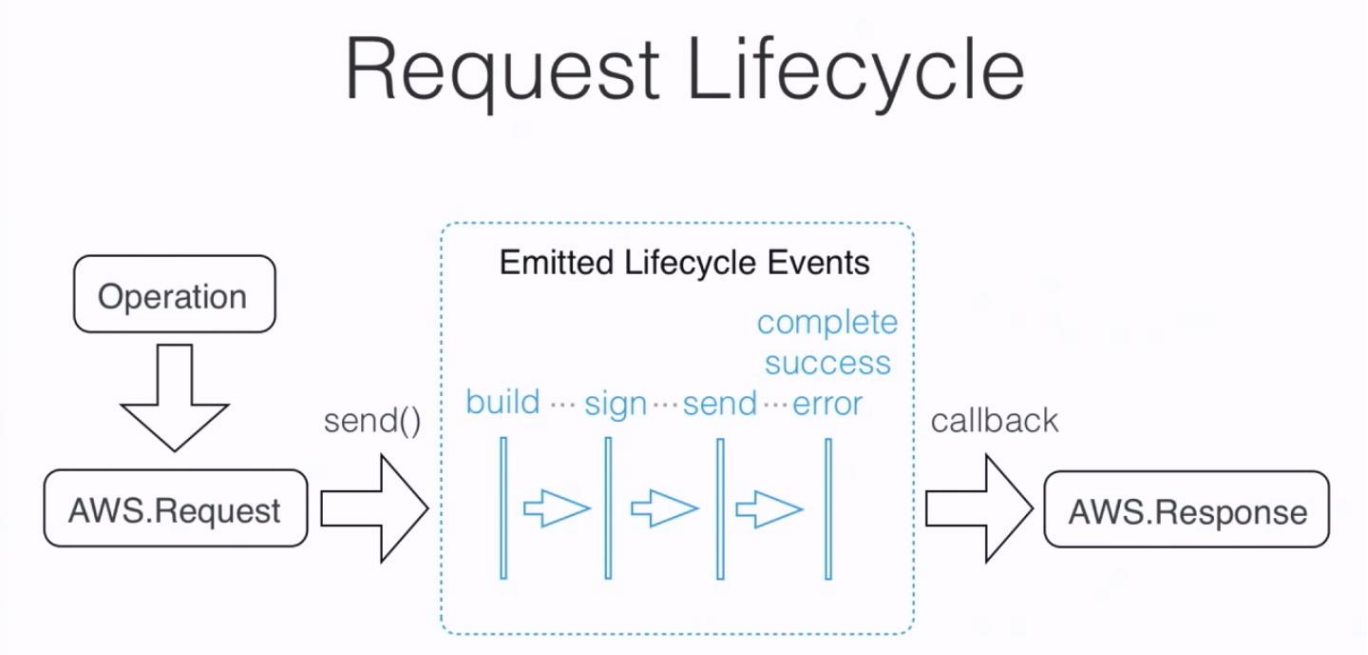
This can be used in an eventing system as shown later

The Request Cycle

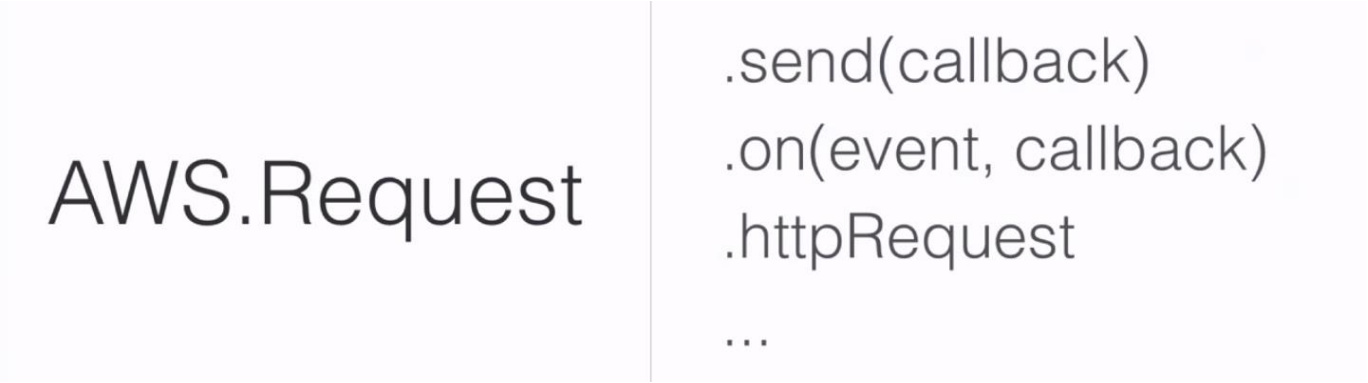
The SDK architecture is mainly made of up requests and responses

Send *AWS.Request*

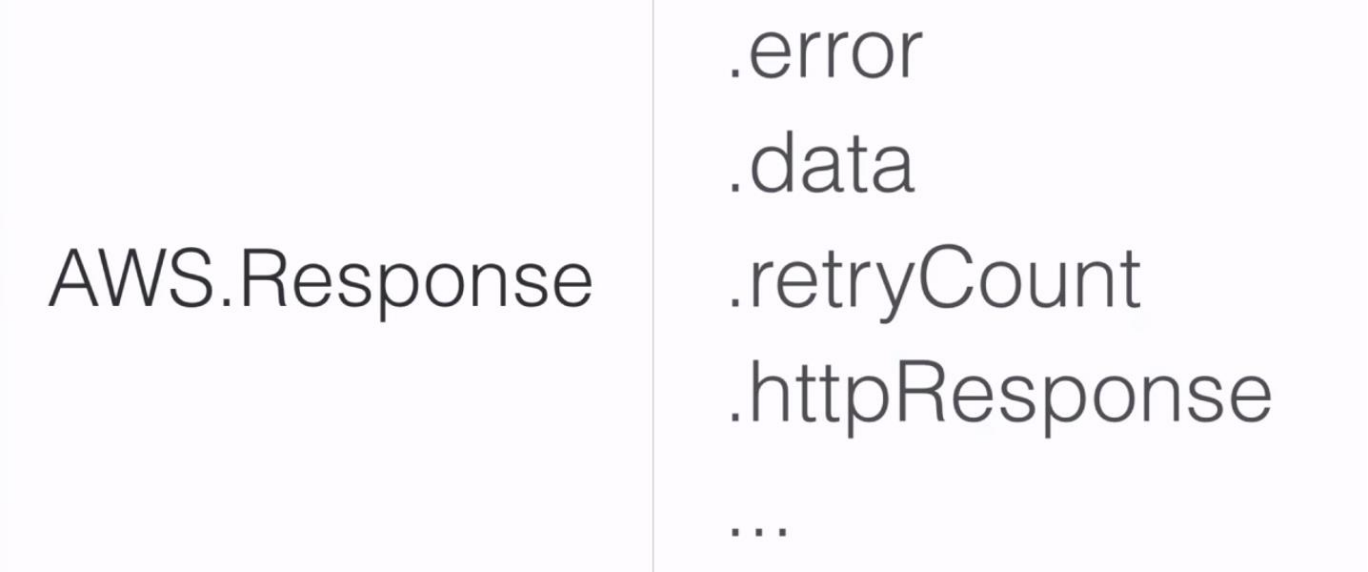
Get *AWS.Response*

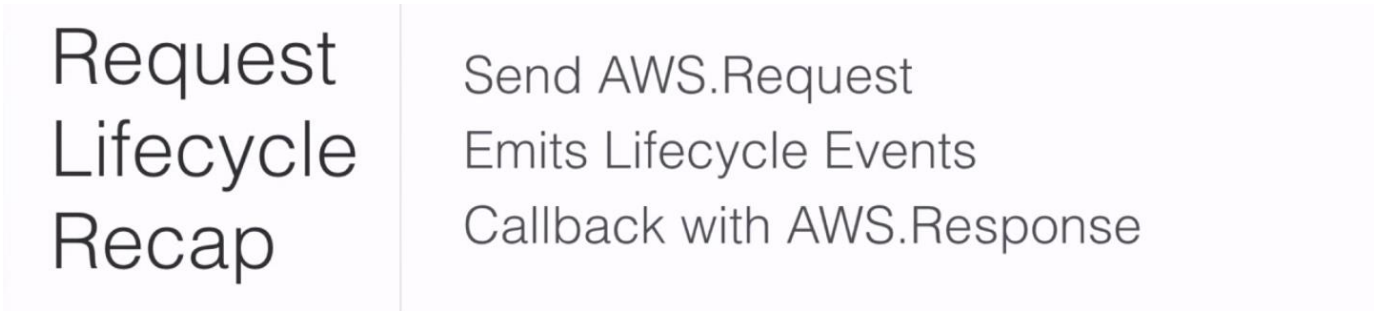


You call and operation that returns to you an AWS.Request object, you can then send the request object to a server. This effectively starts the event lifecycle and events starts getting emitted for the stages of progress. If you passed a callback in the request object, it will get triggered at the end of the request lifecycle and returns an AWS.Response object.

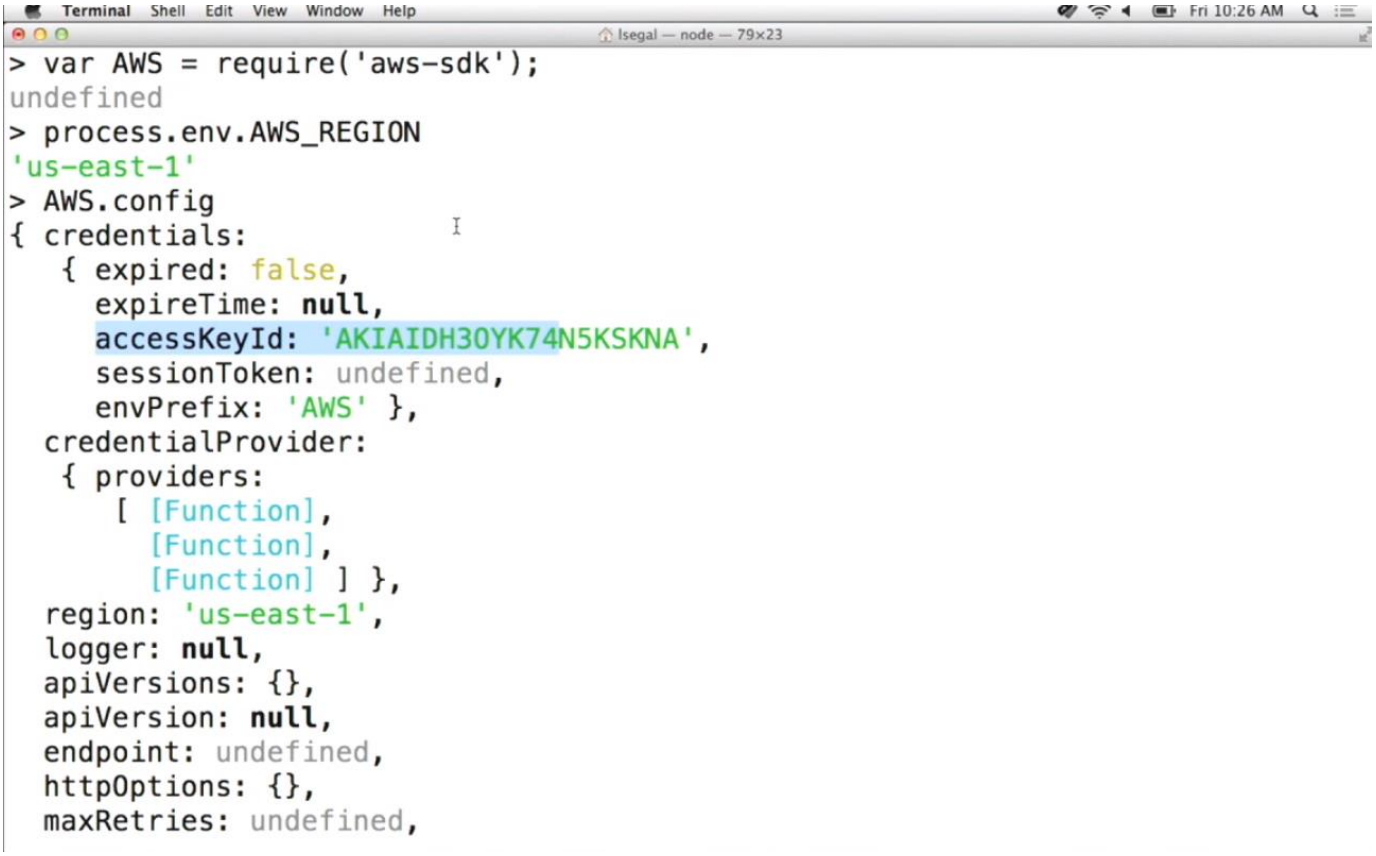


These are some properties on the AWS.Request object, the .httpRequest helps you see the request headers and so on

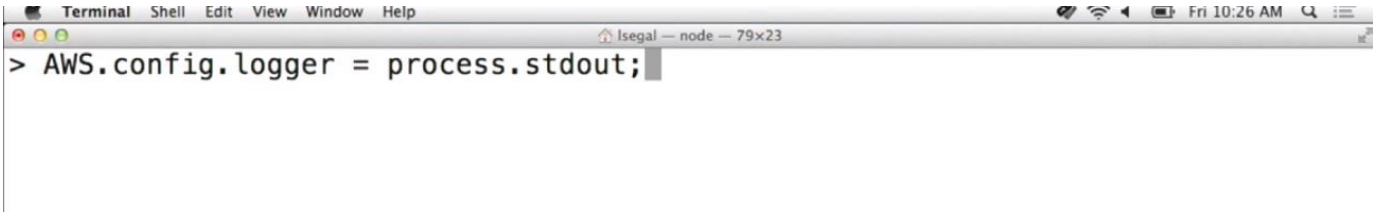




This is how the Request architecture works in the SDK

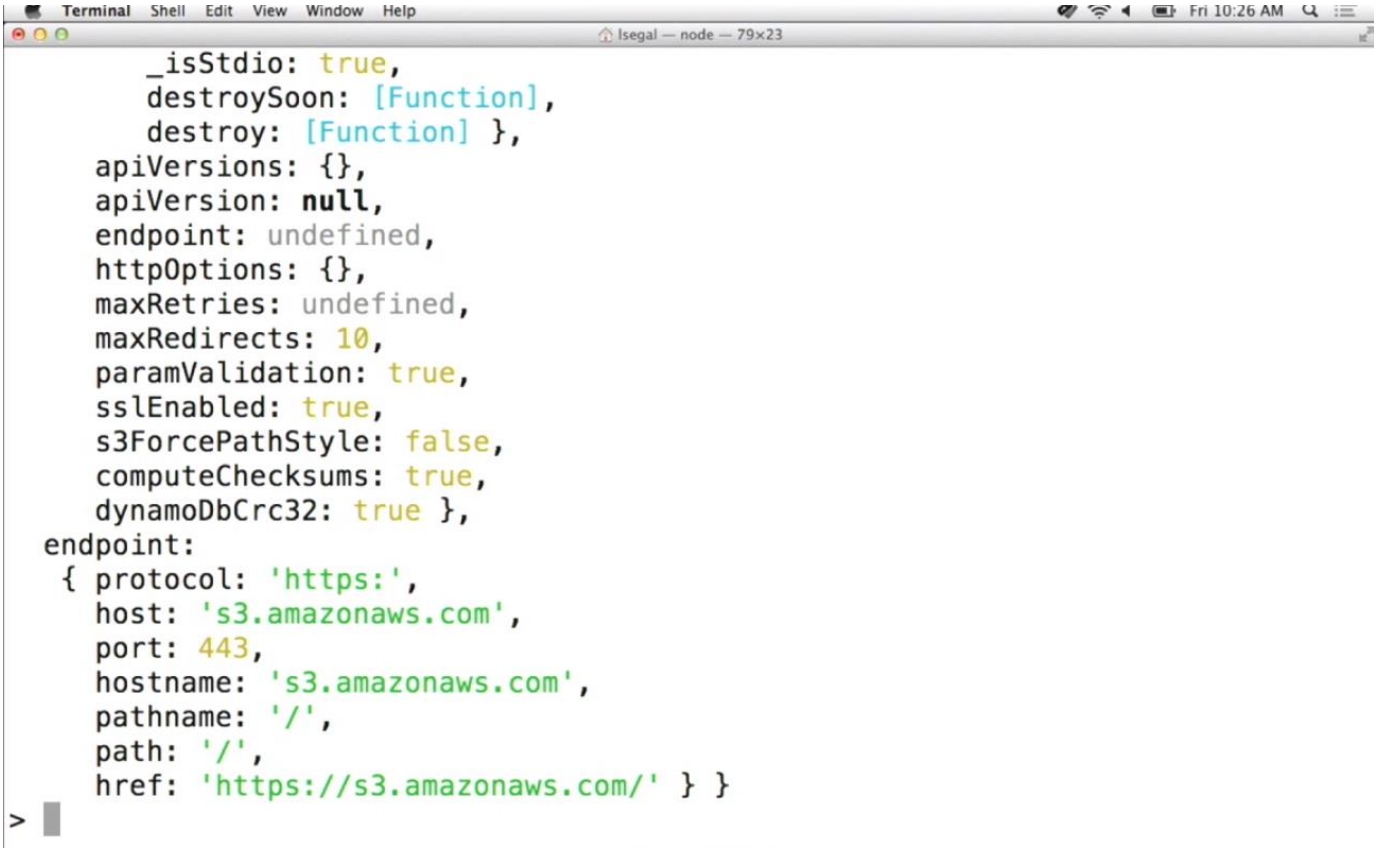


Log in to our terminal and start NodeJS

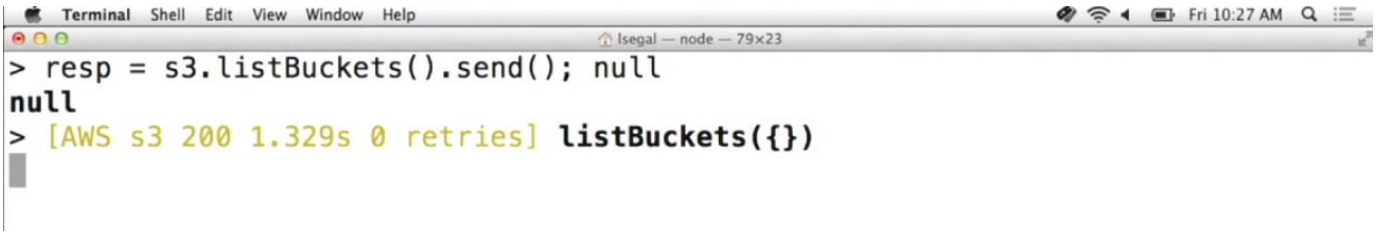


We then set up our ***logger*** and set it to ***stdout*** in this case

Next, we create an S3 object,



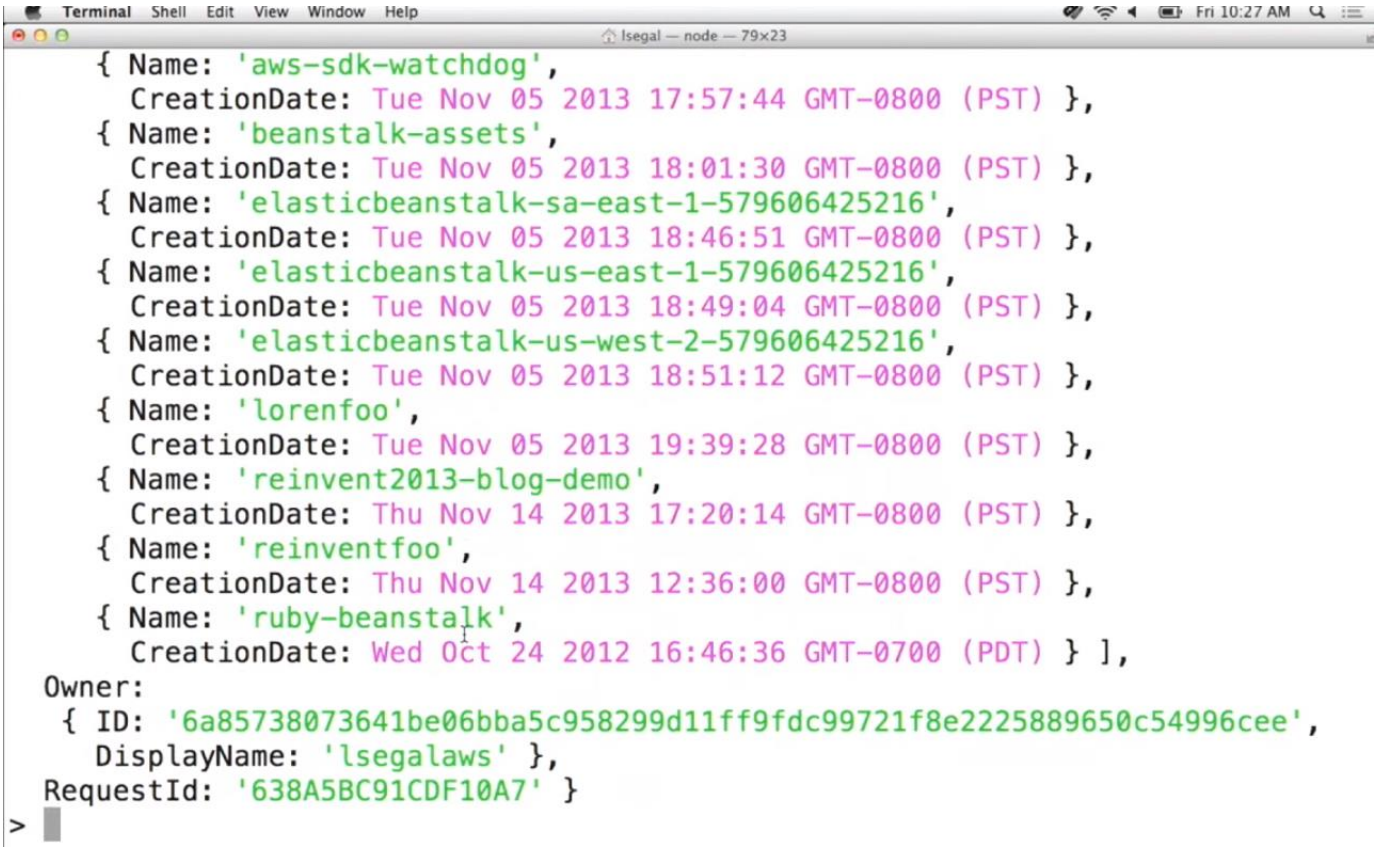
We now have an S3 object

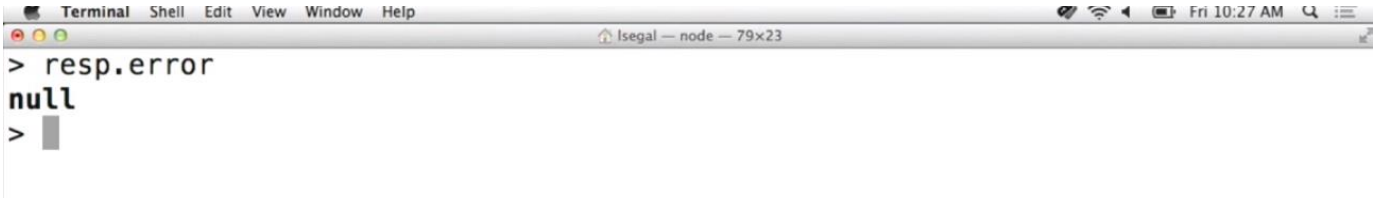


We then send a simple ***listBuckets*** request to the server as above. There is no callback here and we have used a null to limit the response displayed

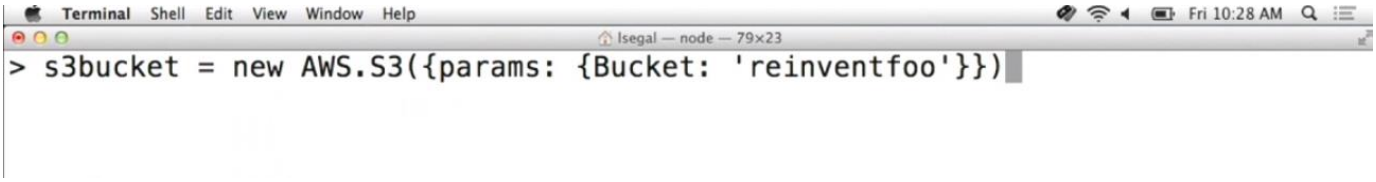
```
> resp = s3.listBuckets().send(); null
null
> [AWS s3 200 1.329s 0 retries] listBuckets({})
resp.data
{ Buckets:
   [ { Name: 'aws-reinvent-2013-test-bucket0',
       CreationDate: Thu Nov 14 2013 11:29:14 GMT-0800 (PST) },
     { Name: 'aws-reinvent-2013-test-bucket1',
       CreationDate: Thu Nov 14 2013 11:29:15 GMT-0800 (PST) },
     { Name: 'aws-reinvent-2013-test-bucket10',
       CreationDate: Thu Nov 14 2013 11:29:29 GMT-0800 (PST) },
```

We now can do a ***resp.data*** to see the data returned from the ***listBuckets*** command

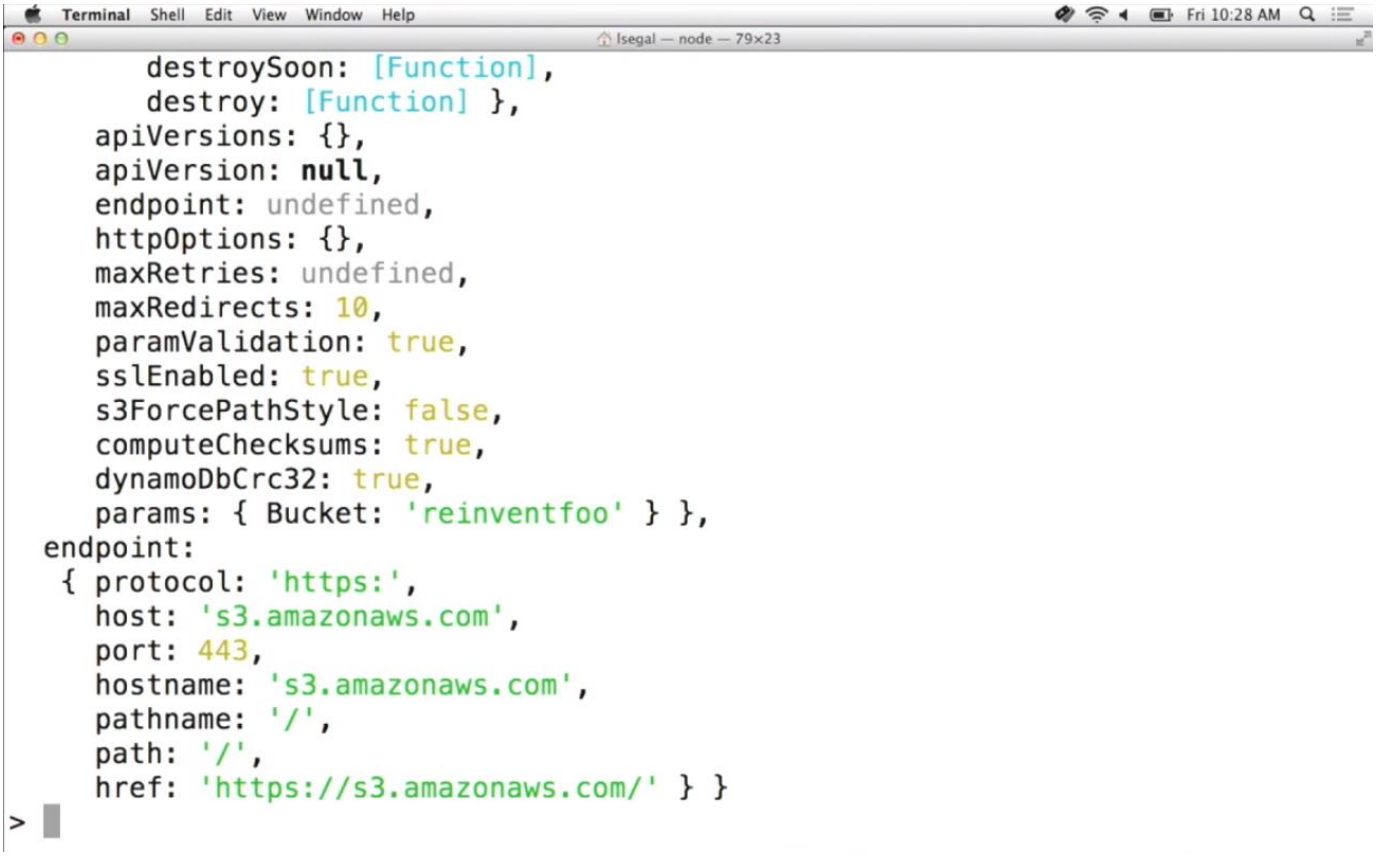




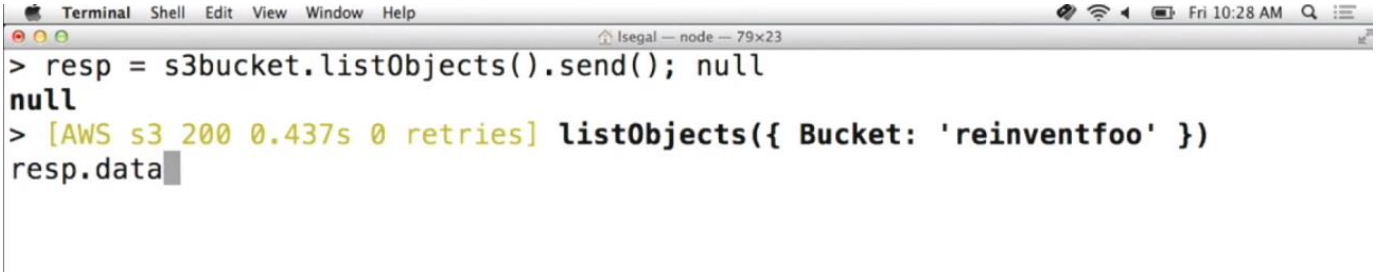
This is how we can use the ***SDK*** at a basic level.



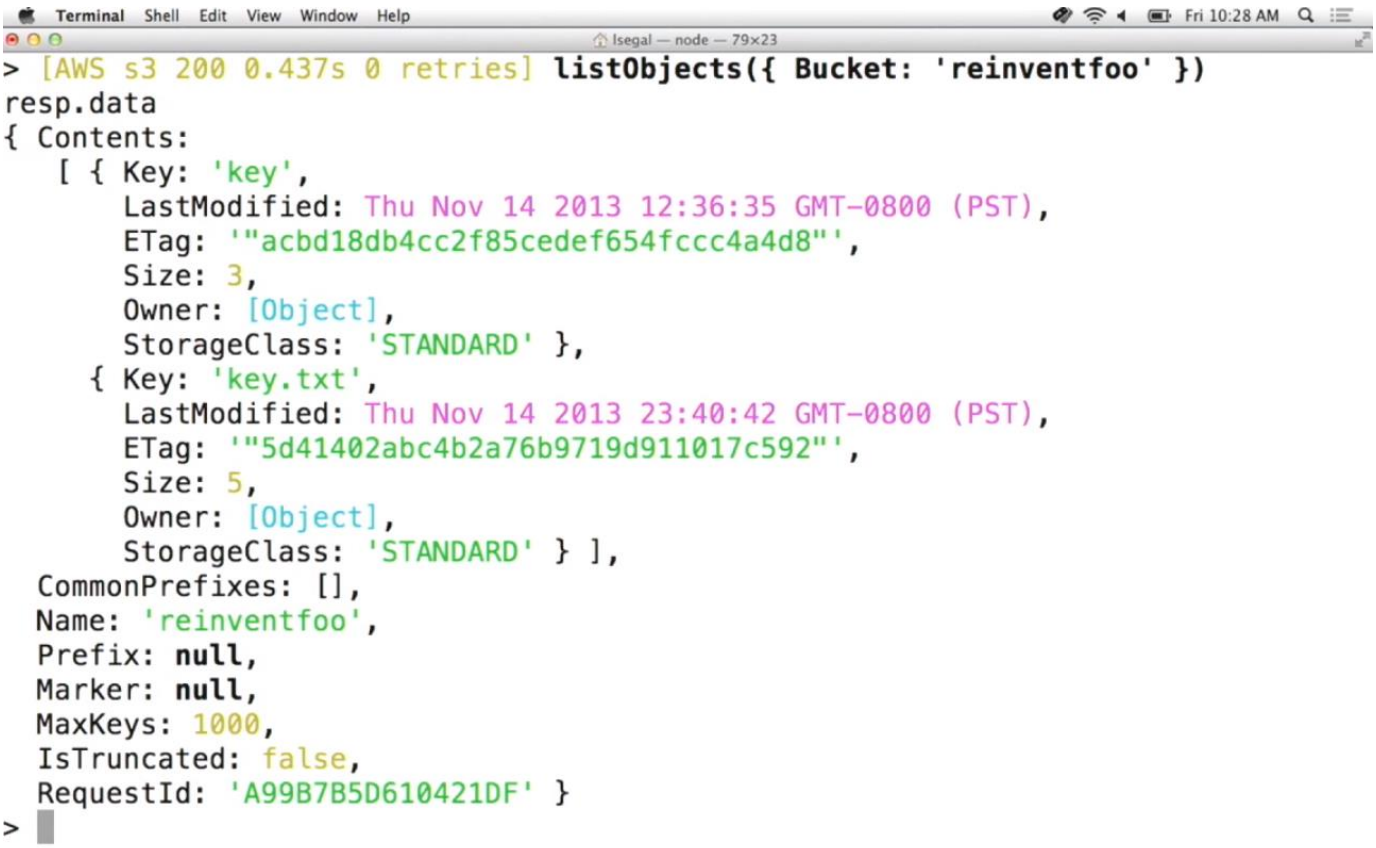
When you are working with services like ***S3***, you are often sending a lot of the same ***parameters*** to the same thing like a bucket. The SDK allows you to use the concept of ***bound parameters*** to send commonly used data once.

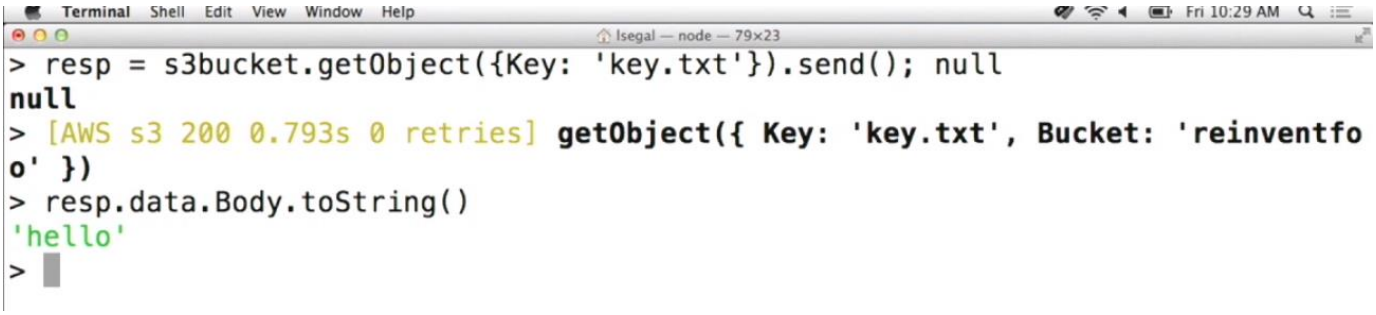


We now have a bucket created with some bounded parameters to it.

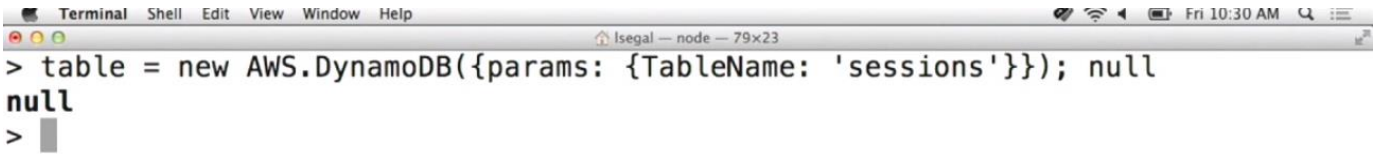


We can now just call the method against the resource we want as above without passing the parameters along anymore

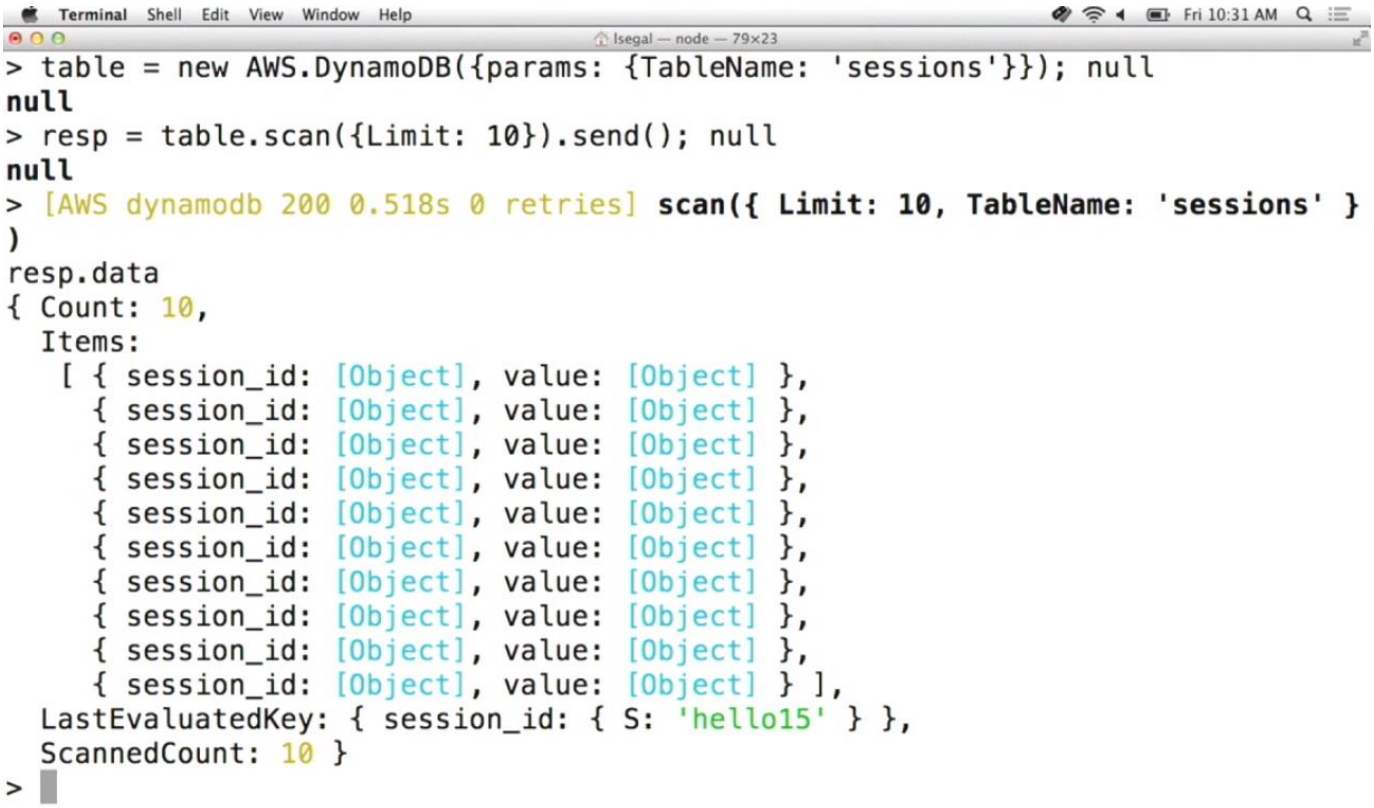




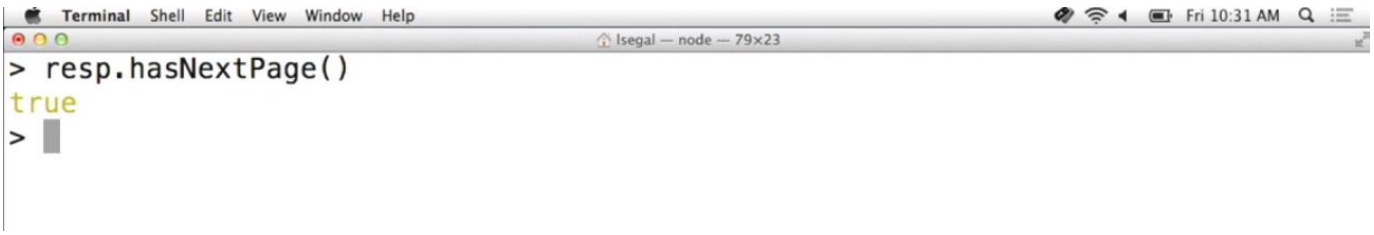
You can bind parameters to any service object available in the SDK



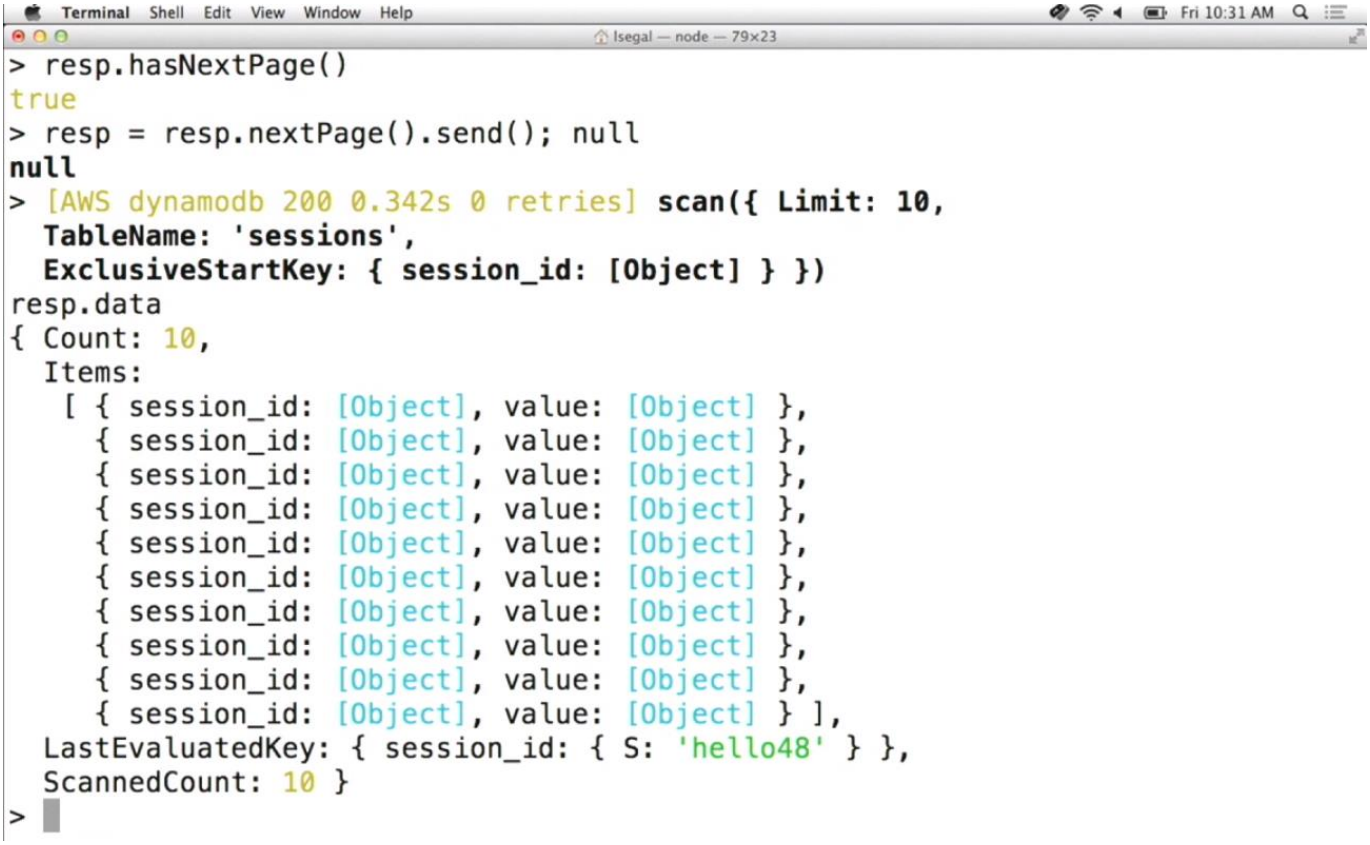
Let us see response pagination feature SDK feature. This is useful when your request to services like S3 of DynamoDB is large and splitted over multiple parts or calls so you have to page repeatedly, the SDK simplifies this for you automatically. We have created a DynamoDB table object above to demonstrate this feature



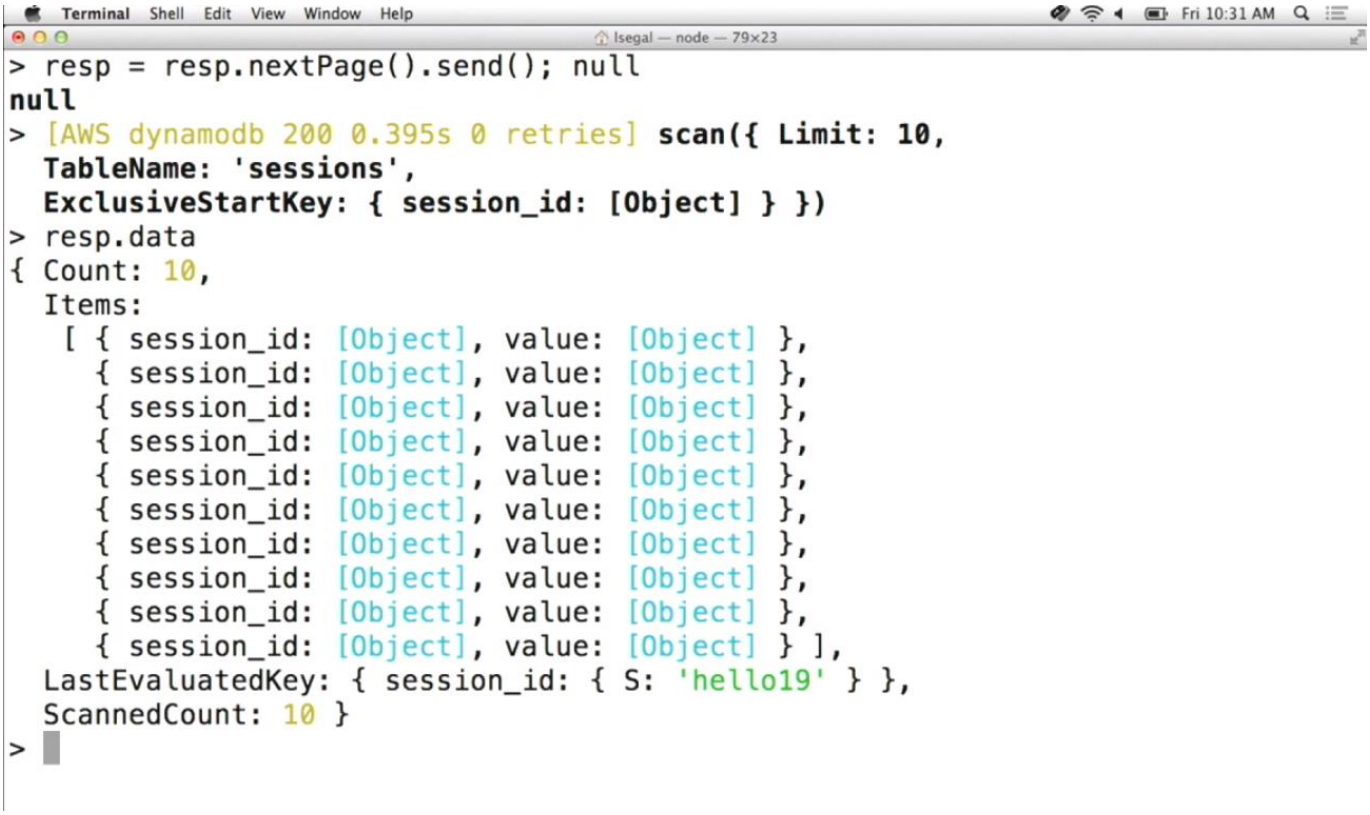
Sometimes there is a lot more data that we want back, so the SDK is going to help us out



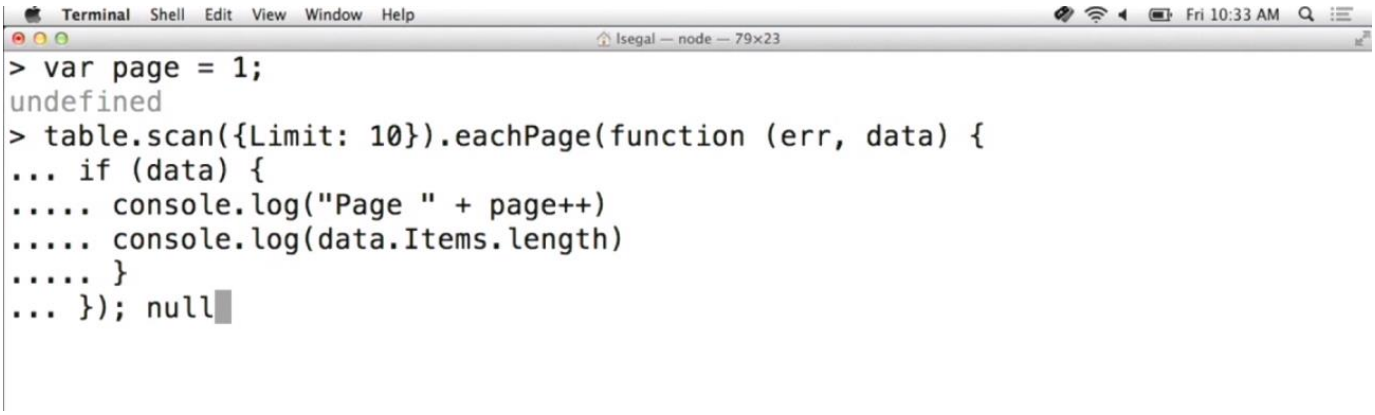
The ***SDK*** has a method on the response object called ***resp.hasNextPage()***, the above case shows that there are more pages to get for the call we made for the response object



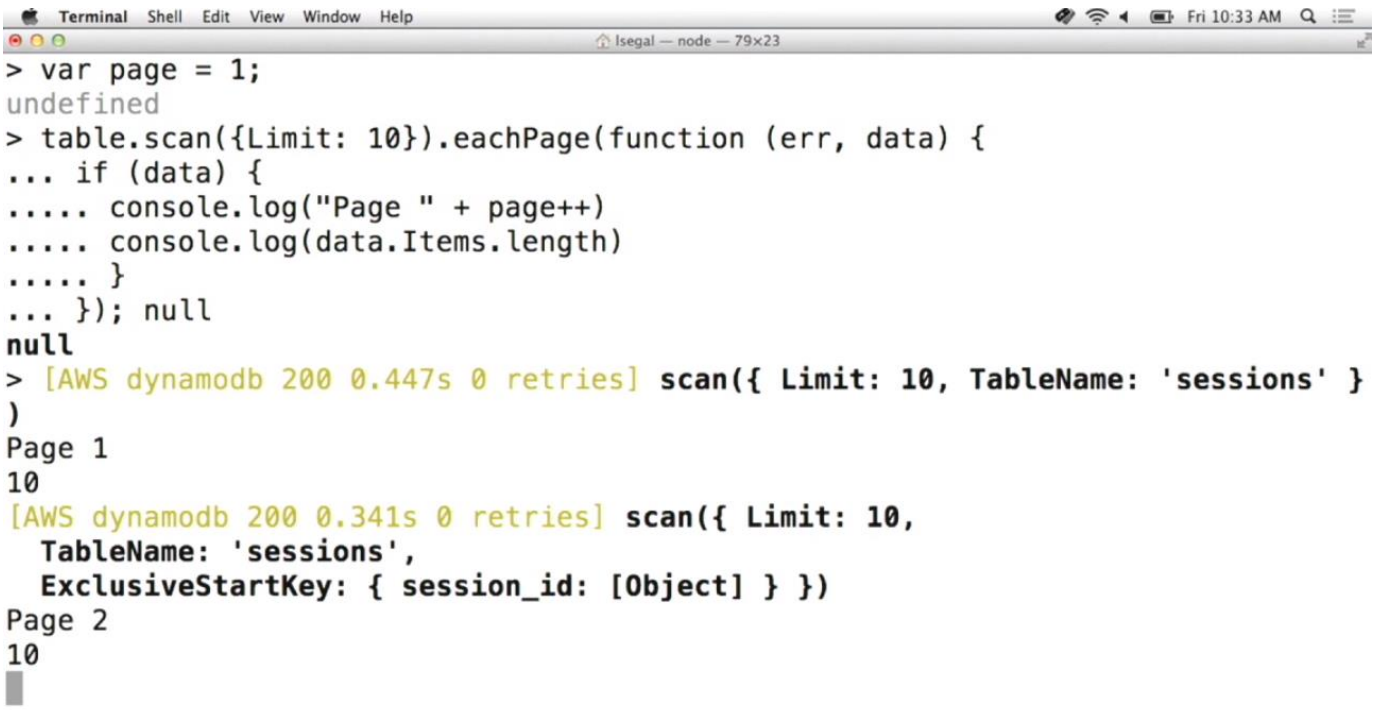
This will submit a new request for the next page of results

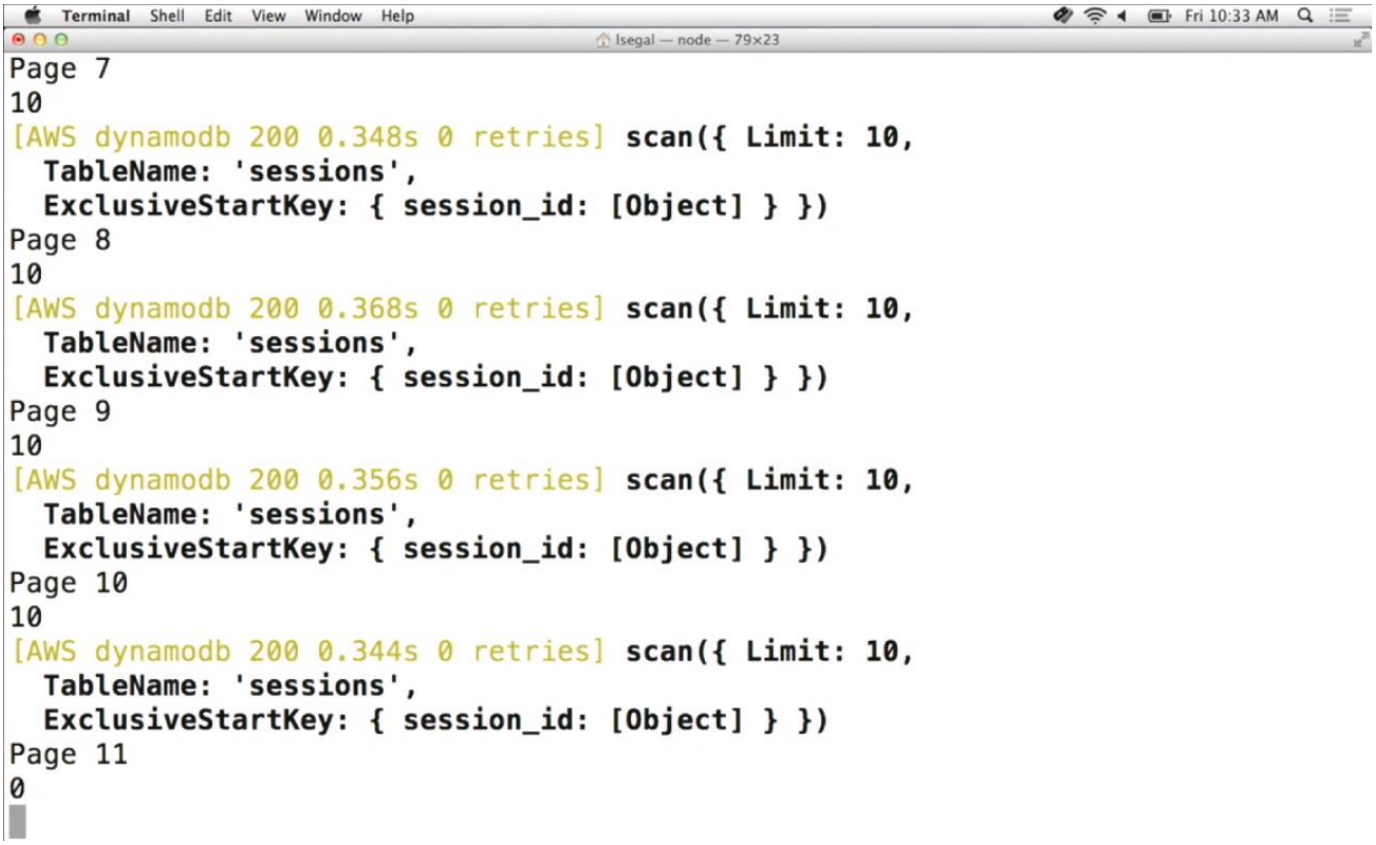


We can continue to use the ***resp.hasNextPage()*** method on our own to get all the response data but the SDK has a way to help us get all the data also as shown below

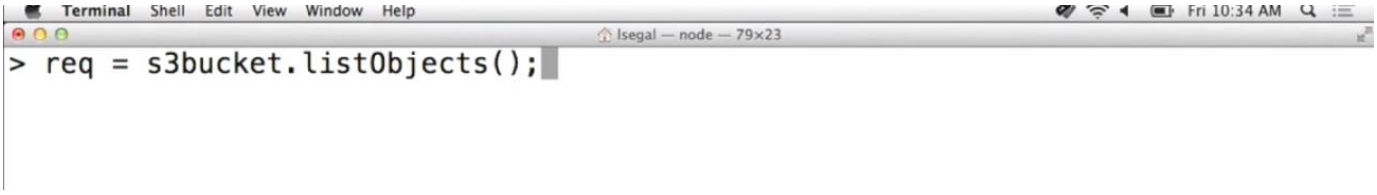


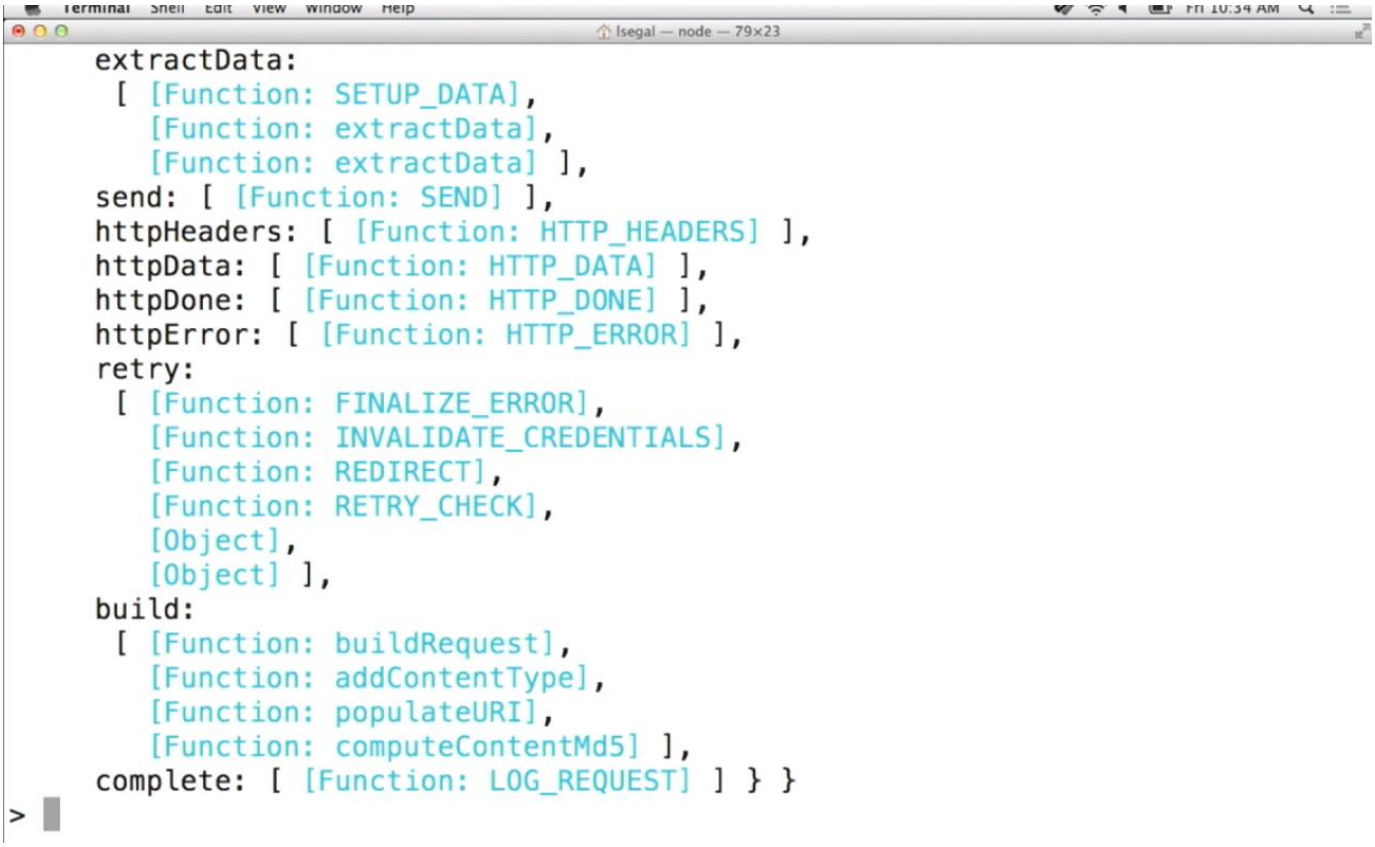
The ***eachPage()*** does the ***hasNextPage()*** method call on our behalf to get all the data in one go.

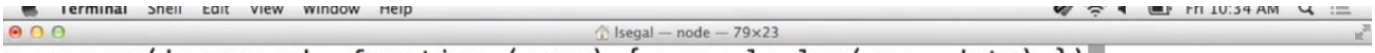


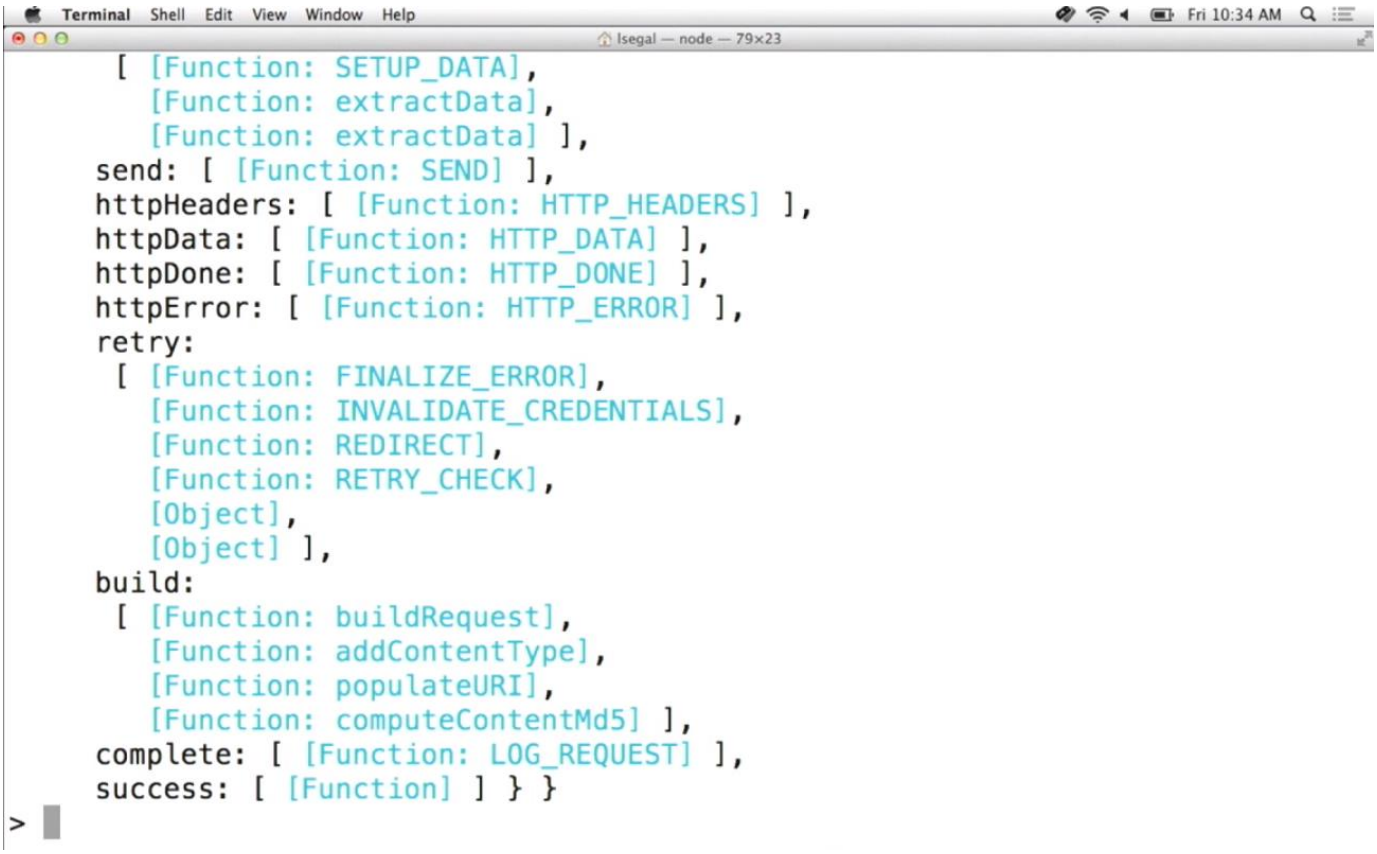


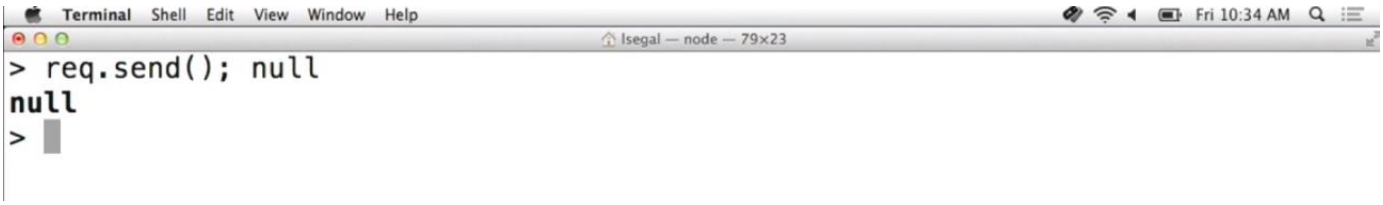
It ***scanned*** and ran through all the available for us automatically.

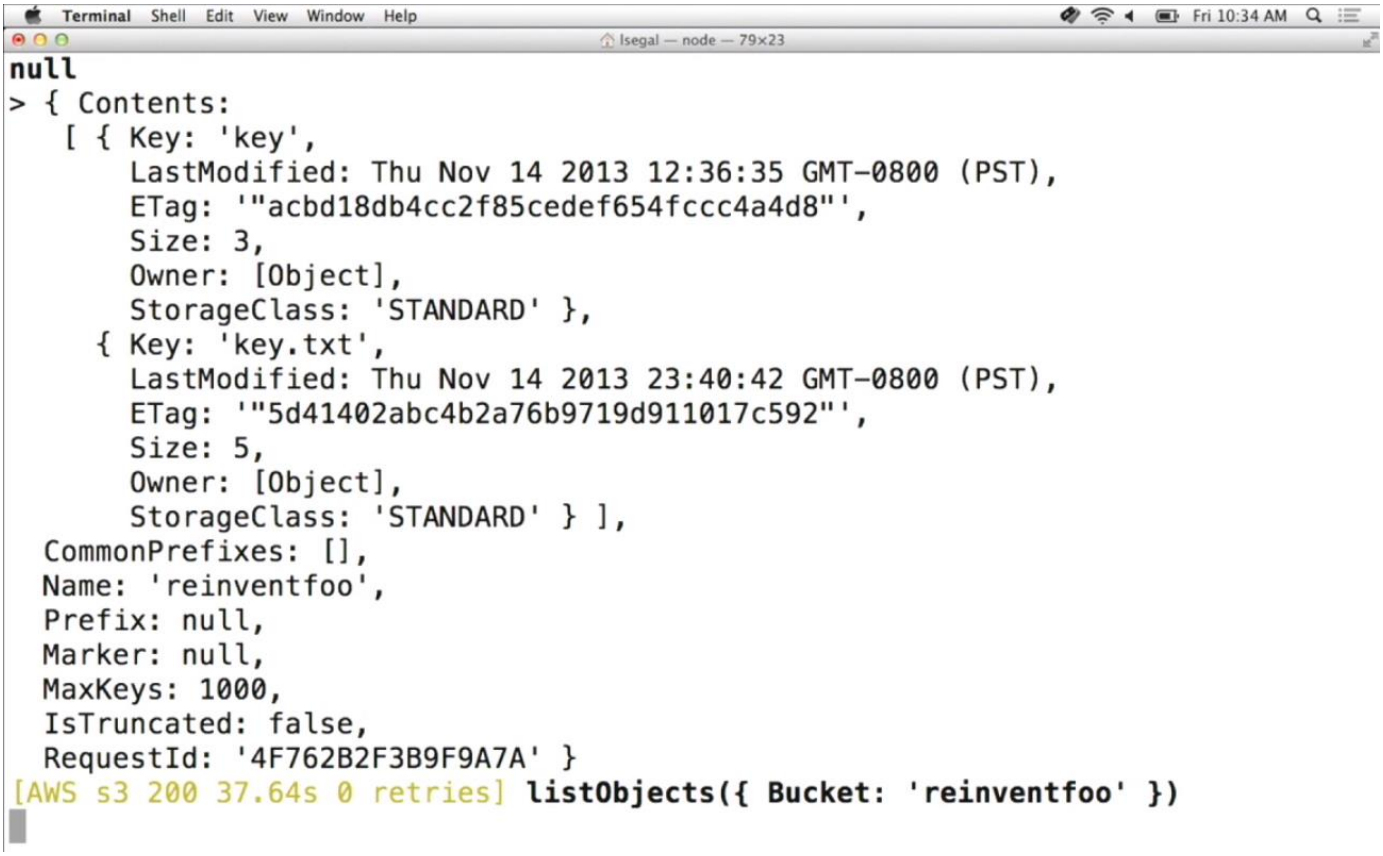



lsegal — node — 79×23
extractData:
[[Function: SETUP_DATA],
[Function: extractData],
[Function: extractData]],
send: [[Function: SEND]],
httpHeaders: [[Function: HTTP_HEADERS]],
httpData: [[Function: HTTP_DATA]],
httpDone: [[Function: HTTP_DONE]],
httpError: [[Function: HTTP_ERROR]],
retry:
[[Function: FINALIZE_ERROR],
[Function: INVALIDATE_CREDENTIALS],
[Function: REDIRECT],
[Function: RETRY_CHECK],
[Object],
[Object]],
build:
[[Function: buildRequest],
[Function: addContentType],
[Function: populateURI],
[Function: computeContentMd5]],
complete: [[Function: LOG_REQUEST]] } }
>


lsegal — node — 79×23
> req.on('success', function (resp) { console.log(resp.data) })

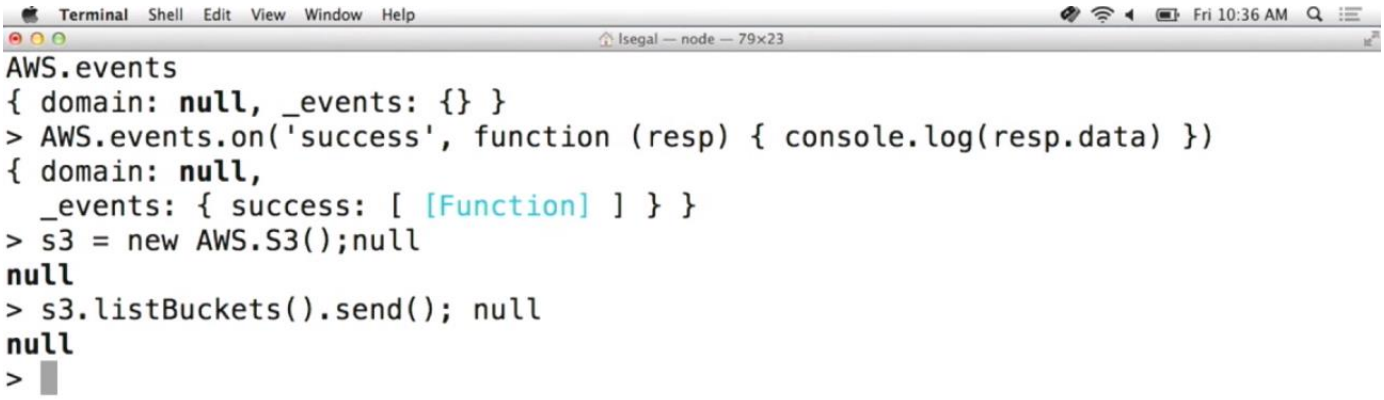

Terminal Shell Edit View Window Help
Fri 10:34 AM
lsegal — node — 79×23
[[Function: SETUP_DATA],
[Function: extractData],
[Function: extractData]],
send: [[Function: SEND]],
httpHeaders: [[Function: HTTP_HEADERS]],
httpData: [[Function: HTTP_DATA]],
httpDone: [[Function: HTTP_DONE]],
httpError: [[Function: HTTP_ERROR]],
retry:
[[Function: FINALIZE_ERROR],
[Function: INVALIDATE_CREDENTIALS],
[Function: REDIRECT],
[Function: RETRY_CHECK],
[Object],
[Object]],
build:
[[Function: buildRequest],
[Function: addContentType],
[Function: populateURI],
[Function: computeContentMd5]],
complete: [[Function: LOG_REQUEST]],
success: [[Function]] } }
>



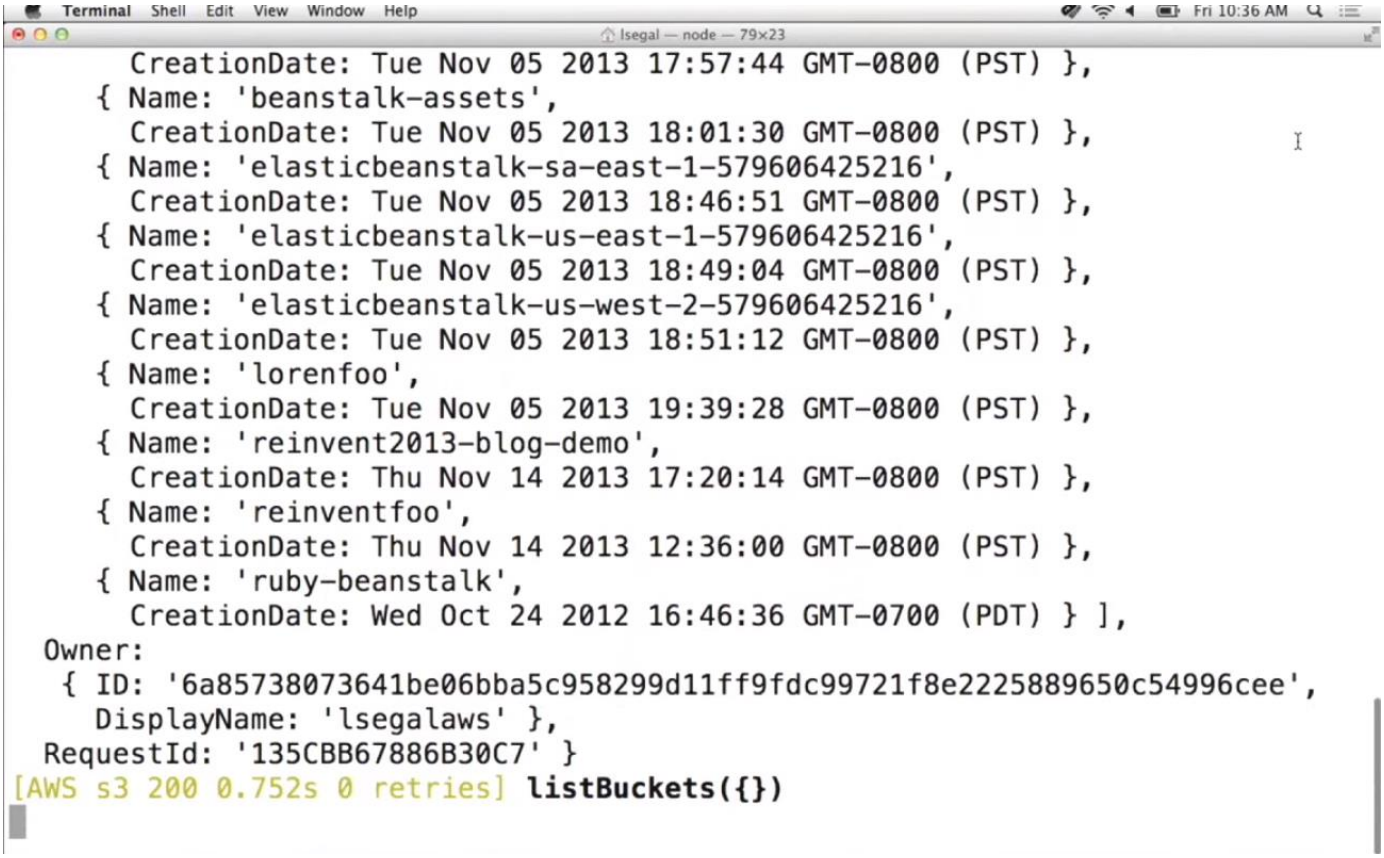


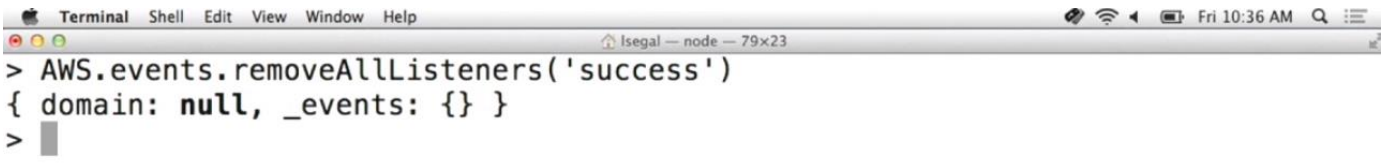
The SDK also has a global events object that is used every time you create a new service object.

We can now attach a success event to the global object as above.

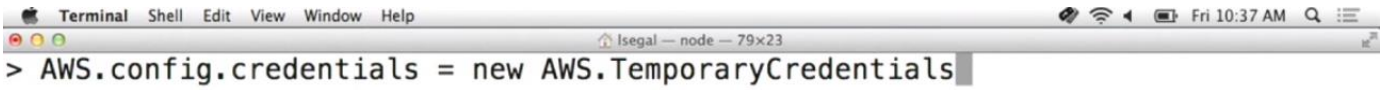


We then recreate the S3 object too

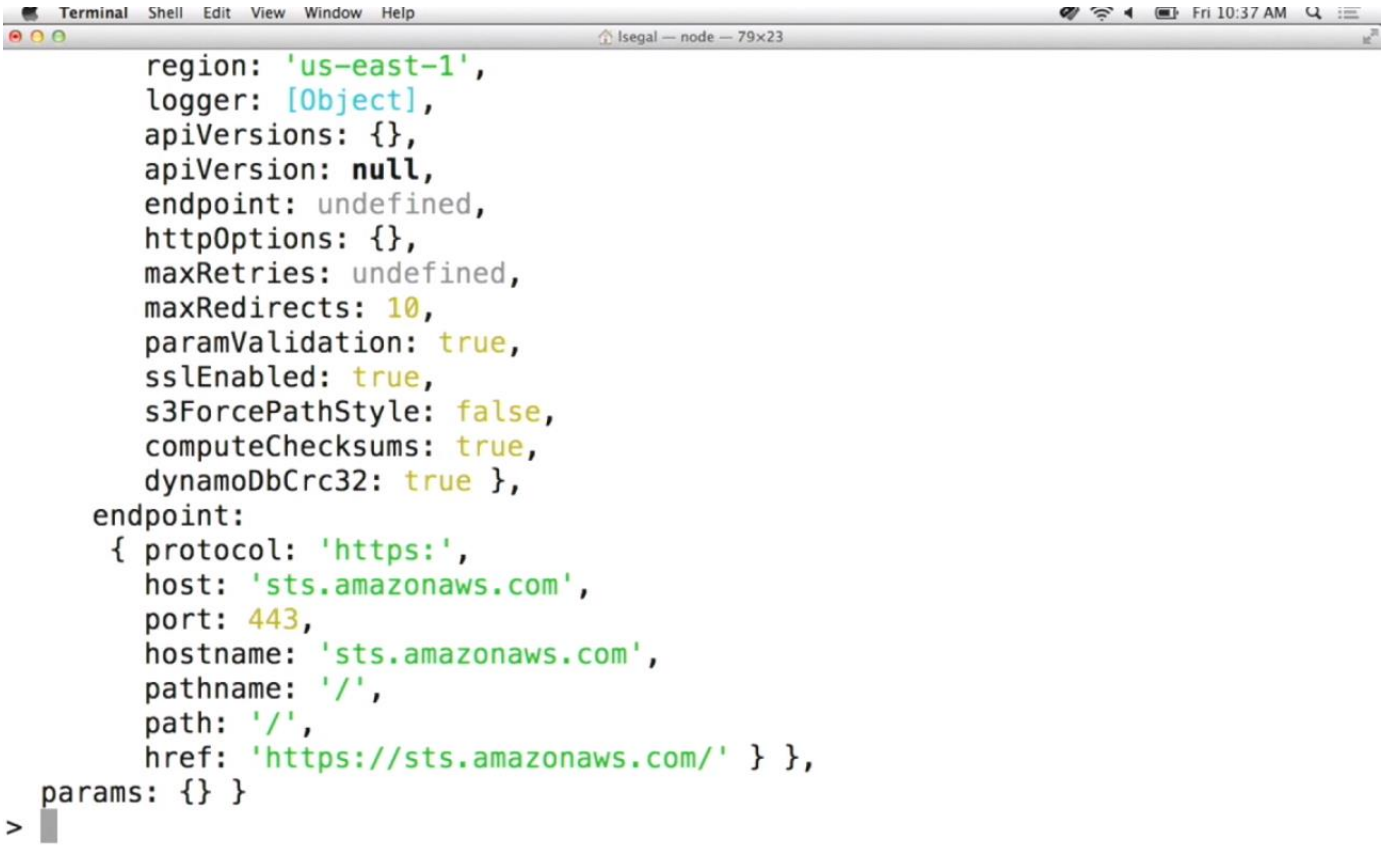


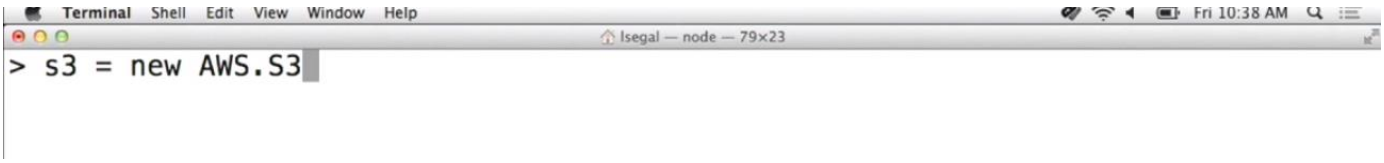


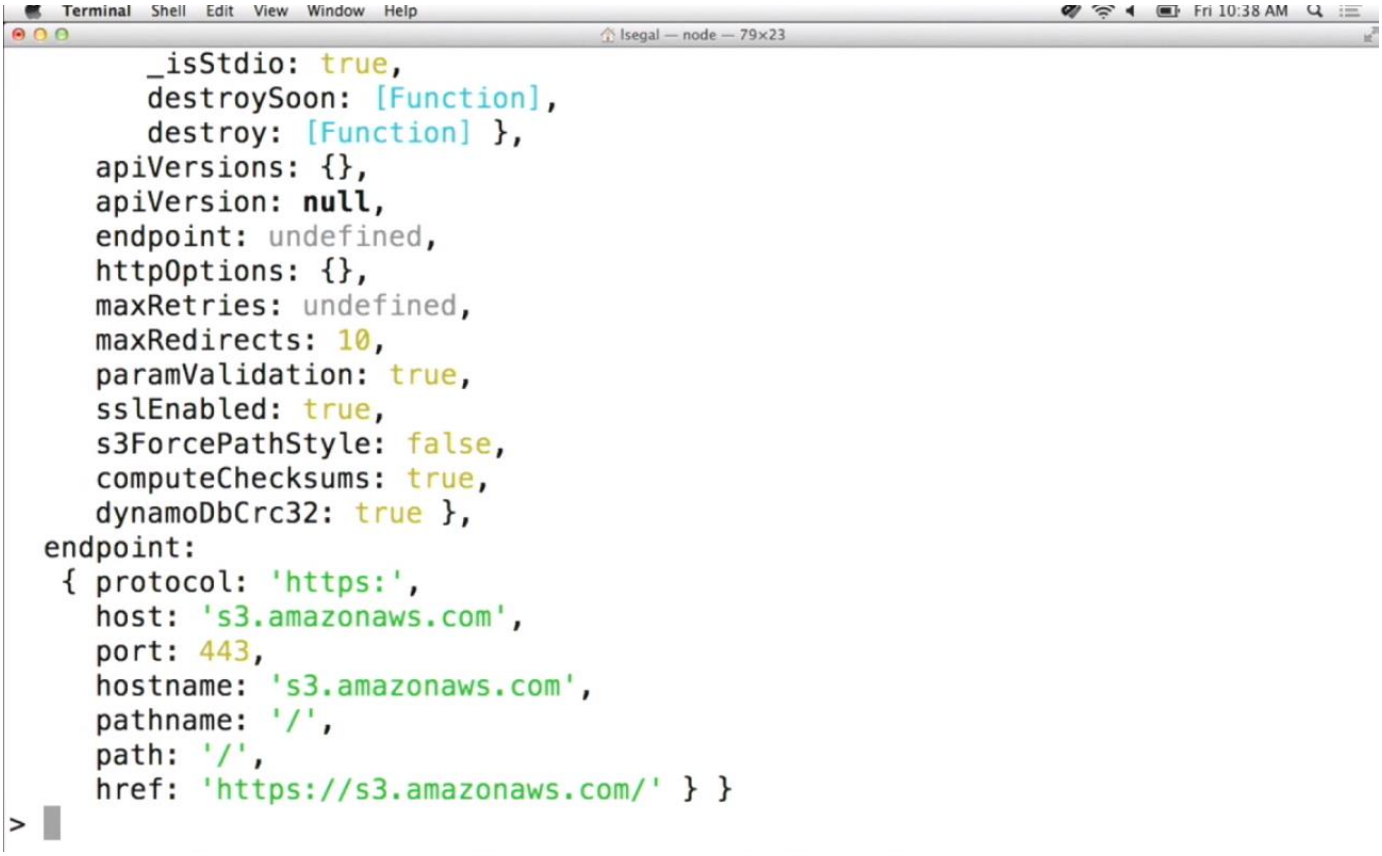
This clears the events object as above.



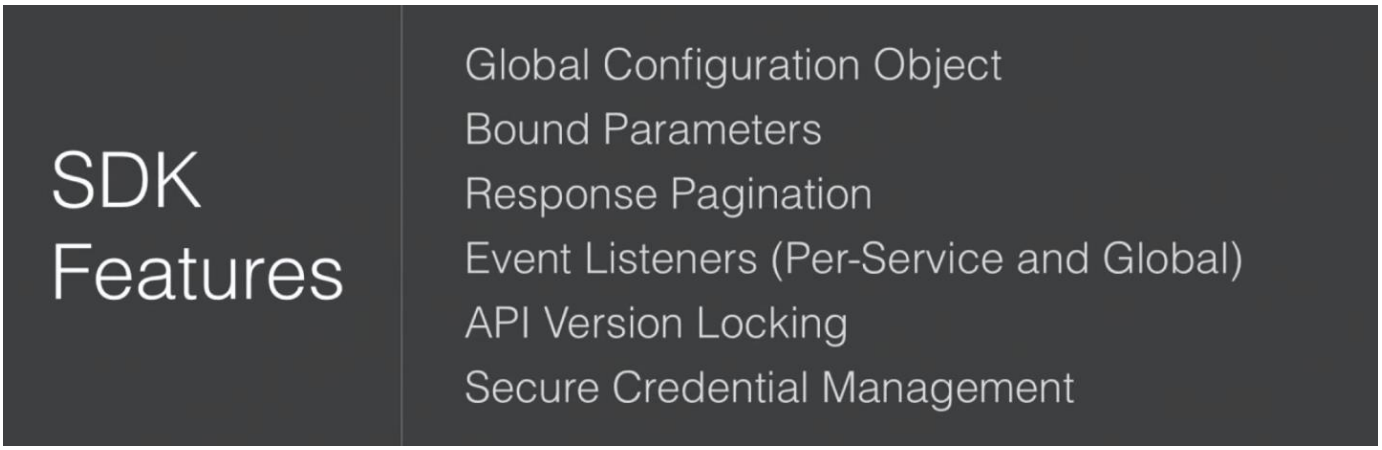
Next, let us see how to manage your credentials properly in your application using some new helpers added to the SDK. The AWS.TemporaryCredentials() method fetches temporary credentials from IAM and STS that will eventually expire after a short time period.







We now have a newly created S3 object that has that temporary credentials set up



AWS SDK for JavaScript in the Browser

Developer Preview

Looking for Feedback

Getting the SDK

```
<script src="https://sdk.amazonaws.com/js/aws-sdk-2.0.0-rc1.min.js" />
```

5 Supported Services

- Amazon S3
- Amazon DynamoDB
- Amazon SQS
- Amazon SNS
- STS

All Modern Browsers

28.0+ 23.0+ 10+ 17.0+ 5.1+

Usage is the same.

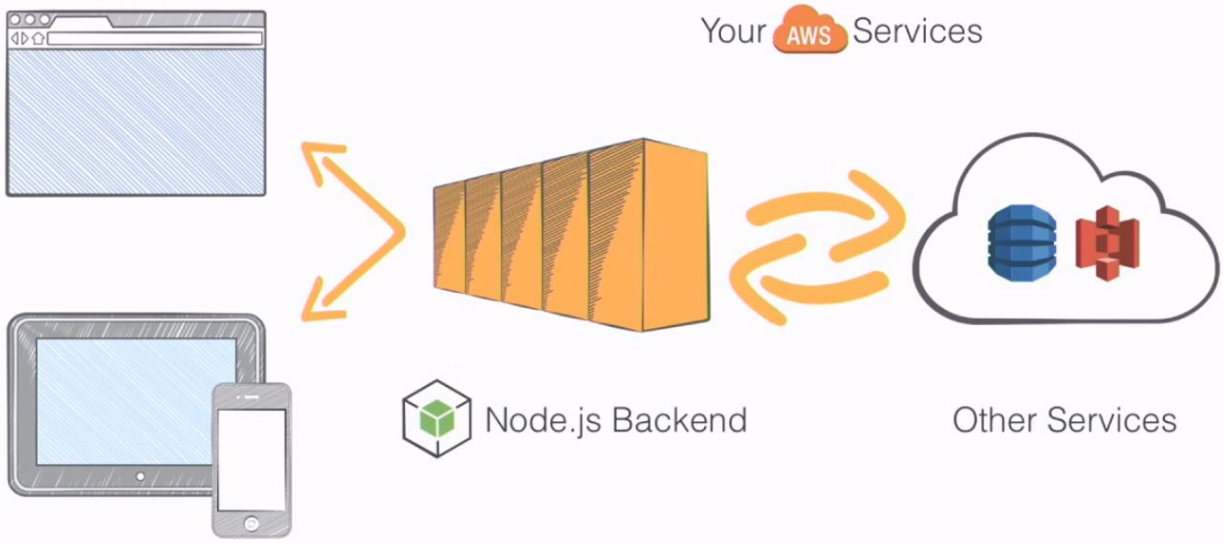
But in your browser or mobile device

Configuration is Different

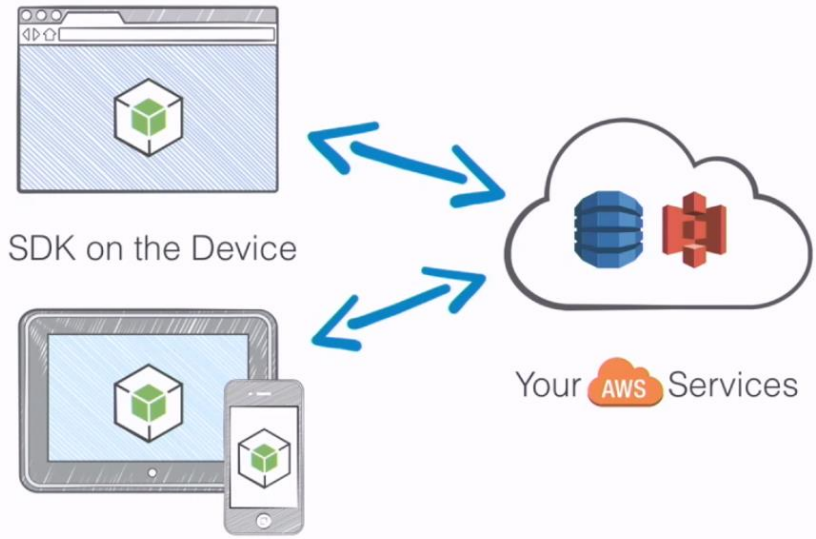
Why is it different?

Two-Tier Web Applications

Traditional Application Architecture



Two-Tier Application Architecture



Benefits

- Fewer moving parts
- Easy prototyping
- Deploying as simple as copying files to Amazon S3
- Fully dynamic app for pennies a month

Next Level Web Apps

App Ideas

- Forum Software
- Blog Commenting Service
- Blogging Platform
- Firefox/Chrome Extensions
- WinRT (Metro Style) Apps
- Any Mobile App!

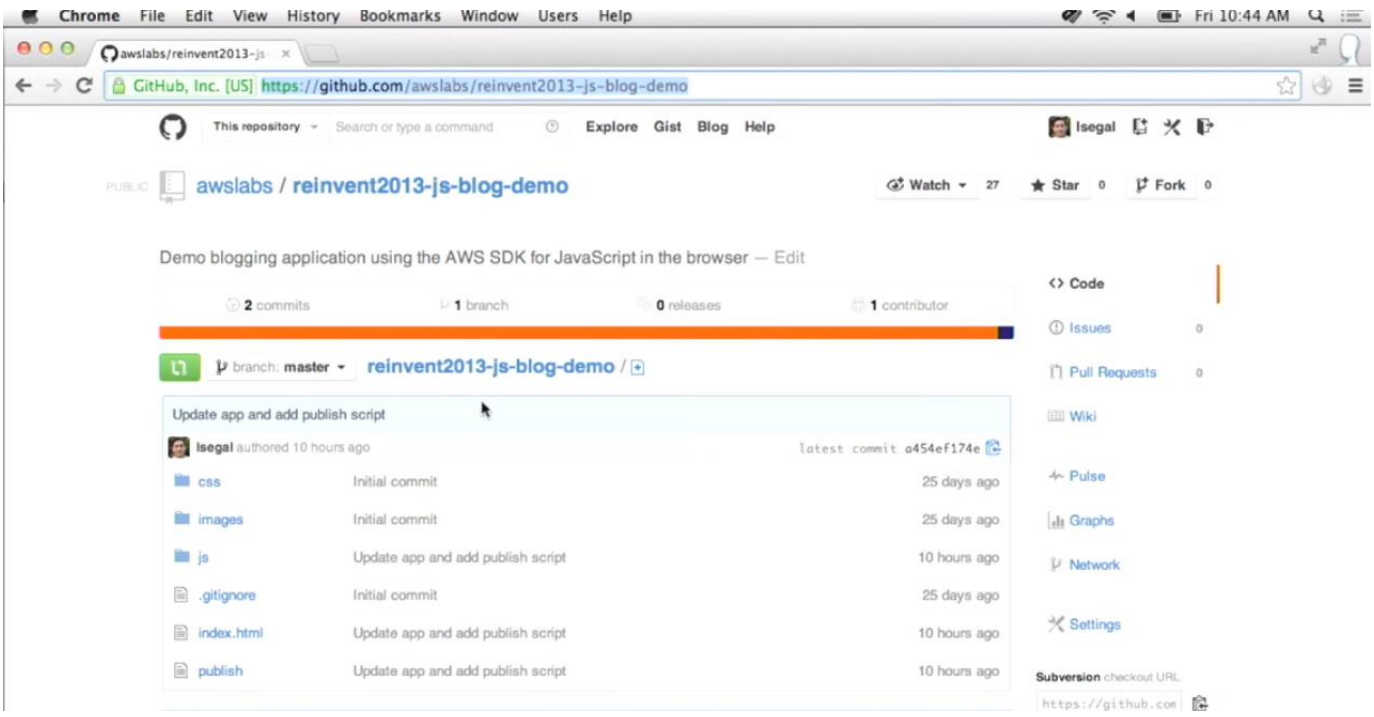
Let's Look at a Web Application

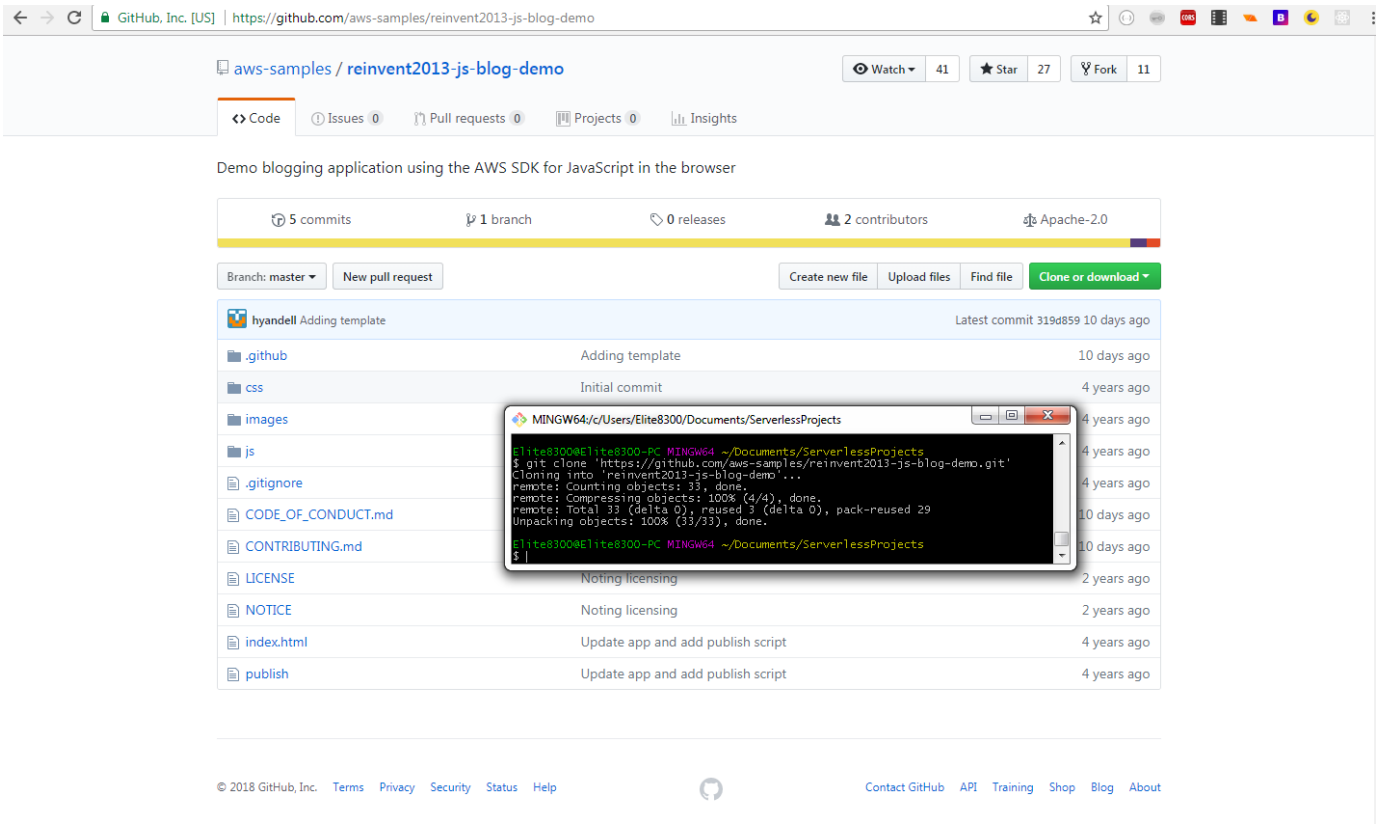
Using nothing but
HTML, CSS, and JavaScript

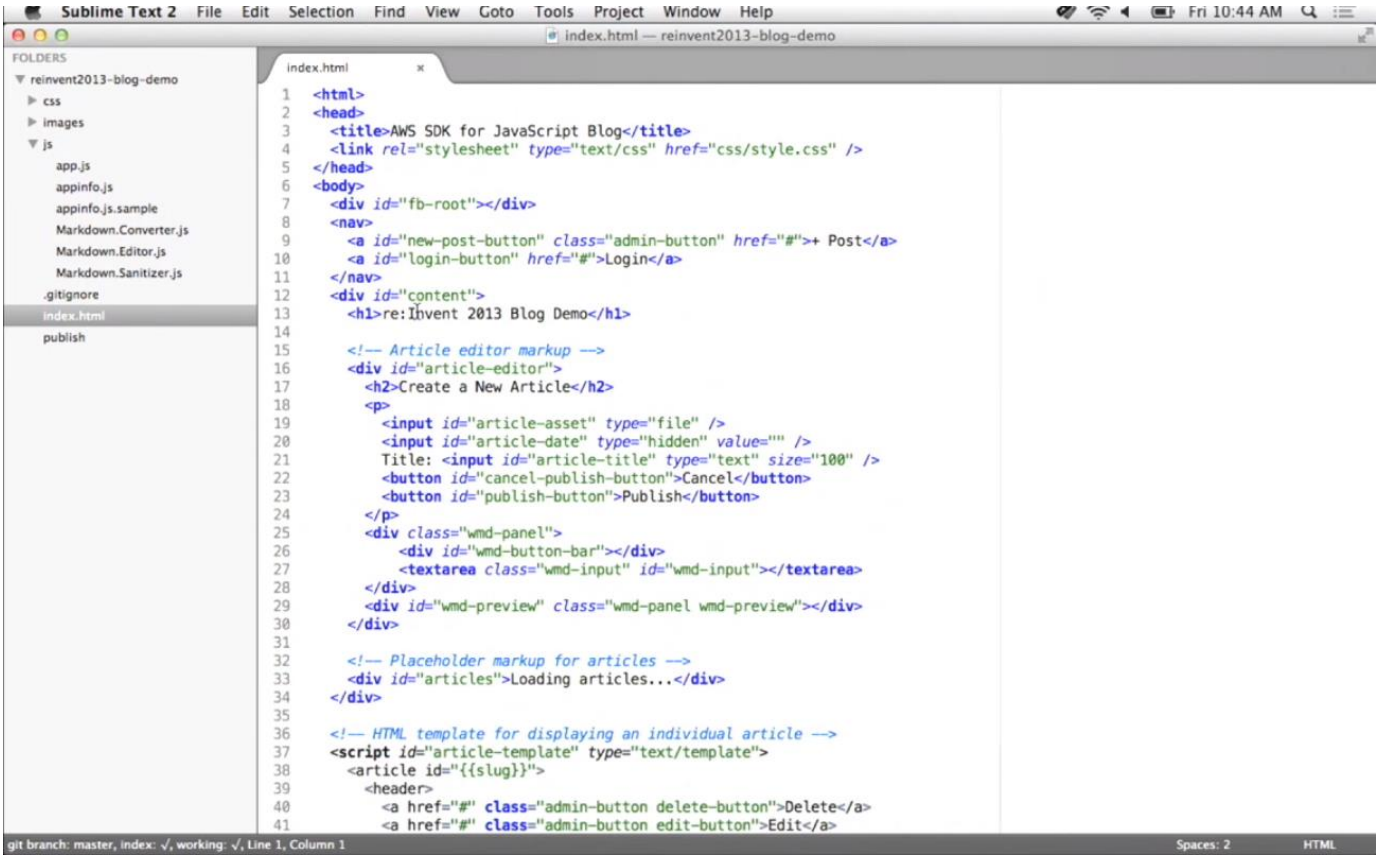
A Simple Blog

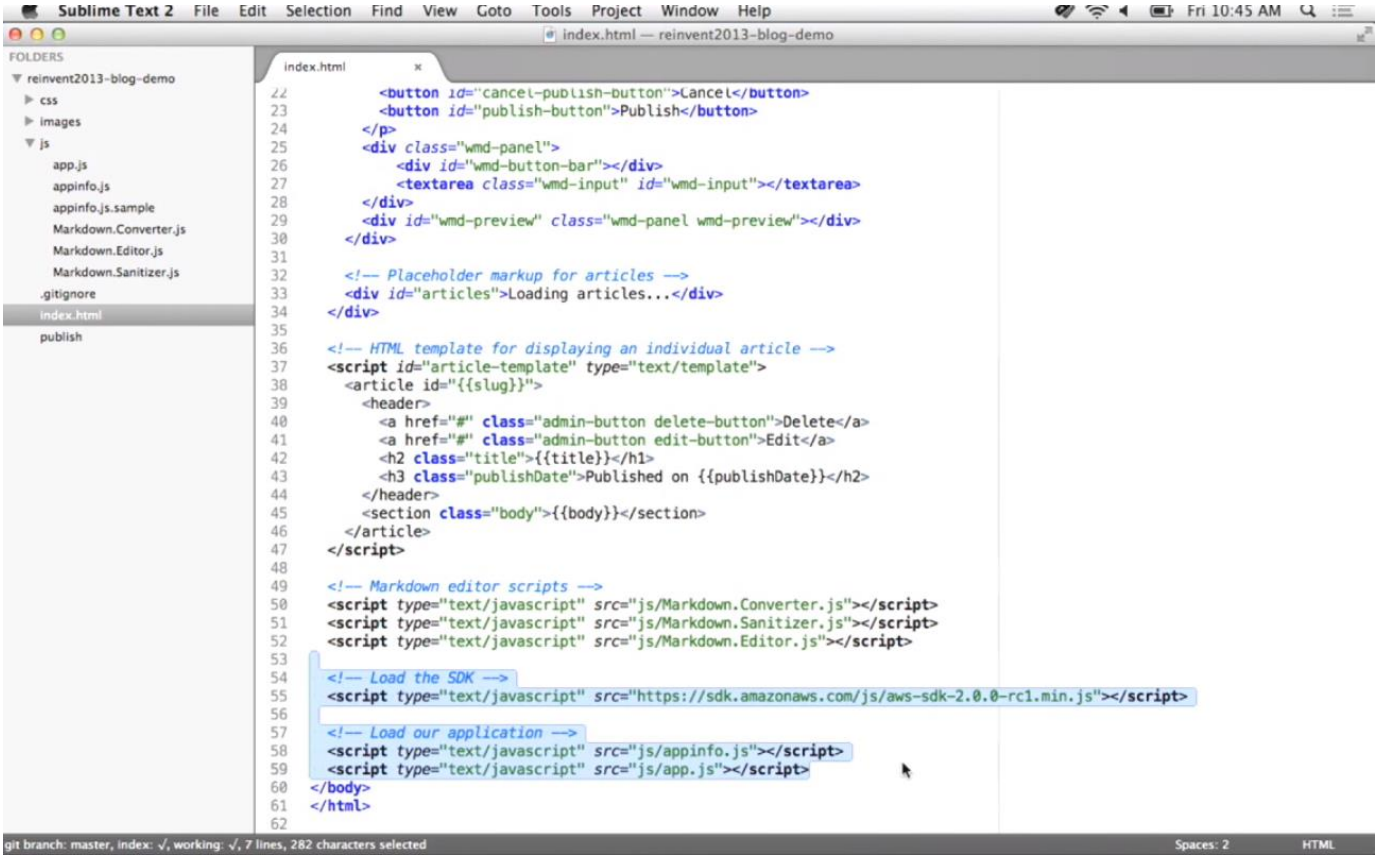
Content stored in Amazon DynamoDB

Assets in Amazon S3

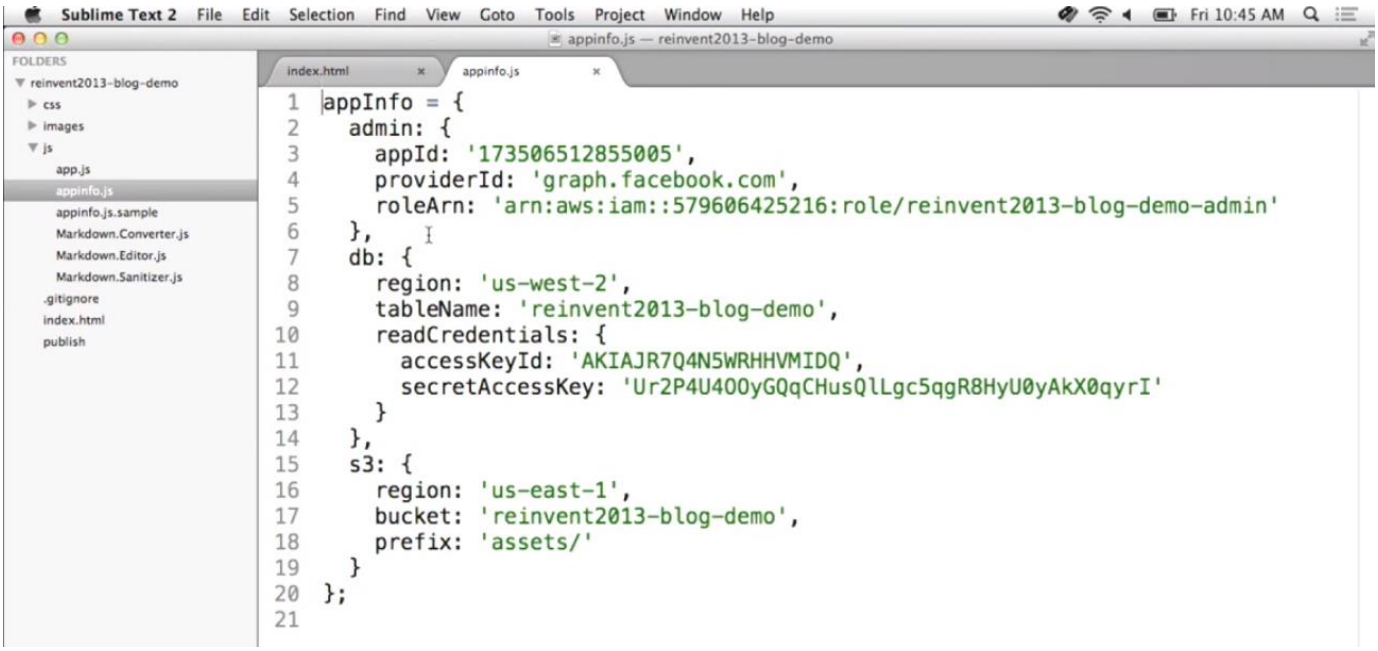



GitHub, Inc. [US] | https://github.com/aws-samples/reinvent2013-js-blog-demo
aws-samples / reinvent2013-js-blog-demo
Watch 41
Star 27
Fork 11
Code
Issues 0
Pull requests 0
Projects 0
Insights
Demo blogging application using the AWS SDK for JavaScript in the browser
5 commits
1 branch
0 releases
2 contributors
Apache-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
hyandell Adding template
Latest commit 319d859 10 days ago
.github Adding template 10 days ago
css Initial commit 4 years ago
images 4 years ago
js 4 years ago
.gitignore 4 years ago
CODE_OF_CONDUCT.md 10 days ago
CONTRIBUTING.md 10 days ago
LICENSE Noting licensing 2 years ago
NOTICE Noting licensing 2 years ago
index.html Update app and add publish script 4 years ago
publish Update app and add publish script 4 years ago
MINGW64:/c/Users/Elite8300/Documents/ServerlessProjects
Elite8300@Elite8300-PC MINGW64 ~/Documents/ServerlessProjects
$ git clone 'https://github.com/aws-samples/reinvent2013-js-blog-demo.git'
Cloning into 'reinvent2013-js-blog-demo'...
remote: Counting objects: 33, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 33 (delta 0), reused 3 (delta 0), pack-reused 29
Unpacking objects: 100% (33/33), done.
Elite8300@Elite8300-PC MINGW64 ~/Documents/ServerlessProjects
$
© 2018 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub API Training Shop Blog About

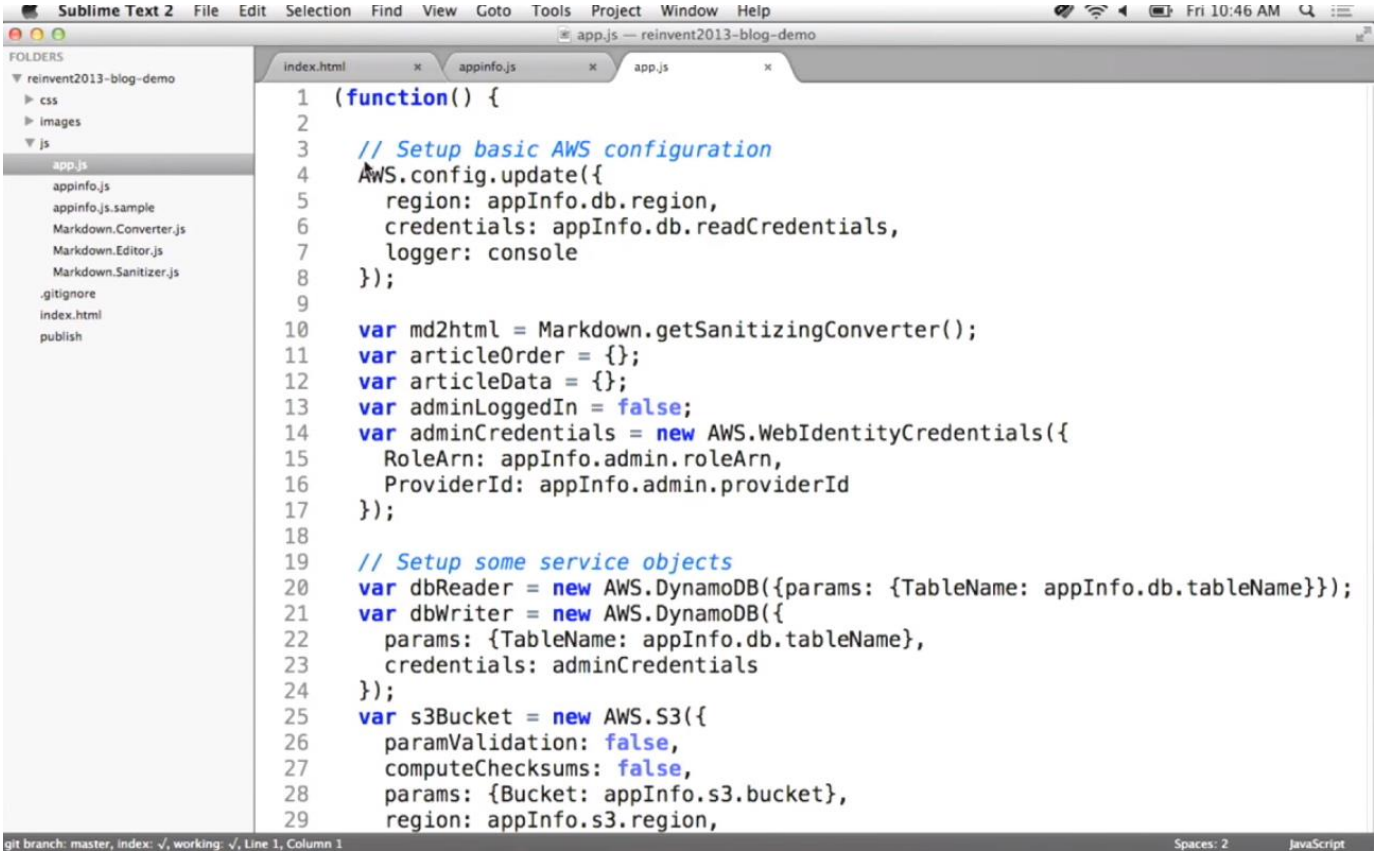

Sublime Text 2 File Edit Selection Find View Goto Tools Project Window Help
Fri 10:44 AM
index.html — reinvent2013-blog-demo
FOLDERS
reinvent2013-blog-demo
css
images
js
app.js
appinfo.js
appinfo.js.sample
Markdown.Converter.js
Markdown.Editor.js
Markdown.Sanitizer.js
.gitignore
index.html
publish
index.html
1 <html>
2 <head>
3 <title>AWS SDK for JavaScript Blog</title>
4 <link rel="stylesheet" type="text/css" href="css/style.css" />
5 </head>
6 <body>
7
8 <nav>
9 + Post
10 Login
11 </nav>
12
13 <h1>re:Invent 2013 Blog Demo</h1>
14
15 <!-- Article editor markup -->
16
17 <h2>Create a New Article</h2>
18
19 <input id="article-asset" type="file" />
20 <input id="article-date" type="hidden" value="" />
21 Title: <input id="article-title" type="text" size="100" />
22 <button id="cancel-publish-button">Cancel</button>
23 <button id="publish-button">Publish</button>
24
25
26
27 <textarea class="wmd-input" id="wmd-input"></textarea>
28
29
30
31
32 <!-- Placeholder markup for articles -->
33 Loading articles...
34
35
36 <!-- HTML template for displaying an individual article -->
37 <script id="article-template" type="text/template">
38 <article id="{{slug}}">
39 <header>
40 Delete
41 Edit
git branch: master, index: √, working: √, Line 1, Column 1
Spaces: 2
HTML

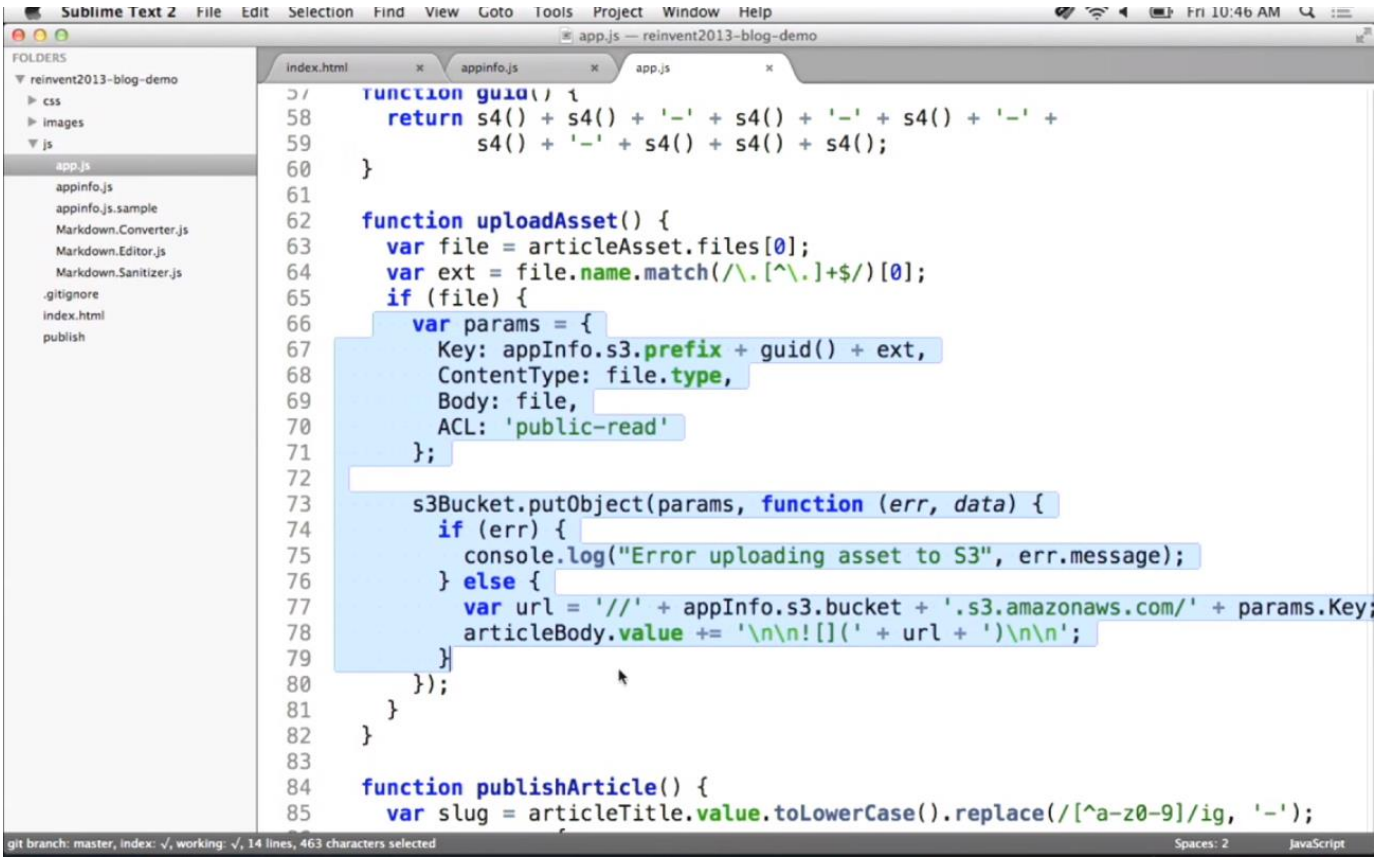


We load the SDK at the bottom of the UI page as above

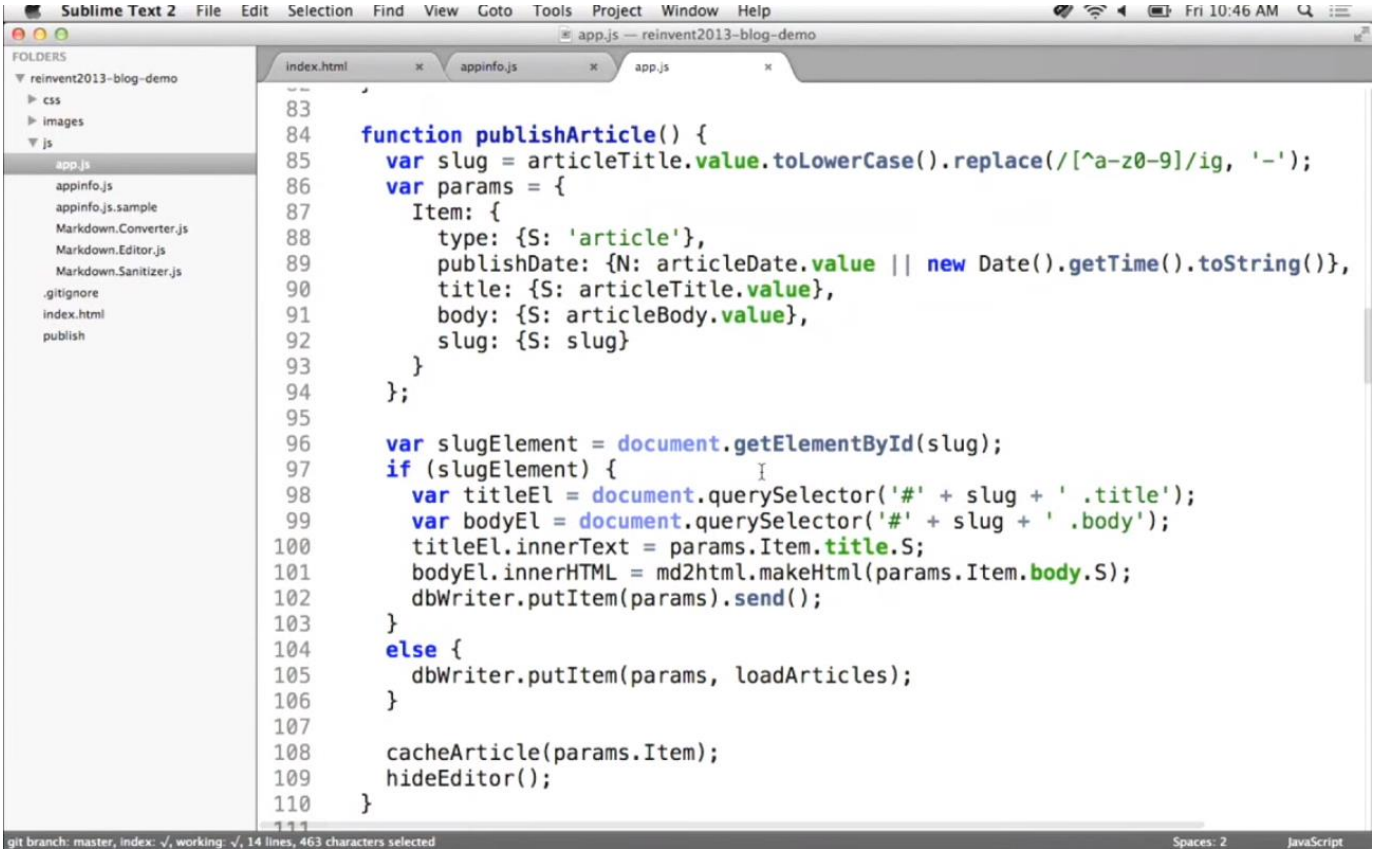


This file contains all of our SDK configuration details, we are using a very scoped credentials that only has read-only access to the relevant DynamoDB table only. We also have the details for the public S3 bucket where our app is going to be getting the static asset files from.

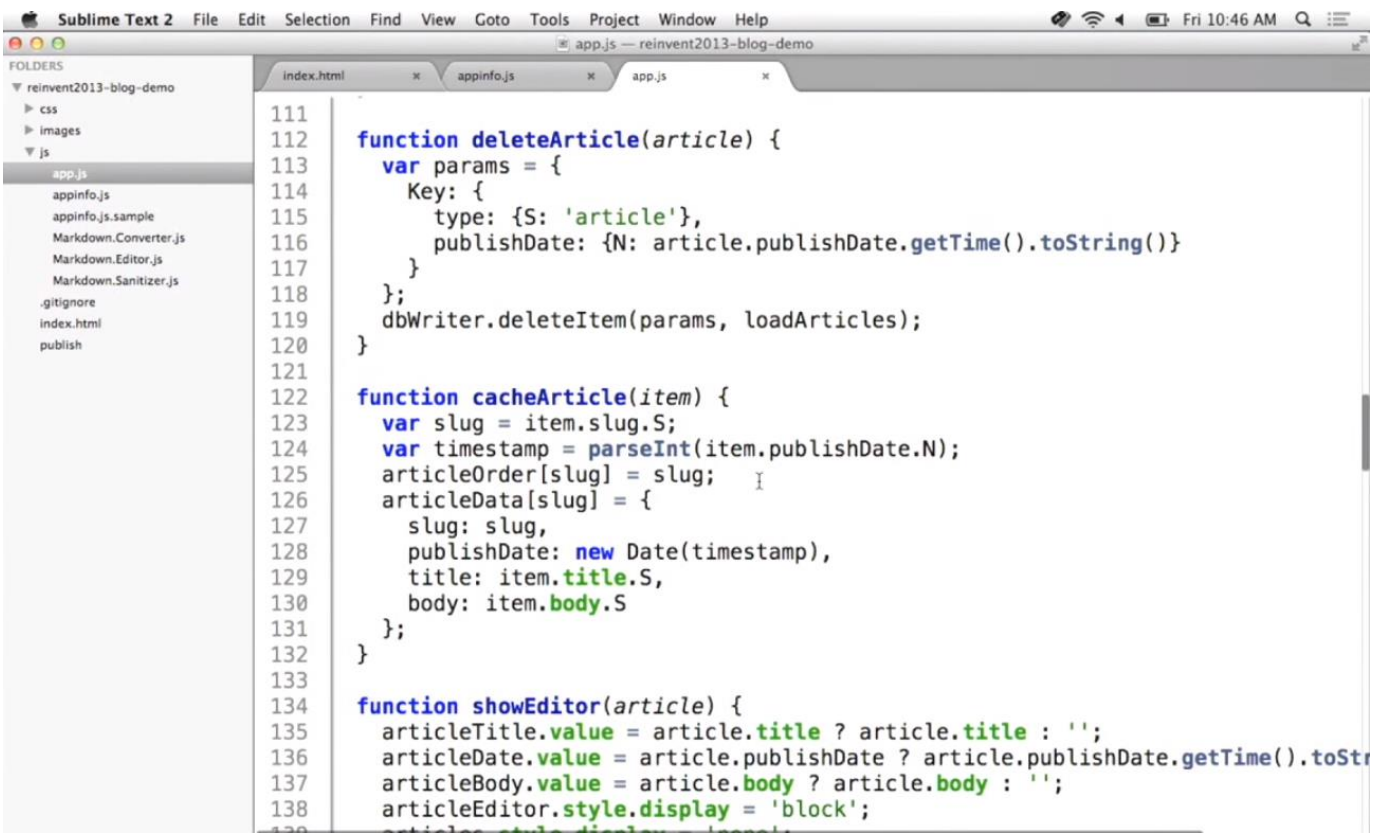


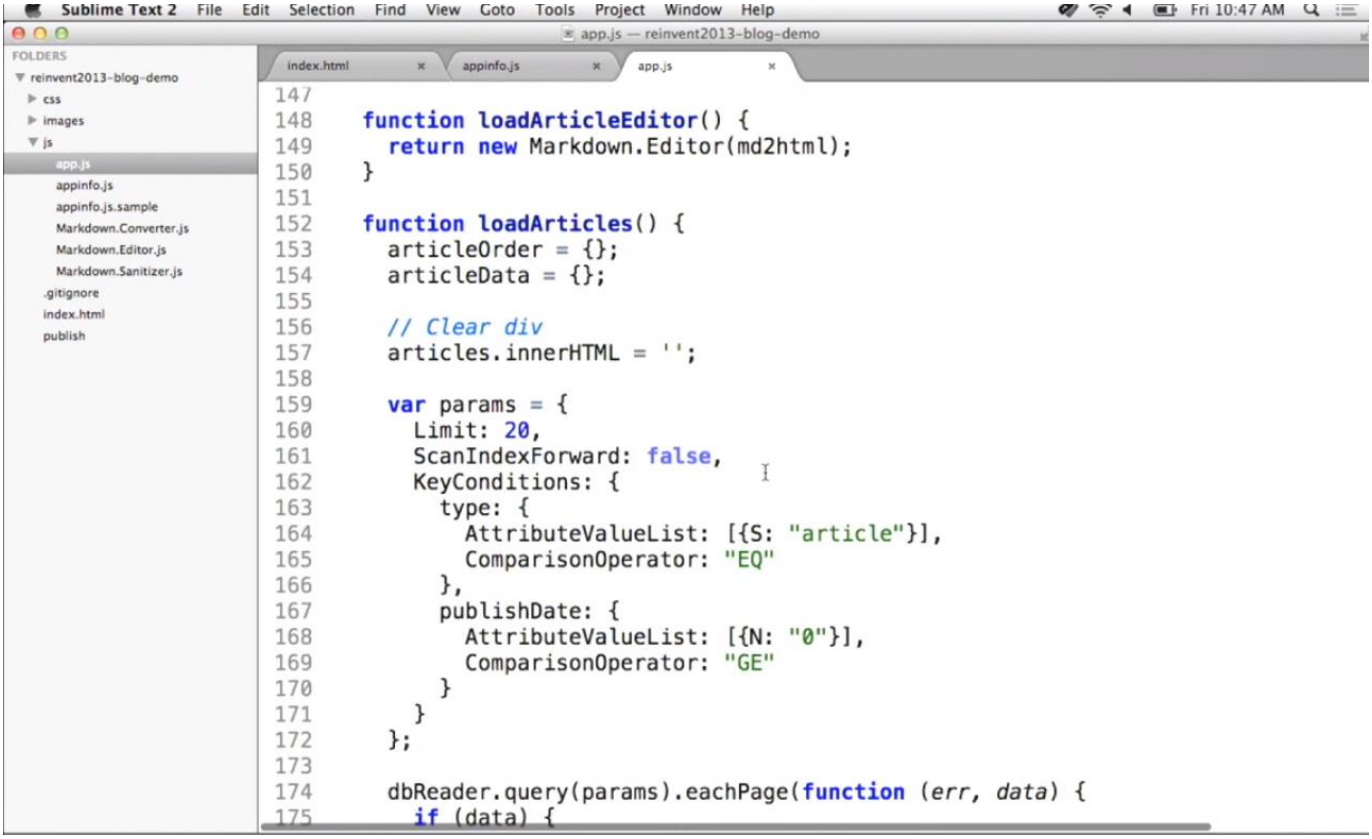


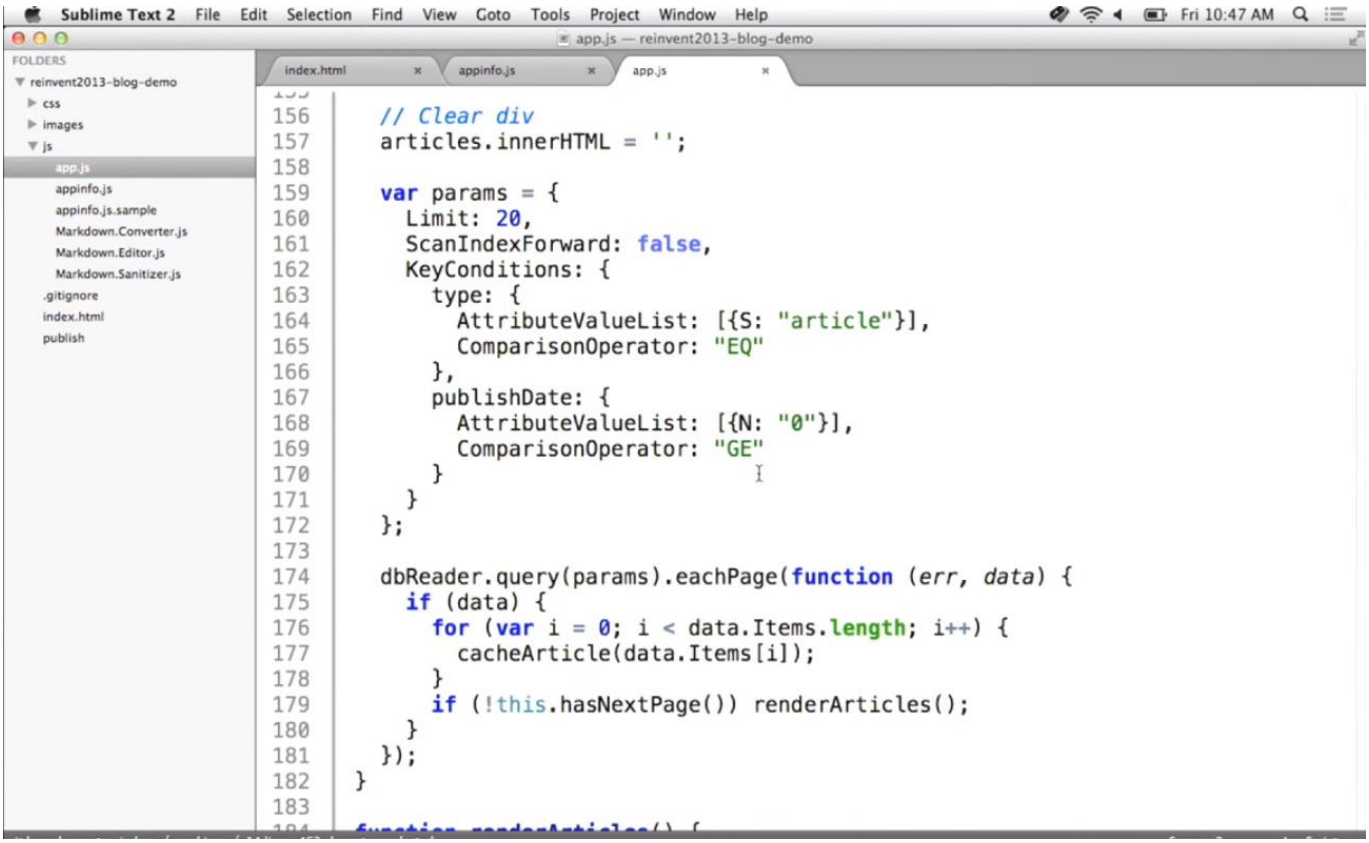
This is where we are uploading the assets into S3



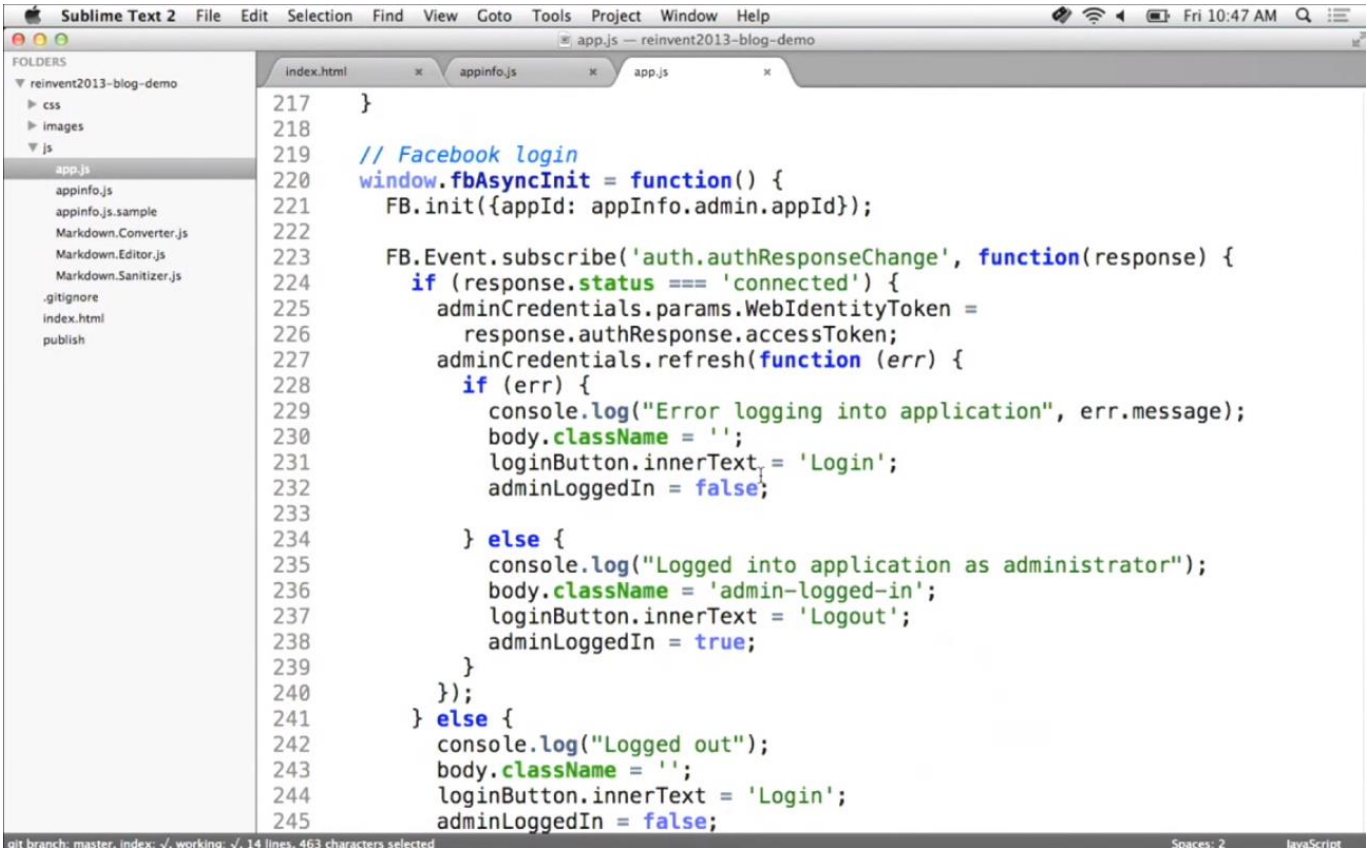
This is the code that interacts with DynamoDB



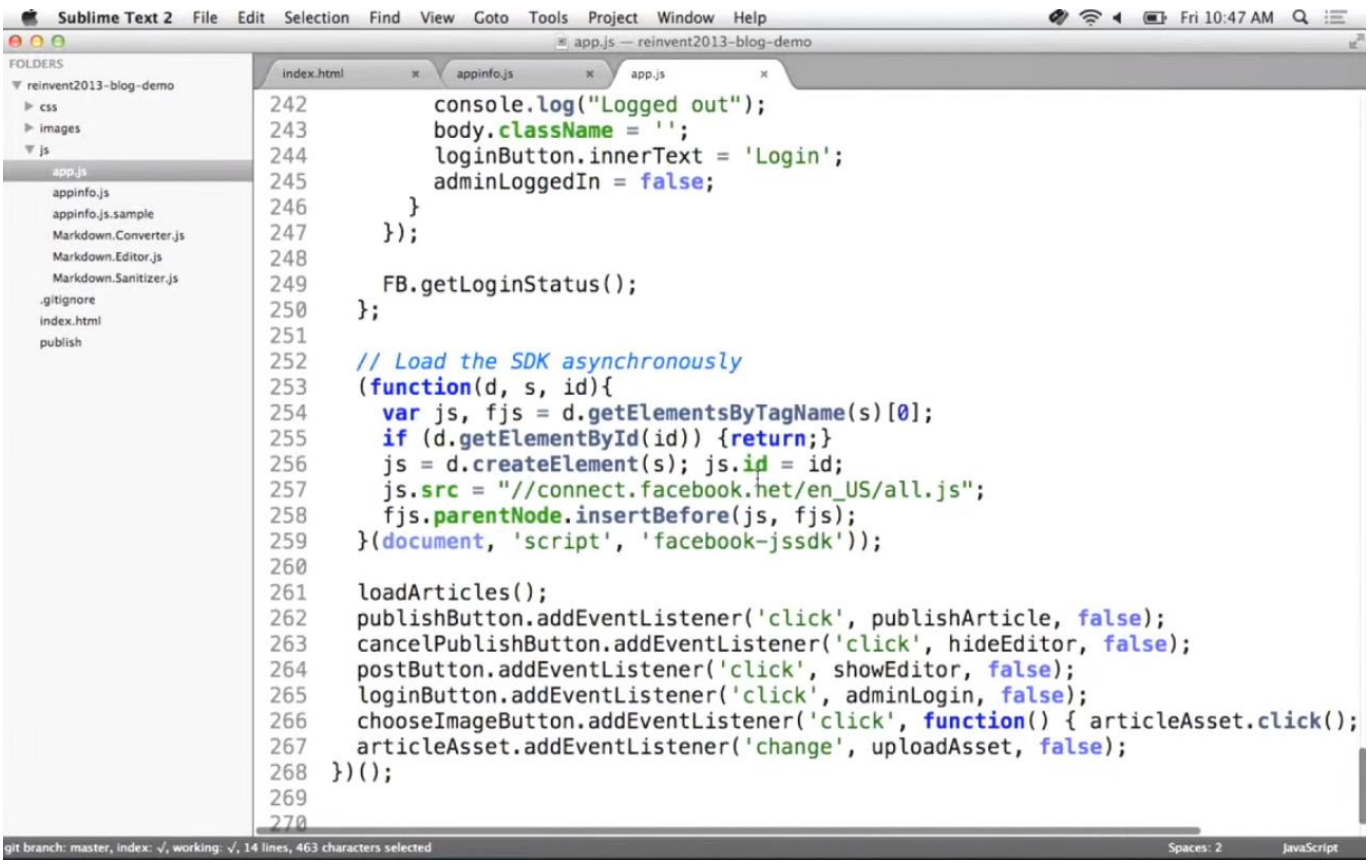


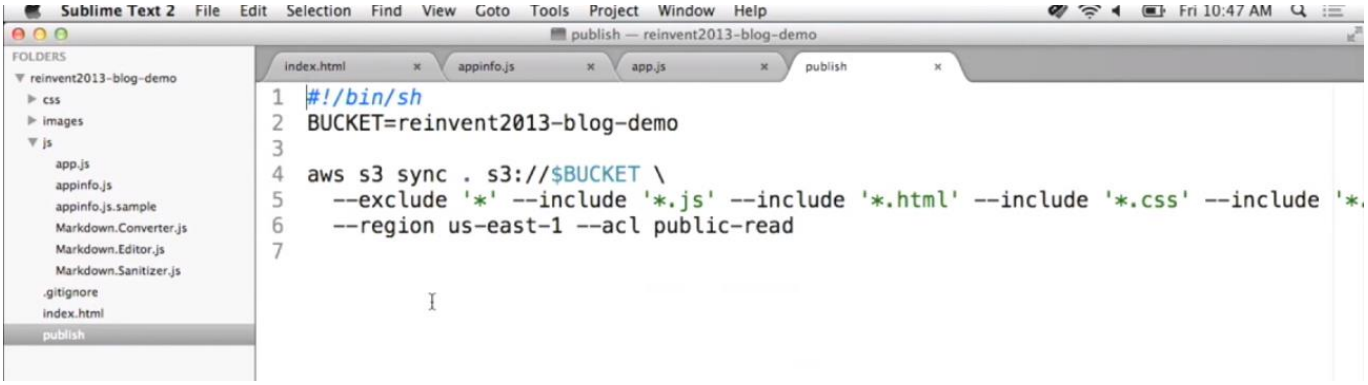


We are doing a simple query here and using response pagination to get each available page

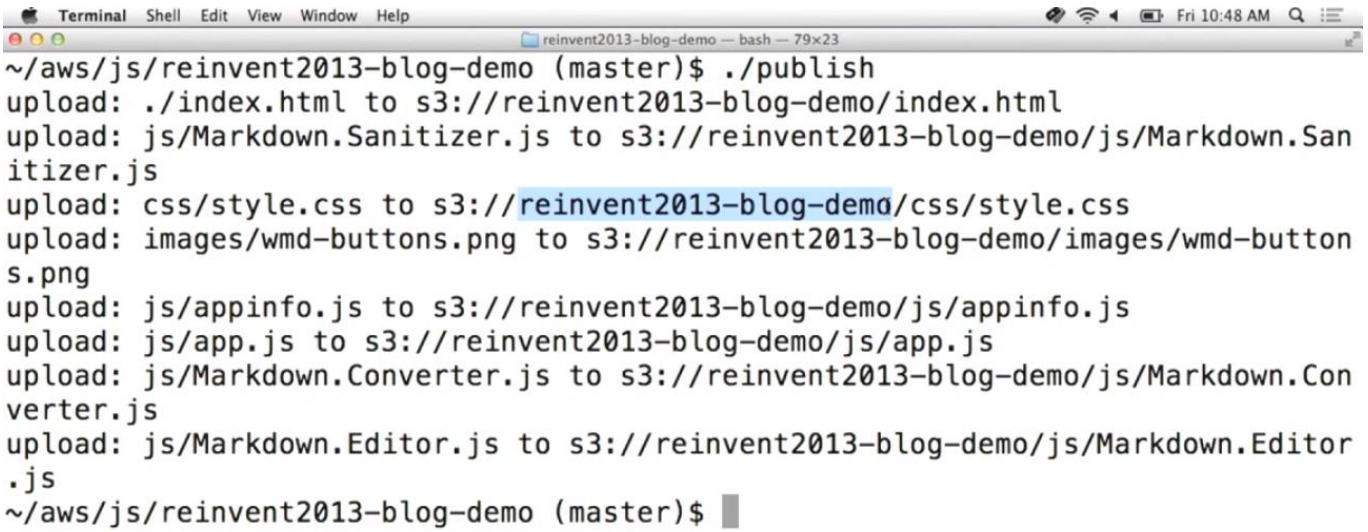


This is the OAuth login

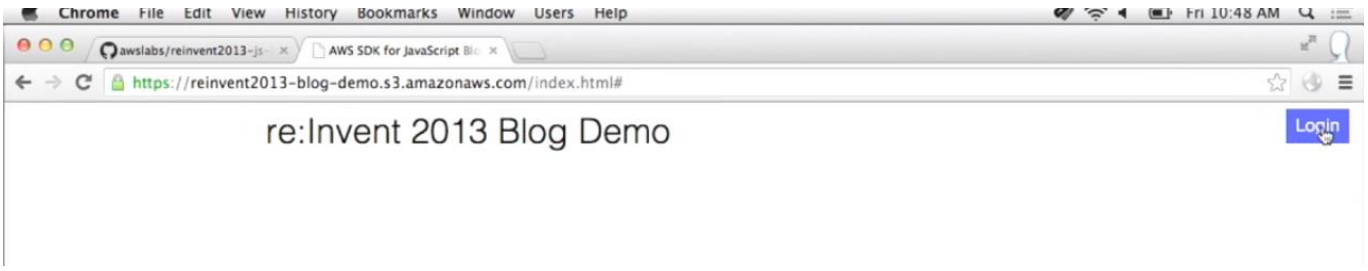


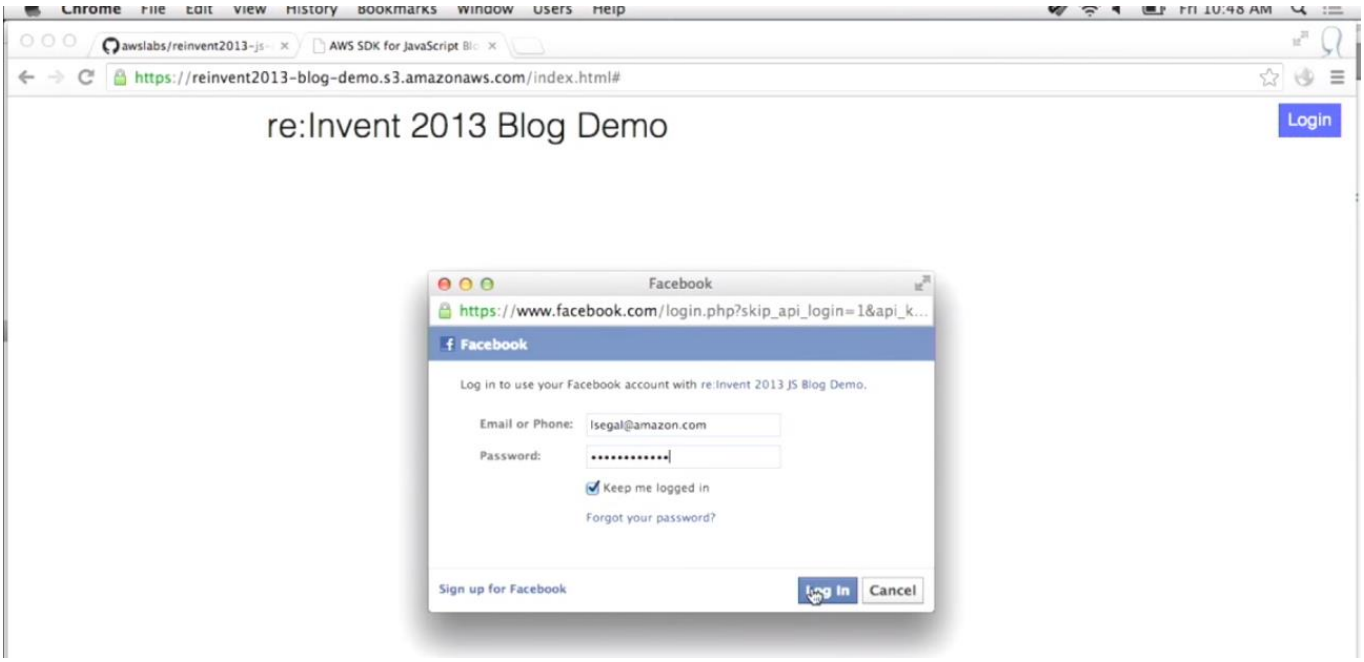


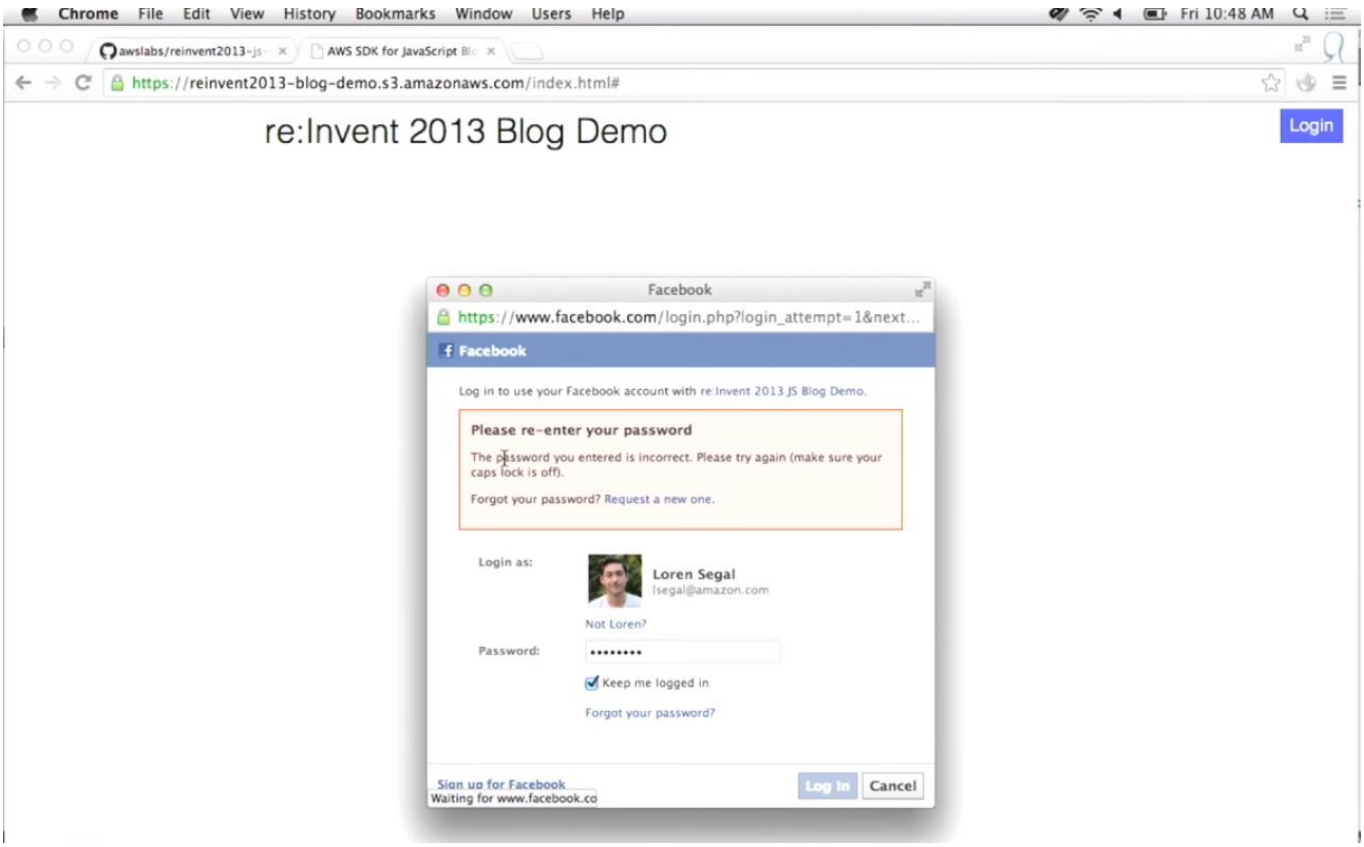
We can then use the CLI to deploy the application using the command above. We are using the AWS S3 sync command to publish all our assets to the S3 bucket.

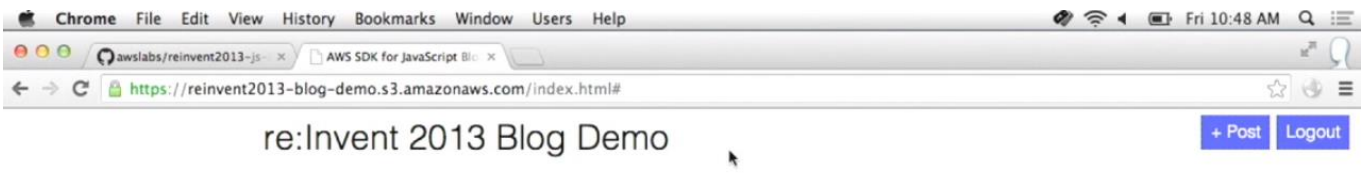


The CLI has now deployed our app into the S3 bucket successfully.

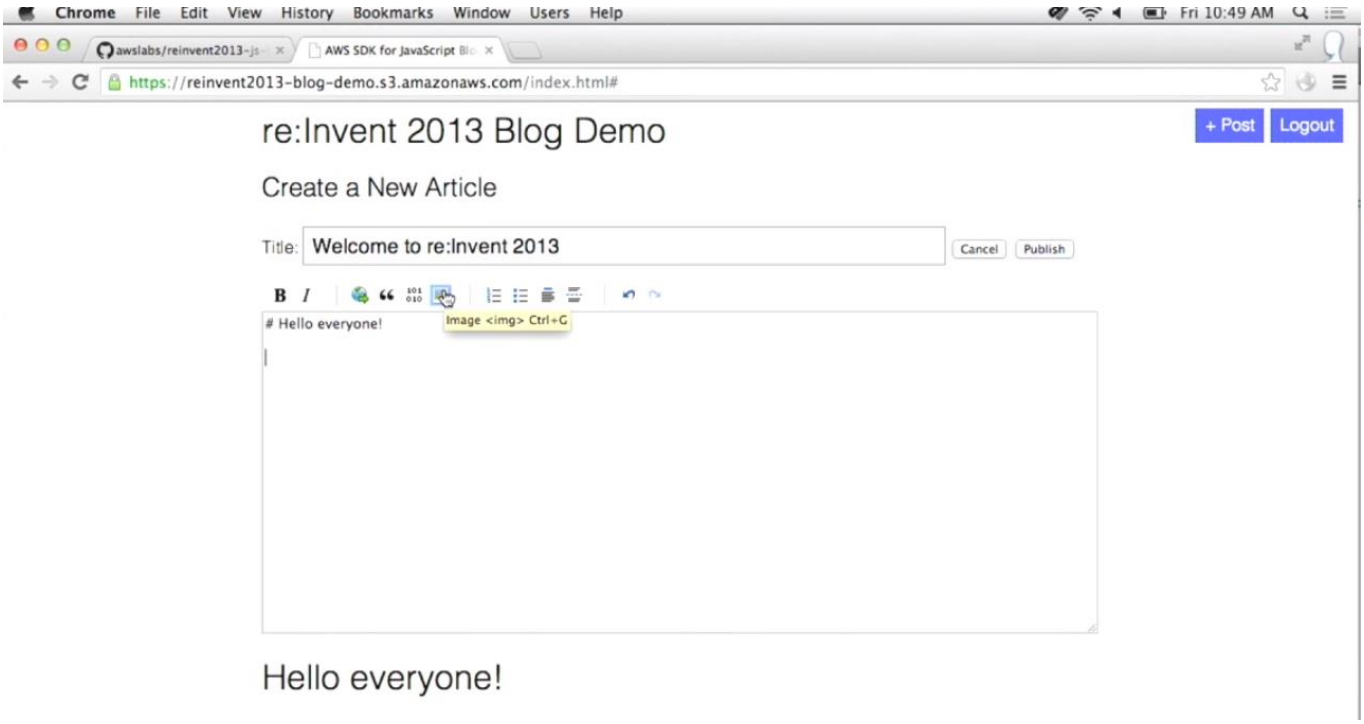


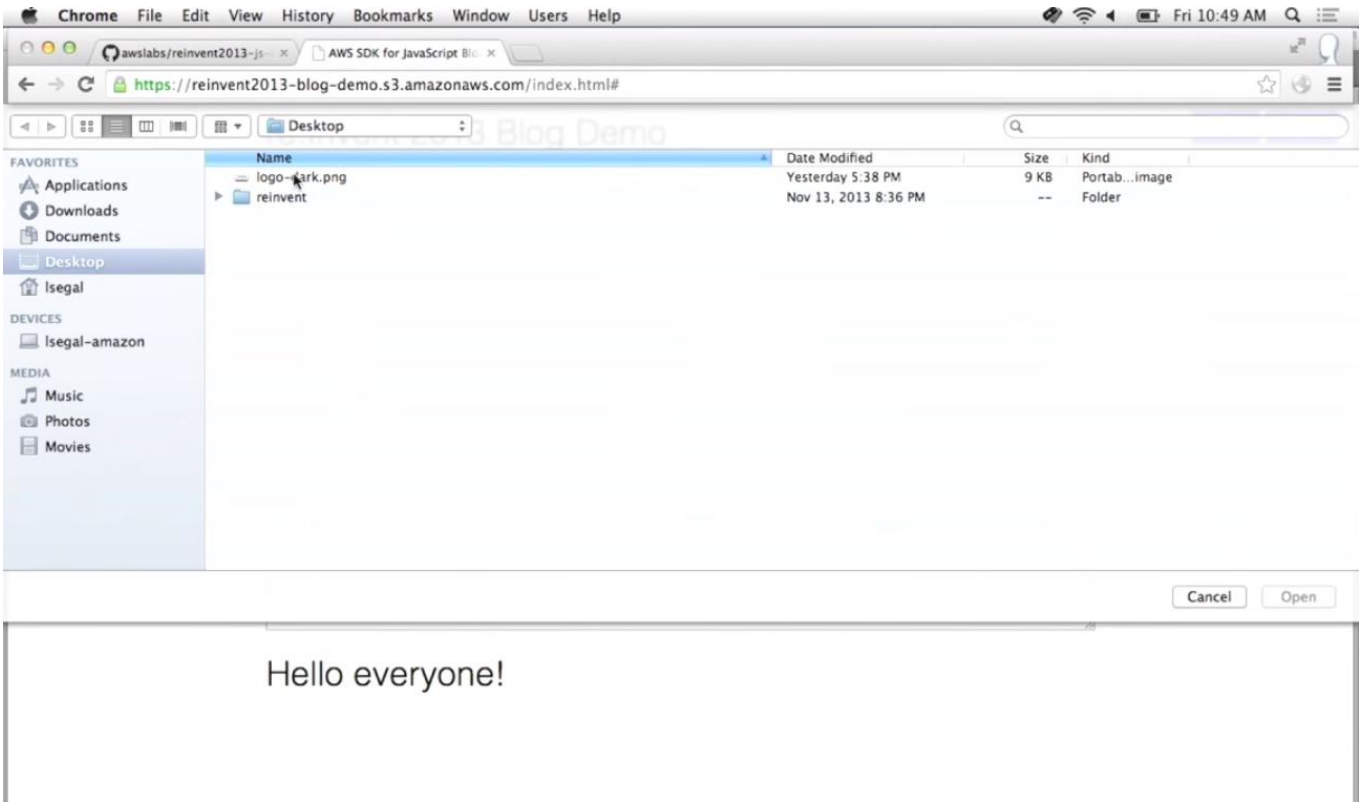


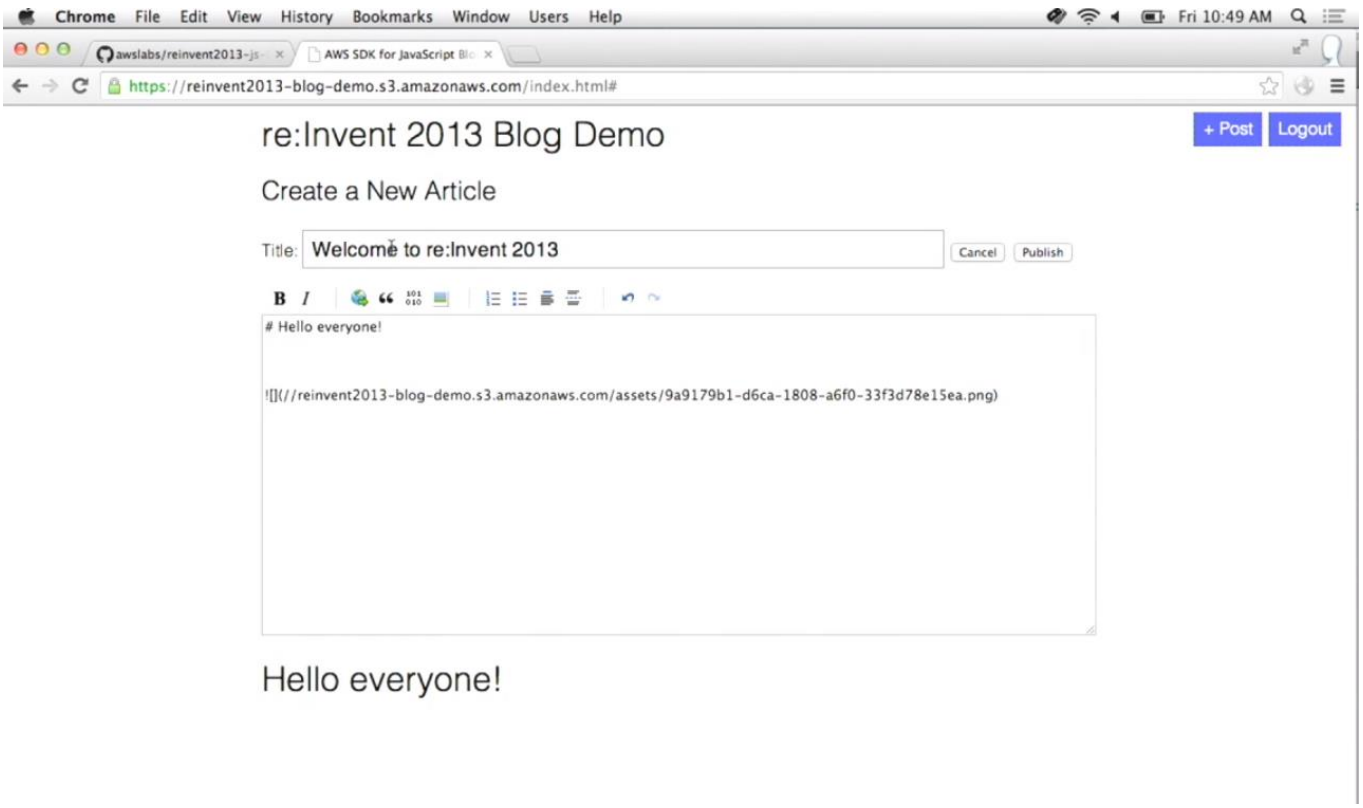




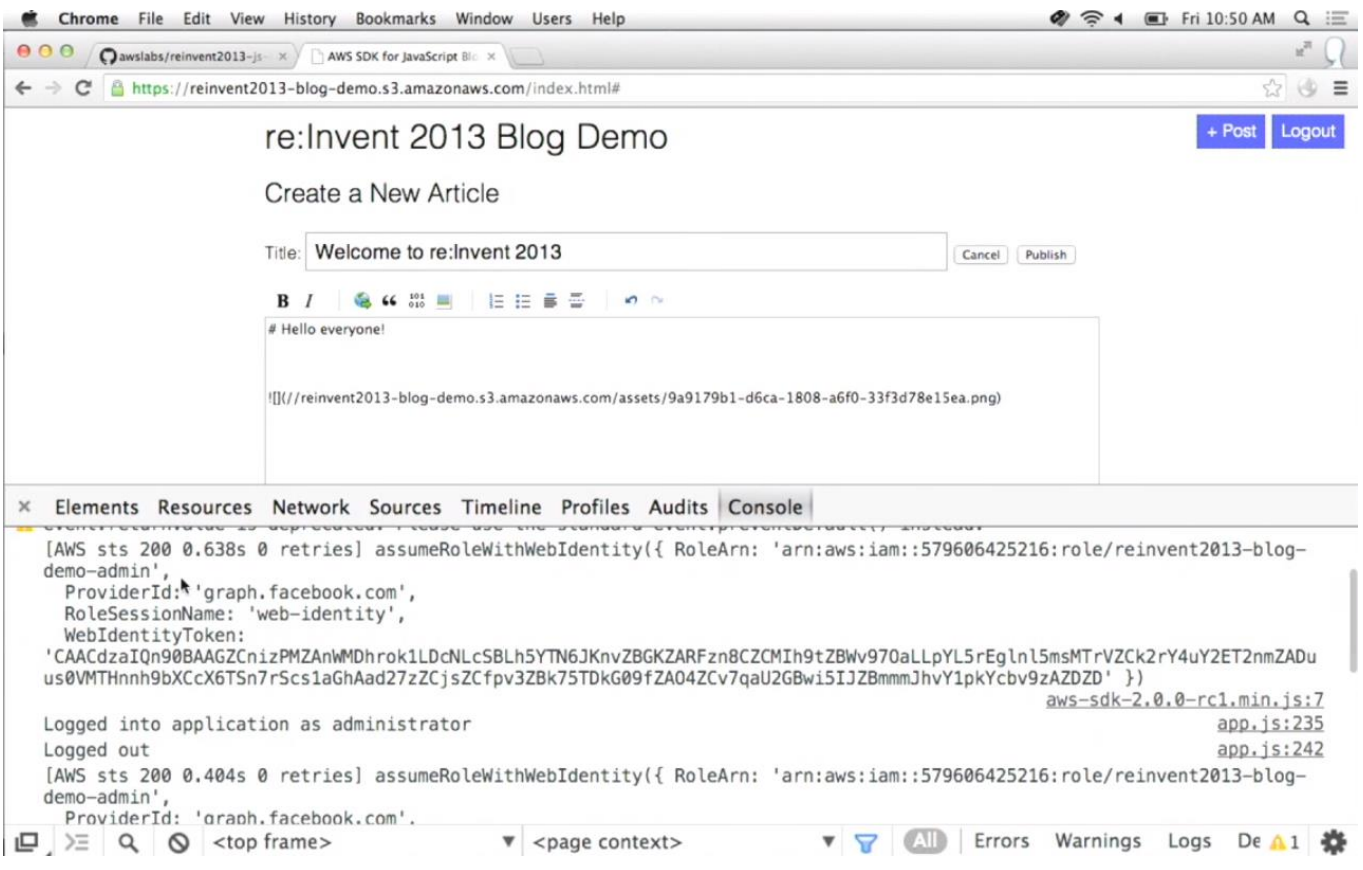
We can now make a post as an admin


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:49 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title:
Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!
Image <img> Ctrl+G
Hello everyone!

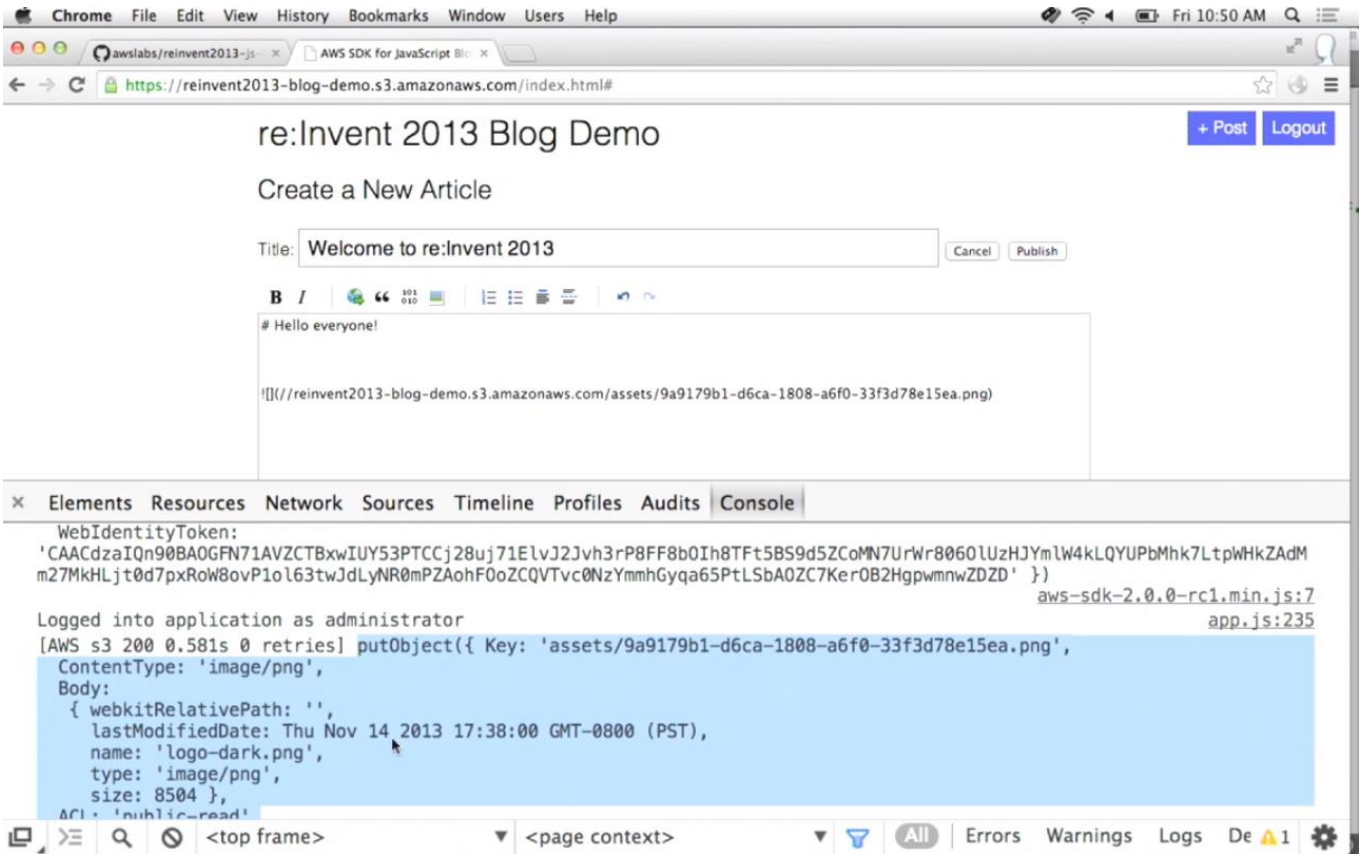

Chrome File Edit View History Bookmarks Window Users Help
Fri 10:49 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
Desktop
FAVORITES
Applications
Downloads
Documents
Desktop
lsegal
DEVICES
lsegal-amazon
MEDIA
Music
Photos
Movies
Name
Date Modified
Size
Kind
logo-dark.png
Yesterday 5:38 PM
9 KB
Portab...image
reinvent
Nov 13, 2013 8:36 PM
--
Folder
Cancel
Open
Hello everyone!


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:49 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title:
Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

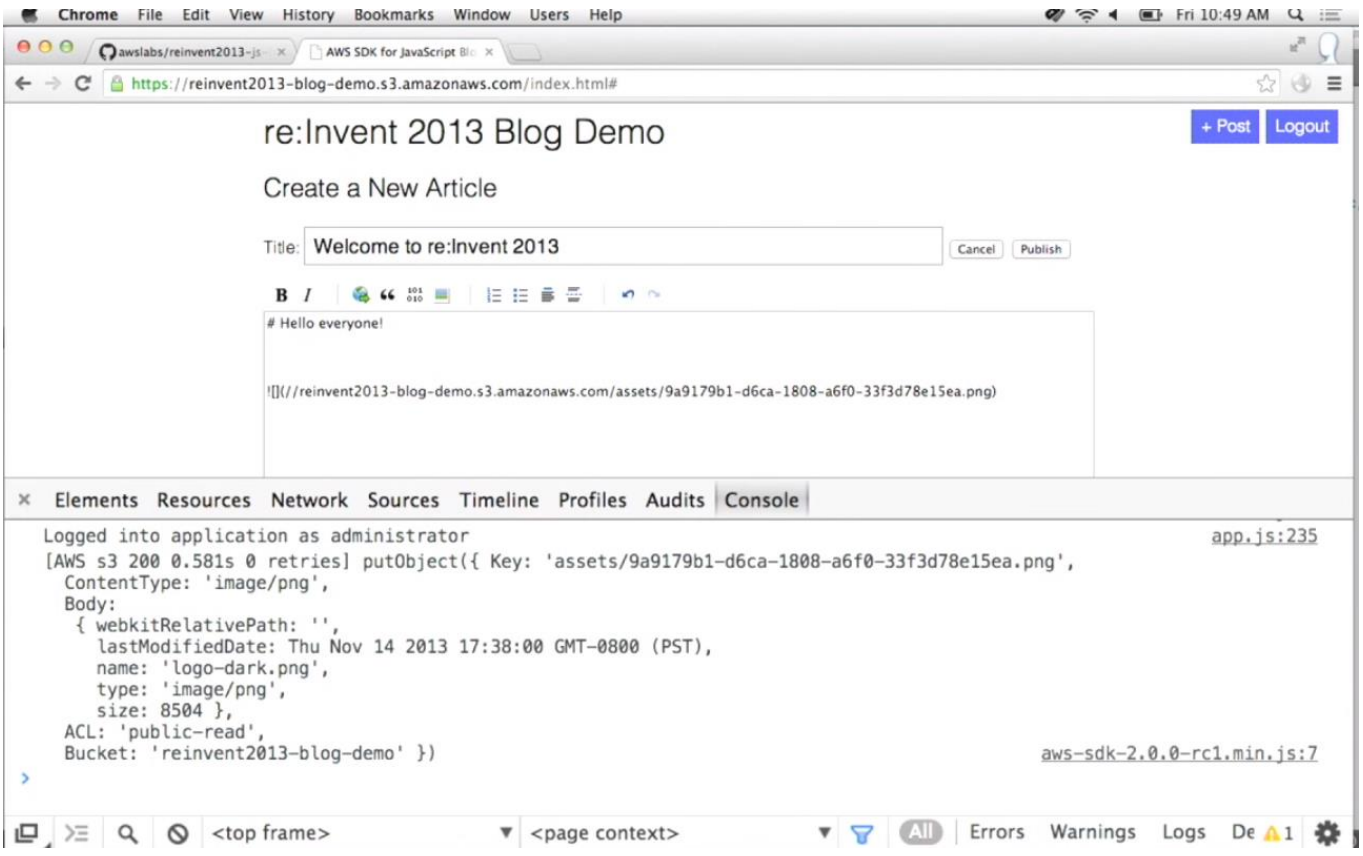
Hello everyone!


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:50 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title:
Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

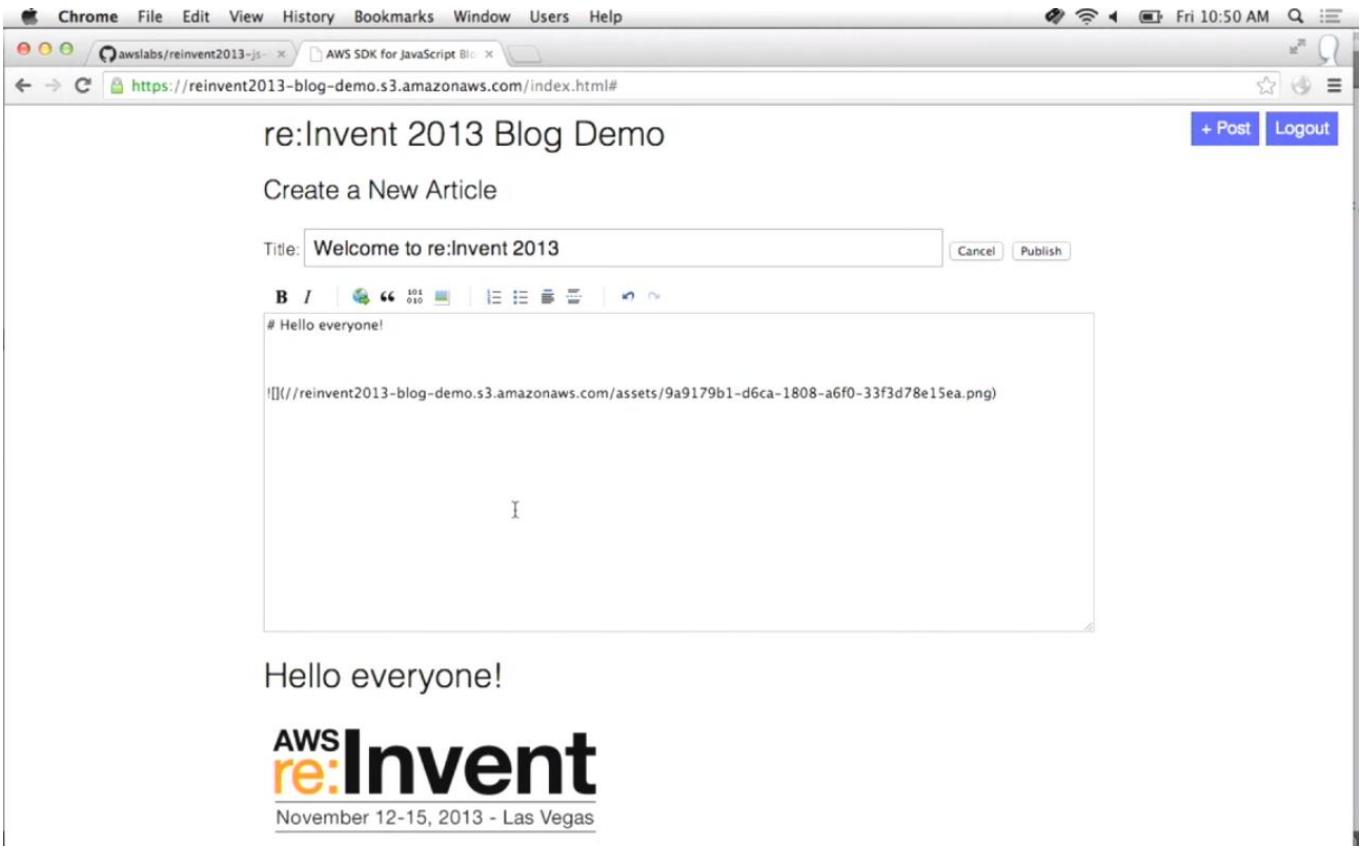
Elements Resources Network Sources Timeline Profiles Audits Console
[AWS sts 200 0.638s 0 retries] assumeRoleWithWebIdentity({ RoleArn: 'arn:aws:iam::579606425216:role/reinvent2013-blog-demo-admin',
ProviderId: 'graph.facebook.com',
RoleSessionName: 'web-identity',
WebIdentityToken:
'CAACdzaIQn90BAAGZCnizPMZAnWMDhrok1LDcNLcSBLh5YTN6JKnvZBGKZARFzn8CZCMIh9tZBWv97OaLLpYL5rEglnl5msMTrVZCk2rY4uY2ET2nmZADuus0VMTHnnh9bXCcX6TSn7rScs1aGhAad27zZCjsZCfpv3ZBk75TDkG09fZAO4ZCv7qaU2GBwi5IJZBmmmJhvY1pkYcbv9zAZDZD' })
aws-sdk-2.0.0-rc1.min.js:7
Logged into application as administrator
app.js:235
Logged out
app.js:242
[AWS sts 200 0.404s 0 retries] assumeRoleWithWebIdentity({ RoleArn: 'arn:aws:iam::579606425216:role/reinvent2013-blog-demo-admin',
ProviderId: 'graph.facebook.com',
<top frame>
<page context>
All
Errors
Warnings
Logs
1


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:50 AM
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title: Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

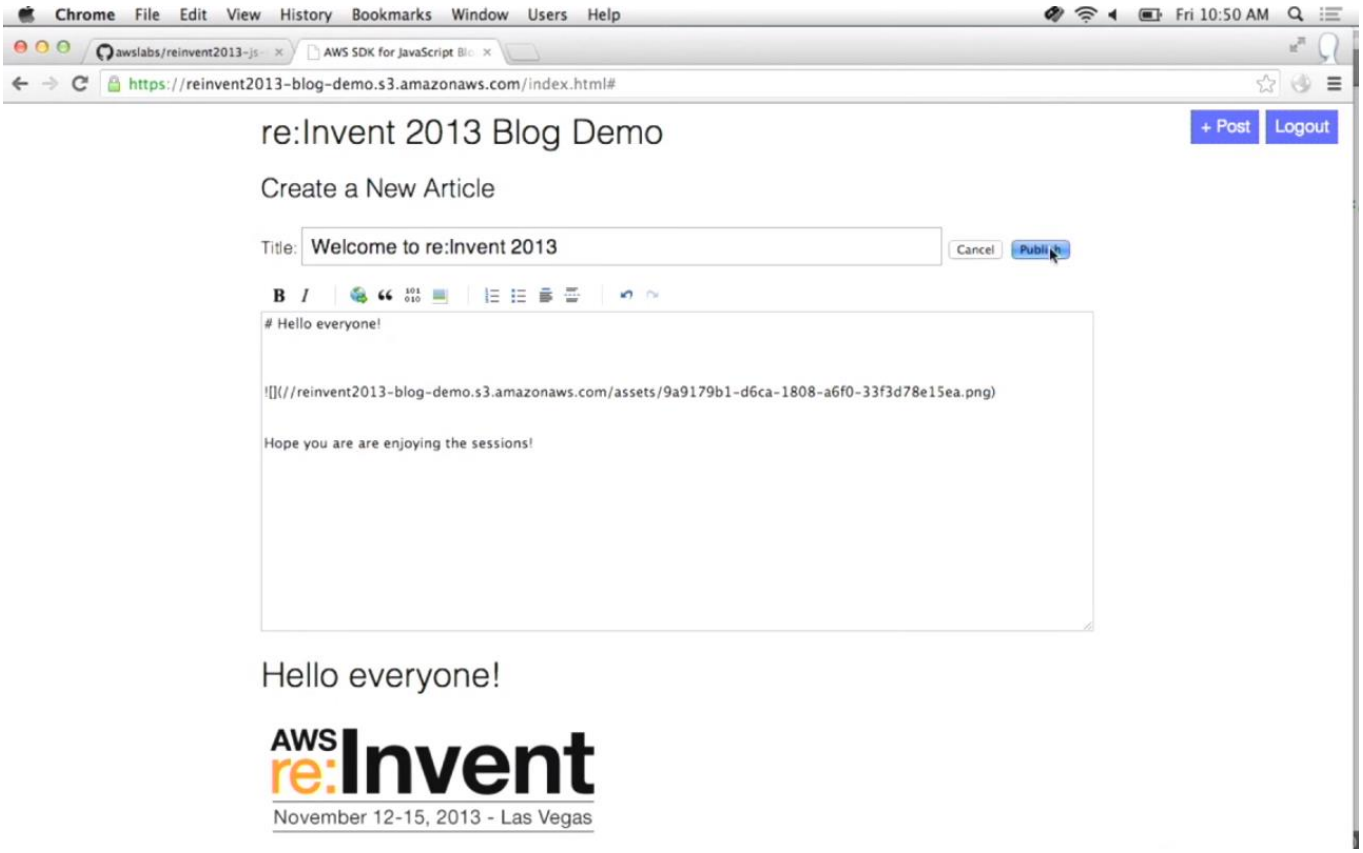
Elements Resources Network Sources Timeline Profiles Audits Console
WebIdentityToken:
'CAACdzaIQn90BAOGFN71AVZCTBxwIUY53PTCCj28uj71ElvJ2Jvh3rP8FF8bOIh8TFt5BS9d5ZCoMN7UrWr806OlUzHJYmlW4kLQYUPbMhk7LtpWHkZAdM
m27MkHLjt0d7pxRoW8ovP1ol63twJdLyNR0mPZAohFOoZCQVTvc0NzYmmhGyqa65PtLSbAOZC7KerOB2HgpwmnwZDZD' })
aws-sdk-2.0.0-rc1.min.js:7
Logged into application as administrator
app.js:235
[AWS s3 200 0.581s 0 retries] putObject({ Key: 'assets/9a9179b1-d6ca-1808-a6f0-33f3d78e15ea.png',
ContentType: 'image/png',
Body:
{ webkitRelativePath: '',
lastModifiedDate: Thu Nov 14 2013 17:38:00 GMT-0800 (PST),
name: 'logo-dark.png',
type: 'image/png',
size: 8504 },
ACL: 'public-read'
<top frame>
<page context>
All
Errors
Warnings
Logs


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:49 AM
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title: Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

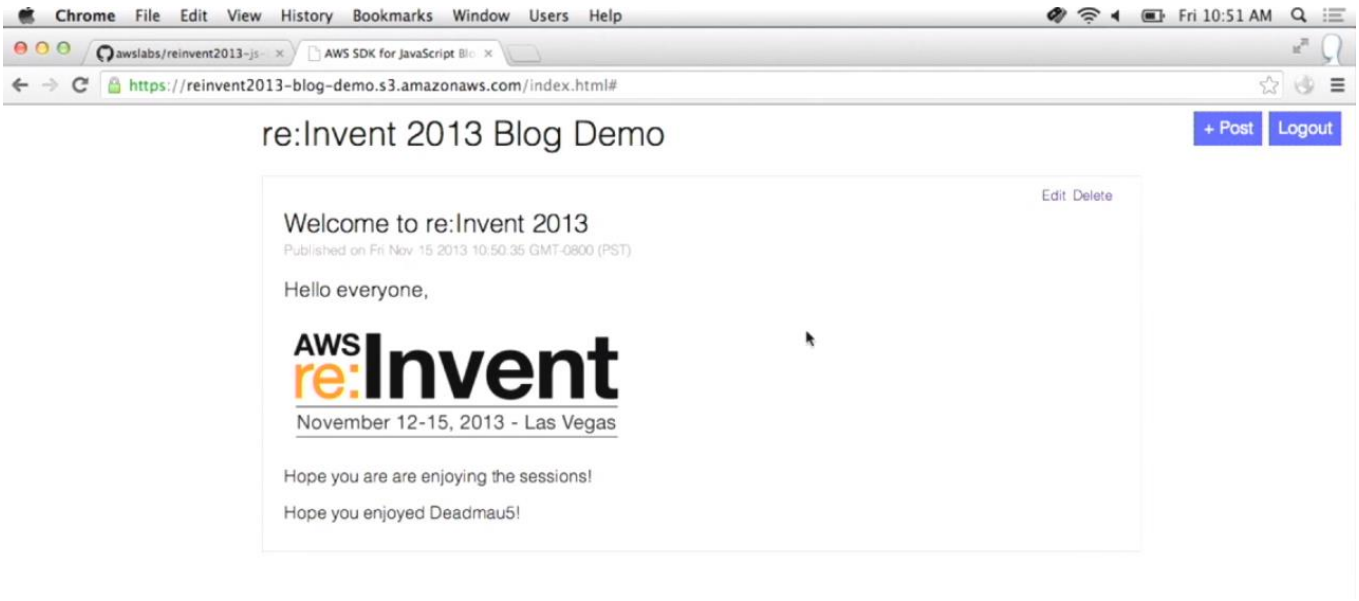
Elements Resources Network Sources Timeline Profiles Audits Console
Logged into application as administrator
app.js:235
[AWS s3 200 0.581s 0 retries] putObject({ Key: 'assets/9a9179b1-d6ca-1808-a6f0-33f3d78e15ea.png',
ContentType: 'image/png',
Body:
{ webkitRelativePath: '',
lastModifiedDate: Thu Nov 14 2013 17:38:00 GMT-0800 (PST),
name: 'logo-dark.png',
type: 'image/png',
size: 8504 },
ACL: 'public-read',
Bucket: 'reinvent2013-blog-demo' })
aws-sdk-2.0.0-rc1.min.js:7
<top frame>
<page context>
All
Errors
Warnings
Logs


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:50 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title:
Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

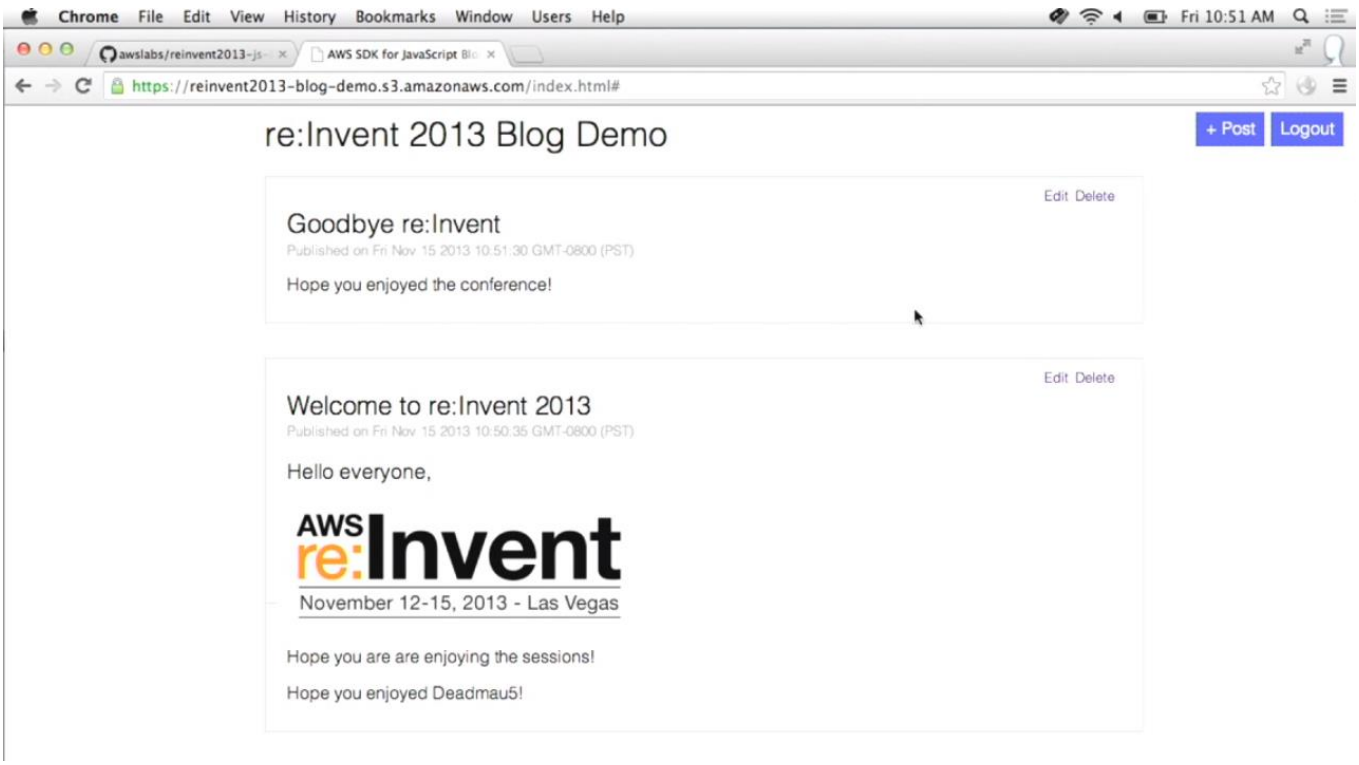
Hello everyone!
AWS re:Invent
November 12-15, 2013 - Las Vegas


Chrome File Edit View History Bookmarks Window Users Help
Fri 10:50 AM
awslabs/reinvent2013-js
AWS SDK for JavaScript Blo
https://reinvent2013-blog-demo.s3.amazonaws.com/index.html#
re:Invent 2013 Blog Demo
+ Post
Logout
Create a New Article
Title:
Welcome to re:Invent 2013
Cancel
Publish
Hello everyone!

Hope you are are enjoying the sessions!
Hello everyone!
AWS re:Invent
November 12-15, 2013 - Las Vegas



We can also add more posts



Key Differences

Three-Tier to Two-Tier

Browser Security

CORS in the browser
Credentials on device

Cross-Origin Resource Sharing

CORS	Browser sends pre-flight request to external host. Host acknowledges browser. Browser sends XHR request.
CORS + S3	CORS needs special configuration on Amazon S3. Configure CORS with bucket policy.

Configuring CORS on Amazon S3

Getting Credentials Onto Your Device

Getting Credentials Onto Your Device

Never hardcode credentials

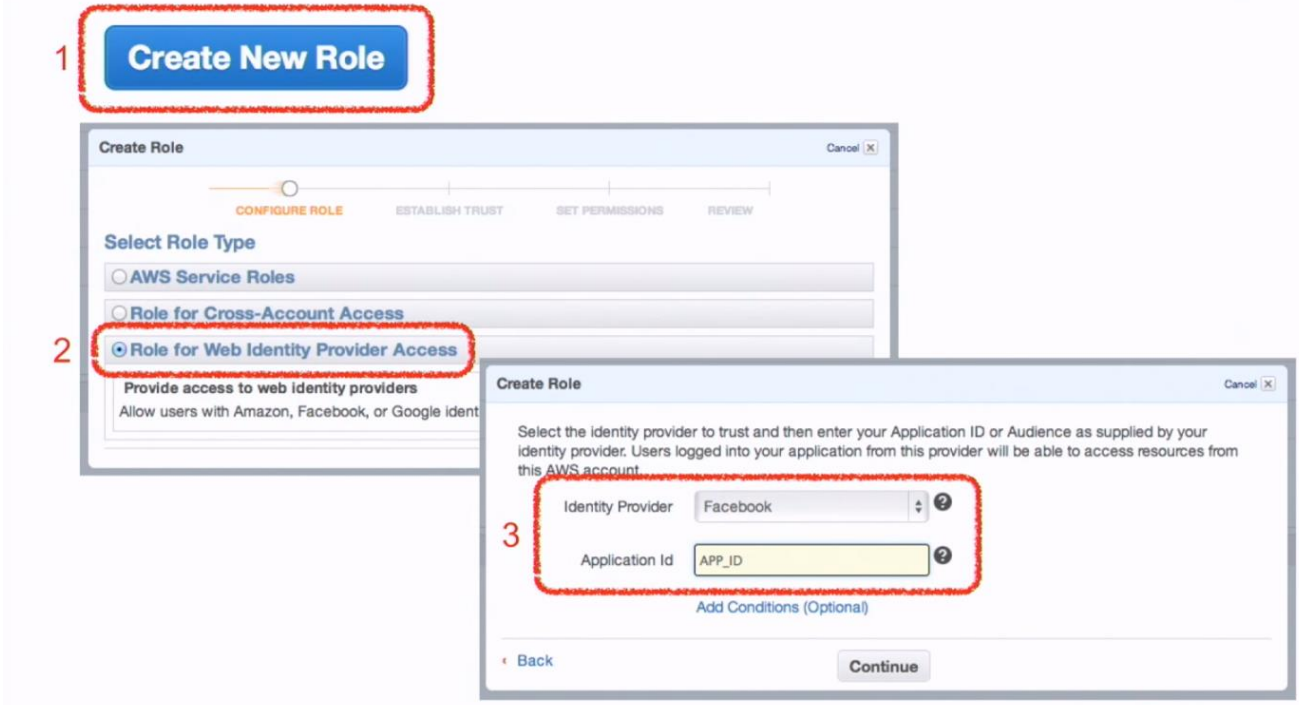
Use Web Identity Federation

Web Identity Federation

Use Facebook, Google, or Login with Amazon as third-party identity providers

Web Identity Federation

Set up IAM roles for these identity providers



Web Identity Federation

Set up permissions for IAM role

AWS.WebIdentityCredentials

JS

```
AWS.config.credentials = new AWS.WebIdentityCredentials({
  RoleArn: 'arn:aws:iam::<ACCOUNT_ID>:role/<ROLE_NAME>',
  ProviderId: 'graph.facebook.com',
  WebIdentityToken: fbAccessToken
});
```

Get a Facebook Access Token

JS

```
// 1. Load the FB JS SDK
// 2. Call FB.login()
FB.login(function (response) {
  if (response.authResponse) {
    fbAccessToken = response.authResponse.accessToken;
    AWS.config.credentials = new AWS.WebIdentityCredentials({...});
});
```

Same Concept

For other identity providers

Our Community

We ♥ Open Source

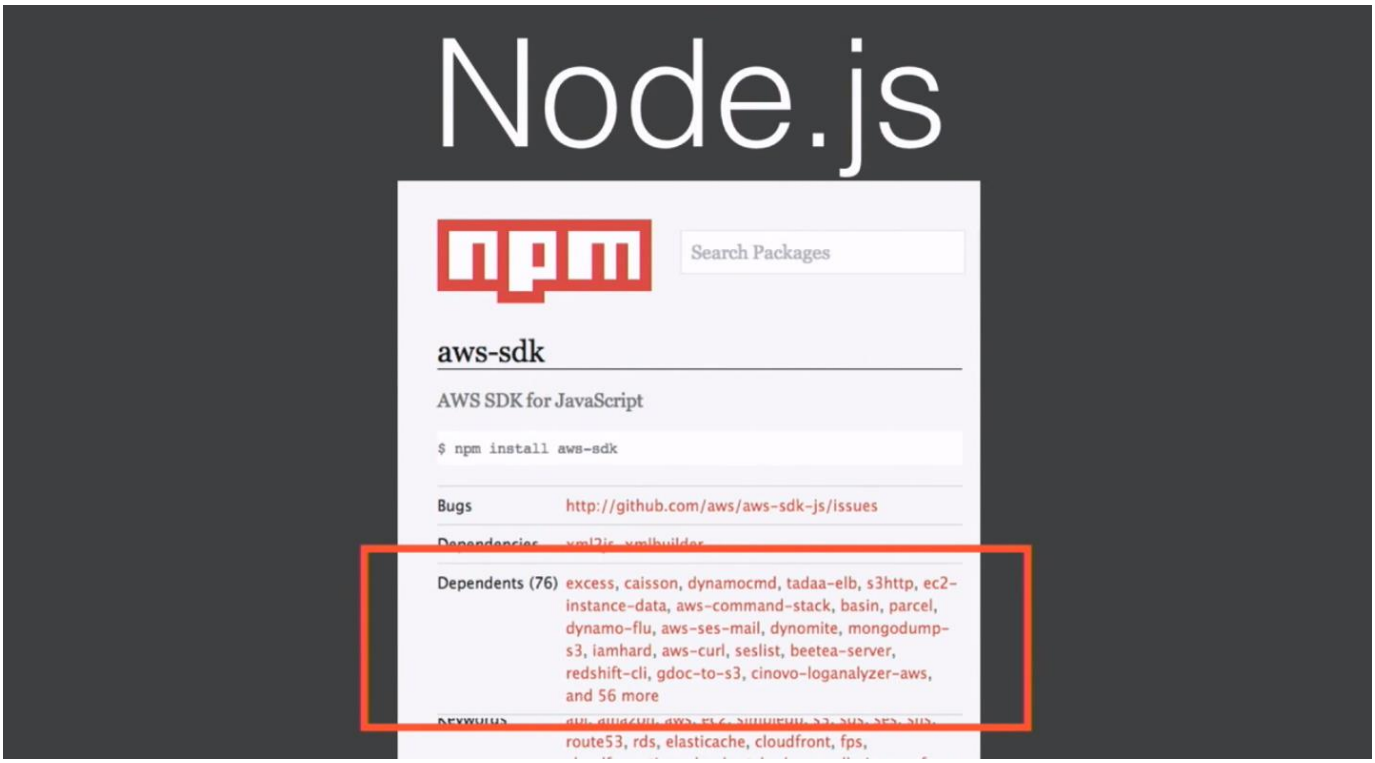
https://github.com/
aws/aws-sdk-js

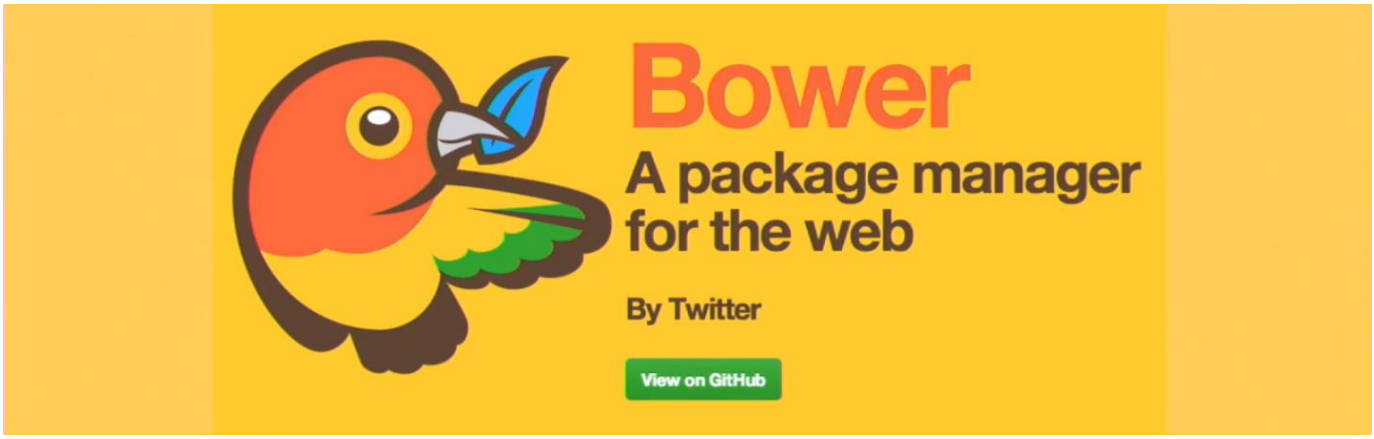
Contributing to the SDK

- Improve Documentation
- Report Issues
- Submit Pull Requests
- Third-Party Plugins

Third Party Plugins

A great way to add features


Node.js
npm
Search Packages
aws-sdk
AWS SDK for JavaScript
$ npm install aws-sdk
Bugs
http://github.com/aws/aws-sdk-js/issues
Dependencies
Dependents (76)
excess, caisson, dynamocmd, tadaa-elb, s3http, ec2-instance-data, aws-command-stack, basin, parcel, dynamo-flu, aws-ses-mail, dynomite, mongodump-s3, iamhard, aws-curl, seslist, beetea-server, redshift-cli, gdoc-to-s3, cinovo-loganalyzer-aws, and 56 more
route53, rds, elasticache, cloudfront, fps,


Bower
A package manager for the web
By Twitter
View on GitHub