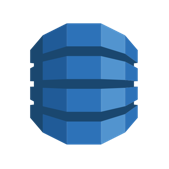
Serverless Development 101

**Module 05B – The Web Application using Serverless Computing- Search Function**

2/17/2019 Developed by Kevin Wang

Revised by Sam Chung

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Learning Outcomes**

* Learn how to scan a DynamoDB table.
* Learn how to make an API call from the front-end.
* Learn how to use the jQuery ui library.
* Learn how to deploy a fetching API with Lambda.

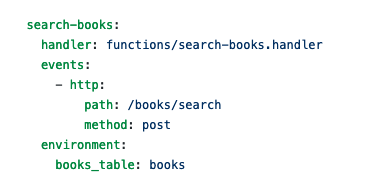
1. Open the VSCode and open the “**myproject**” project folder that we developed a web application using serverless computing in the previous module.

**Configuration – A New Function “search-books”**

1. Open the “**serverless.yml**” file under the root folder and check the difference with <http://bit.ly/2LaRknu>. **Do not copy and paste it!**

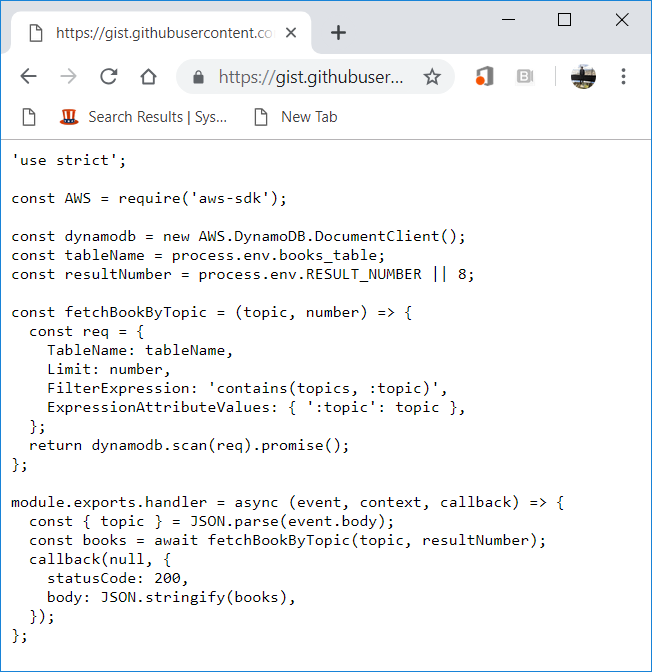
Note: You cannot simply copy and paste the content to your “serverless.yml” anymore from now since everybody added a different “fetch\_books\_api” in the last module

The only content you should add to your “serverless.yml” is the section shows below

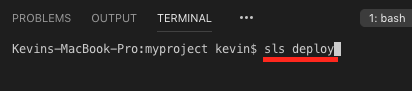


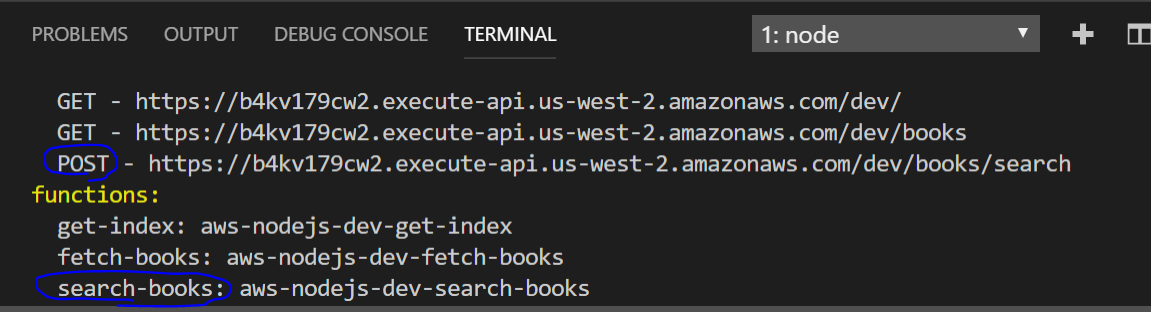
**Lambda Function – A New Function “search-books.js”**

1. Create a “**search-books.js**” file under the functions folder and copy the content from <http://bit.ly/2EcAhQH>. Save the file.



1. Open a terminal (press control and ~ keys) in VSCode and type “**sls deploy**” to deploy your new function. (**It may take a while based on your network speed**.)

****

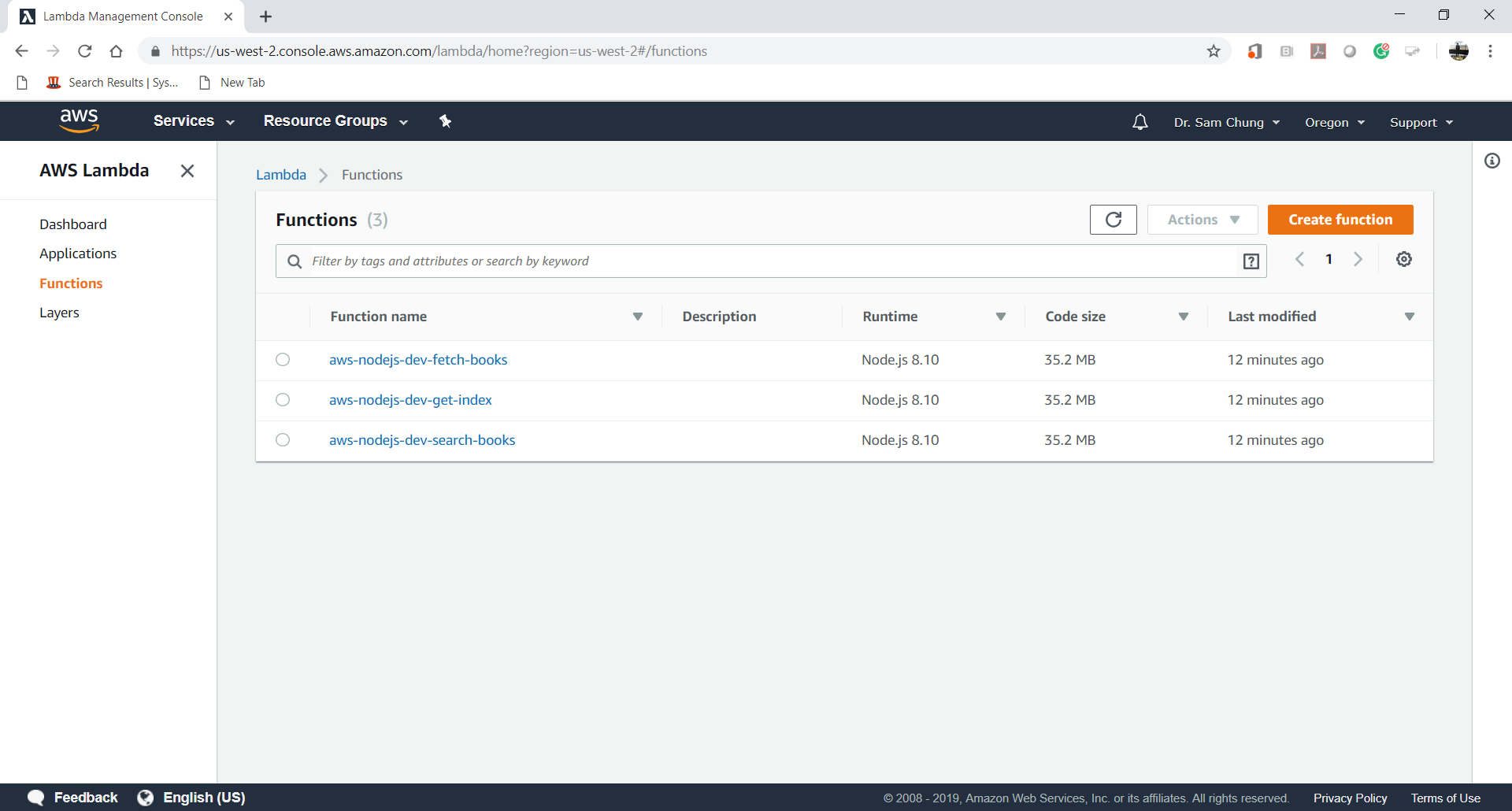
A new Lambda function was deployed.  


Note: you will have different URL endpoints.   
Record the URL for the new function (search-books).   
We will use it for the next step.

1. Paste the URL that you got from the previous step to the “serverless.yml”  
   Add “search\_books\_api:” to the correct place and paste your URL after “fetch\_books\_api:”



1. Access your “AWS Management Console” and visit your “Lambda” service.  
   Then, visit your Lambda function that you just created.

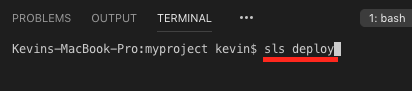


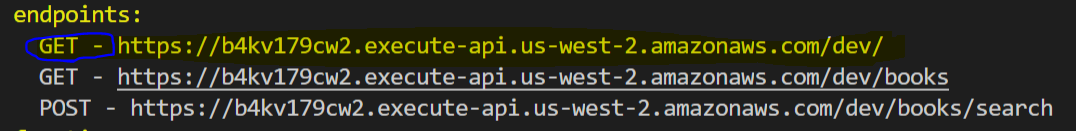
**User Interface – Update “index.html”**

1. Update the index.html with <http://bit.ly/2SKEHCz>



1. Type “**sls deploy**” in the terminal to deploy the change



1. Copy the URL of “ …/dev/” from “**endpoints**” and put it in the browser to invoke it or press “ctrl+click” to follow link.  
     
     
   Note: you will have different URL endpoints.

**Testing - Web Application**

1. Try to put keywords such as “javascript” to the search input.

|  |  |
| --- | --- |
|  |  |

1. Try to put keywords such as “data” to the search input.

|  |
| --- |
|  |

1. Try to put keywords such as “python” to the search input.

|  |
| --- |
|  |

**Questions:**

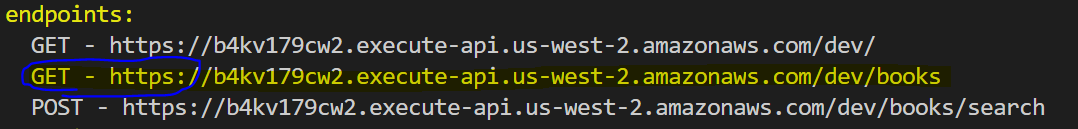
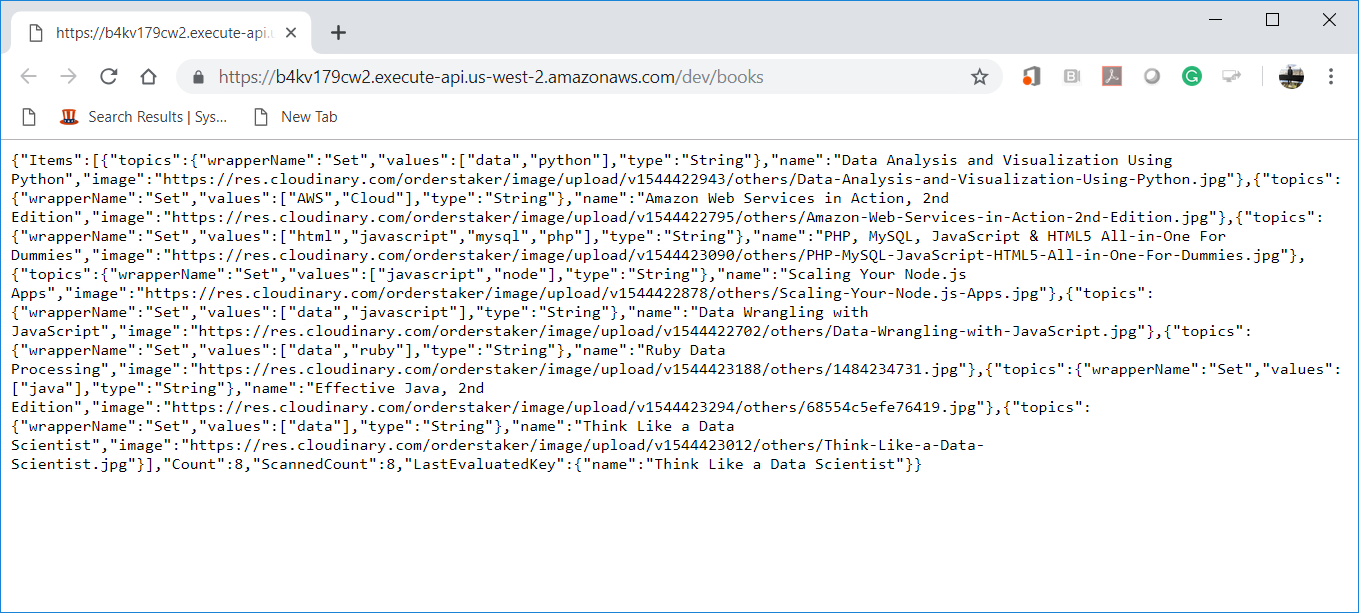
1. Explain a new function “**search-books.js**”?  
   (Answer)  
   The “**search-books.js**” filtering our data from table in AmazonDB to check if there are any book topic we input. “FilterExpression: 'contains(topics, :topic)'”
2. How could we call an API from the client side (web app’s front-end)?  
   (Answer)  
   We bind a “onclick =onSearch()” function to the html button element “id=searchButton”. Once user clicked that button, we use <https://unpkg.com/axios/dist/axios.min.js> library to post request using api url from get-index.js file

const SEARCH\_API = '{{& searchAPI}}'; // From the get-index.js

1. What can the “jQuery ui” library do?  
   (Answer)

The “jQuery ui” is collection of common javascript funtions that makes syntax is much simpler. For example, we used “const btn = $('#searchBtn');” where “$” is a jQuery syntax to get html element with “#” using to find a specific element by element Id. If we would use JavaScript, it would look like “const btn = document.getElementById("searchBtn");”

**Security Analysis**

Copy the URL of “ …/dev/books” from “**endpoints**” and give to another team. Let them invoke it by using their computer.  
  
Note: you will have different URL endpoints.   
  


1. Analyze whether the given application is secure or not in terms of CIA triad.  
   <https://geek-university.com/ccna-security/confidentiality-integrity-and-availability-cia-triad/>   
   (Answer)

Confidentiality – App doesn’t have confidential information, so I assume it is not applied here

Integrity – the data is not in the format that end user should see it. I would store this information in DB and make it only available by the request in index.html page.

Availability – I think this app doesn’t have limitation for user access so I assume it supposed to be available for everyone.

1. Analyze whether the given application is secure or not in terms of Microsoft’s STRIDE.  
   <https://cloudblogs.microsoft.com/microsoftsecure/2007/09/11/stride-chart/>  
   (Answer)

Confidentiality – the same as in #4

Integrity – the same as in #4

Availability – the same as in #4

Authentication – I would be careful to open any links with that kind of URL like “<https://wfi698quki.execute-api.us-west-2.amazonaws.com/dev>”. It could potentially be Spoofing threat

Non-repudiation – the app doesn’t have Repudiation threat

Authorization – like Availability, the app doesn’t have any restriction to access information.

1. Analyze whether the given application is secure or not in terms of OWASP’s Security by Design Principles.  
   <https://www.owasp.org/index.php/Security_by_Design_Principles#Minimize_attack_surface_area>  
   (Answer)

Based on the provided link, we can minimize the risk to perform SQL injections by creating user accounts, so only authorized users can access this app. Also, I think we could establish principle of Least privilege if we would have real books’ information accessible on the page. For example, user with privilege ONE would get access to unlimited reading books, user with privilege TWO will get 5 books a month and user privilege THREE will get only one book a month to read.

1. Analyze whether the given application is secure or not in terms of IEEE Center for Secure Design’s top 10 software security design flaws  
   <http://theinstitute.ieee.org/special-reports/special-reports/10-recommendations-for-avoiding-software-security-design-flaws>  
   (Answer)

I couldn’t open provided link, but I found information at <https://www.hack2secure.com/blogs/avoiding-top-10-software-security-design-flaws>.

Our app doesn’t have sensitive data. According to the article, policy enforcement and any sensitive data on client software often leads to compromise in the security. Any data that came from untrusted client environment should be considered as scam and must be validated. It is recommended to use authentication techniques such as “something you know, something you have and something you are” in order to re-authenticate clients.

Also, during design of security implemented software we should consider security updates in cases like thread change, environment, conditions and others.