

Team Bug Crawlers

RisenOne Employee Portal – Project 1

Team Members: Ethan Williams, Logan Johnson, Huzaifa Nadeem, Jacob Andrews, Kedar Jagtap

Problem Summary

- Create an employee portal for daily status updates
- Print data from the backend to the front end.
- Add reports to the table.
- Technologies Include: Angular Material, AWS Serverless Framework, NodeJS

Running Results

John Doe's Status Updates

Date	Projects	Report Status
4/20/2023	FLP; FEMA	Submitted
4/10/2023	N/A	Submitted
4/1/2023	FLP; FEMA	Submitted
3/1/2023	FEMA	Submitted
2/1/2023	FLP	Submitted
1/1/2023	N/A	Submitted

Filter

Add
Report

Export
Data

Items per page:

10

1 - 6 of 6

<

<

>

>

Add Report

Employee Name

John Doe

Date

4/26/2023



MM/DD/YYYY

Report Text Goes Here

Select Project(s)



FLP



FEMA



Cancel




Save

Filter Option


John Doe's Status Updates

Date	Projects	Report Status
4/20/2023	FLP; FEMA	Submitted
4/1/2023	FLP; FEMA	Submitted
3/1/2023	FEMA	Submitted





Filter 
FEMA

Add Report

Export Data

Items per page: 10 







1 – 3 of 3

Export Data Option

The screenshot shows a web application interface. At the top, a browser's address bar and tabs are visible. Below the browser, a red box highlights a 'Downloads' panel that has appeared, showing a file named 'Employee Data.csv' with an 'Open file' link and a 'See more' link. The main content area features a table titled 'John Doe's St' (partially obscured). The table has three columns: 'Date', 'Projects', and 'Report Status'. Below the table, there is a navigation bar with a 'Filter' button, an 'Add Report' button, and an 'Export Data' button, which is highlighted with a red box. To the right of these buttons are controls for 'Items per page' (set to 10) and pagination (showing '1 - 6 of 6').




Date	Projects	Report Status
4/20/2023	FLP; FEMA	Submitted
4/10/2023	N/A	Submitted
4/1/2023	FLP; FEMA	Submitted
3/1/2023	FEMA	Submitted
2/1/2023	FLP	Submitted
1/1/2023	N/A	Submitted

Filter  Add Report Export Data Items per page: 10  1 - 6 of 6    

Dynamo DB

Items:

- StatusID
- Username
- Date
- Description
- Projects (FLP/FEMA)
- Submitted (True/False)

Items returned (6)						Actions ▼	Create item
					< 1 >  		
<input type="checkbox"/>	StatusID ▼	Date ▼	Description ▼	Projects ▼			
<input type="checkbox"/>	3	4/1/2023	Status 3	3			
<input type="checkbox"/>	2	3/1/2023	Status 2	2			
<input type="checkbox"/>	4	4/10/2023	Status 4	0			
<input type="checkbox"/>	1	2/1/2023	Status 1	1			
<input type="checkbox"/>	0	1/1/2023	Status 0	0			
<input type="checkbox"/>	5	4/20/2023	Status 5	3			

Lambda Functions

serverless-dev-getTable:

- Get data of entire table.

serverless-dev-getTableItem:

- Get data of one item of the table, based on the StatusID.

serverless-dev-insertItem:

- Insert or update a new item into the table.

Lambda > Functions

Functions (3)

 *Filter by tags and attributes or search by keyword*

☐

Function name

☐

serverless-dev-getTable

☐

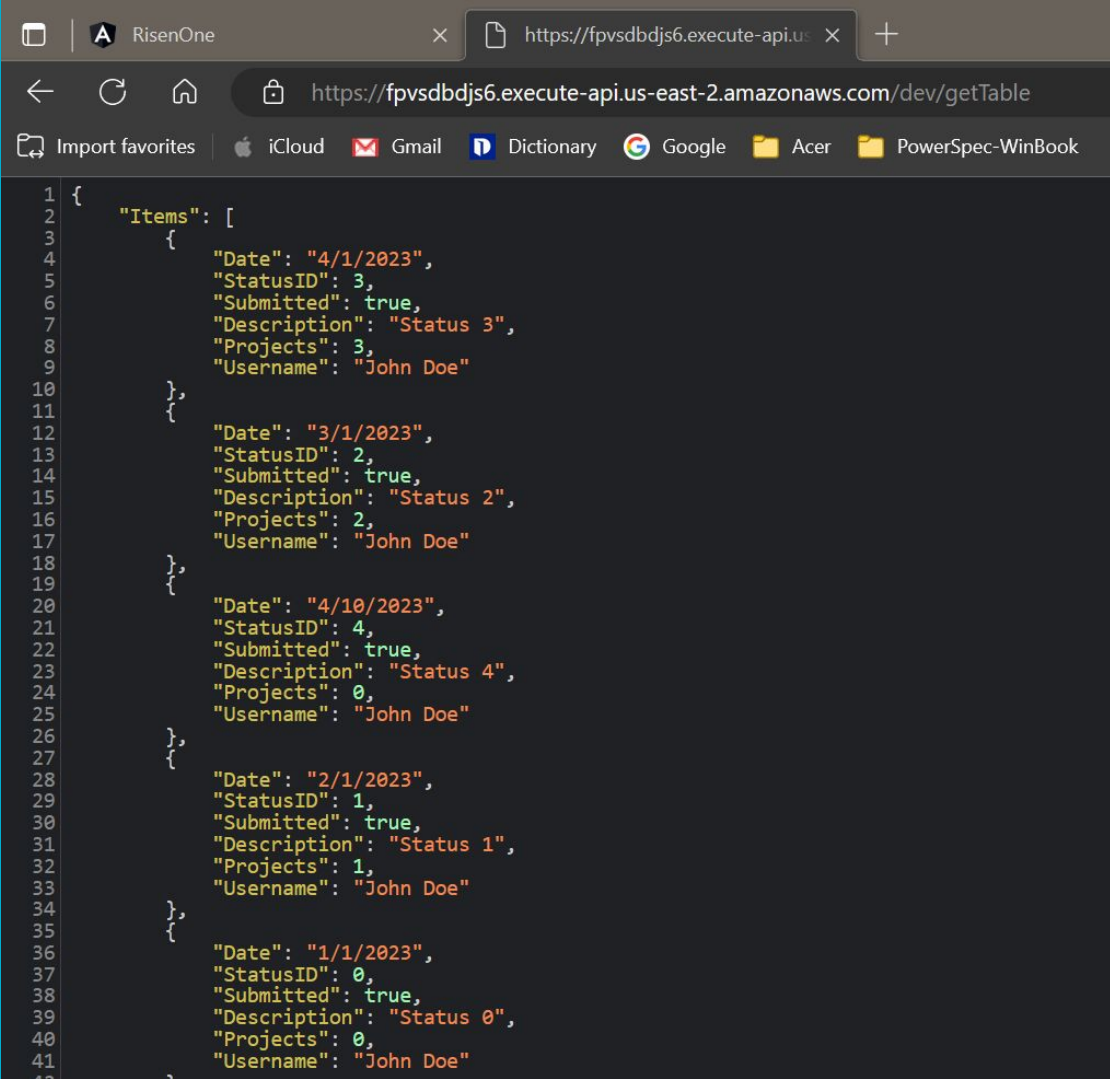
serverless-dev-getTableItem

☐

serverless-dev-insertItem

API Gateway

Calling the API will return the data as a JSON string.



The screenshot shows a web browser window with the address bar displaying `https://fpvsdbdjs6.execute-api.us-east-2.amazonaws.com/dev/getTable`. The browser's address bar also shows the domain `fpvsdbdjs6.execute-api.us-east-2.amazonaws.com`. The browser's toolbar includes icons for back, forward, home, and search, as well as a search bar. Below the toolbar, there are links to 'Import favorites', 'iCloud', 'Gmail', 'Dictionary', 'Google', 'Acer', and 'PowerSpec-WinBook'. The main content area of the browser displays a JSON string, which is a list of four objects, each representing a record with fields: Date, StatusID, Submitted, Description, Projects, and Username. The JSON is formatted with syntax highlighting and line numbers on the left side of the editor.

```
1 {  
2   "Items": [  
3     {  
4       "Date": "4/1/2023",  
5       "StatusID": 3,  
6       "Submitted": true,  
7       "Description": "Status 3",  
8       "Projects": 3,  
9       "Username": "John Doe"  
10    },  
11    {  
12      "Date": "3/1/2023",  
13      "StatusID": 2,  
14      "Submitted": true,  
15      "Description": "Status 2",  
16      "Projects": 2,  
17      "Username": "John Doe"  
18    },  
19    {  
20      "Date": "4/10/2023",  
21      "StatusID": 4,  
22      "Submitted": true,  
23      "Description": "Status 4",  
24      "Projects": 0,  
25      "Username": "John Doe"  
26    },  
27    {  
28      "Date": "2/1/2023",  
29      "StatusID": 1,  
30      "Submitted": true,  
31      "Description": "Status 1",  
32      "Projects": 1,  
33      "Username": "John Doe"  
34    },  
35    {  
36      "Date": "1/1/2023",  
37      "StatusID": 0,  
38      "Submitted": true,  
39      "Description": "Status 0",  
40      "Projects": 0,  
41      "Username": "John Doe"  
42    }  
43  ]  
44 }
```


Get Data From Backend To Frontend

```
getData() {  
  console.log("Getting Data")  
  for (let i = 0; i < this.tableData.Count; i++) {  
    let Projects = "N/A";  
    let Status = "Missing"  
    if (this.tableData.Items[i].FLP == true && this.tableData.Items[i].FEMA == true)  
      Projects = "FLP; FEMA";  
    else if (this.tableData.Items[i].FLP == true)  
      Projects = "FLP";  
    else if (this.tableData.Items[i].FEMA == true)  
      Projects = "FEMA";  
    if (this.tableData.Items[i].Submitted == true)  
      Status = "Submitted"  
    REPORT_DATA.push({date: this.tableData.Items[i].Date, projects: Projects, reportStatus: Status});  
  }  
  this.dataSource.paginator = this.paginator;  
  this.dataSource.sort = this.sort;  
}
```

What we need to finish

- Get the **Add Report** button to properly function.
- Get the **Export Data** button to export all data from the backend.