ONSJHJSA

Initial Steps:

1. Git Clone the repository. This should create a new project.
   1. Make sure to delete the old .git file
   2. Create a new repository according to the project.
   3. “git init” to initialize the empty Git Repository.
   4. “git remote add origin git\_repo\_URL” to map the local git repository with remote repository.
2. Create a database on AWS RDS.
   1. Change the configuration file and update the DB name. Location = Resources > application.properties
   2. spring.datasource.url=jdbc:sqlserver://databasesimpleinsiteaws.compoko3fbyj.us-east-2.rds.amazonaws.com:1433;database=TetonCounty
3. Run the SprinBoot project. This will create the tables mapped in the “model” within TetonCounty Database.
   1. Ensure the tables created successfully within TetonCounty Database.
4. Verify the front end using Visual Studio code
   1. Open the project/folder within VS Studio code.
   2. Open terminal. “cd” relative path of the folder that contains “package.json” folder.
   3. “npm start” cmd to start react-script / front end
   4. IMP NOTE: For the first time, it throws error – “Local package.json exists, but node\_modules missing, did you mean to install?”
   5. “npm install –g” to install the required dependencies at the global level. [ npm install to install dependency at the project level]
5. Add the users into the system.

IMP NOTES:

* In the configuration file located at Resources > application.properties , I have configured the data source as AWS RDS connection string and “TetonCounty” database.
* Now, I run the Spring boot project/service at the local machine and I can use the API end-point to sign up the user. This user will be added to the AWS RDS database. I run the project on local machine. Use below endpoint to add user.

POST API - <http://localhost:5000/api/auth/signup>

Request Payload –

{

"name": "Patrick Tierney",

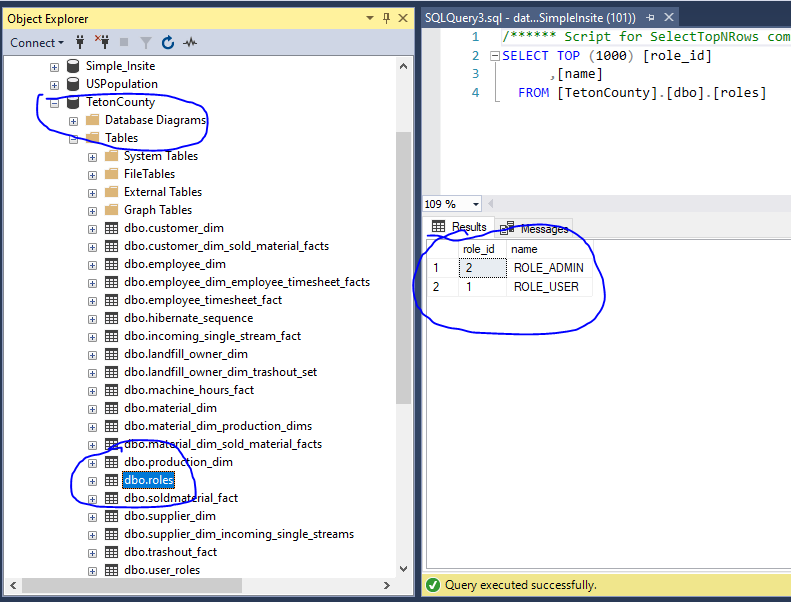
"username" : "ptierney@revolutionsystems.net",

"email": "ptierney@revolutionsystems.net",

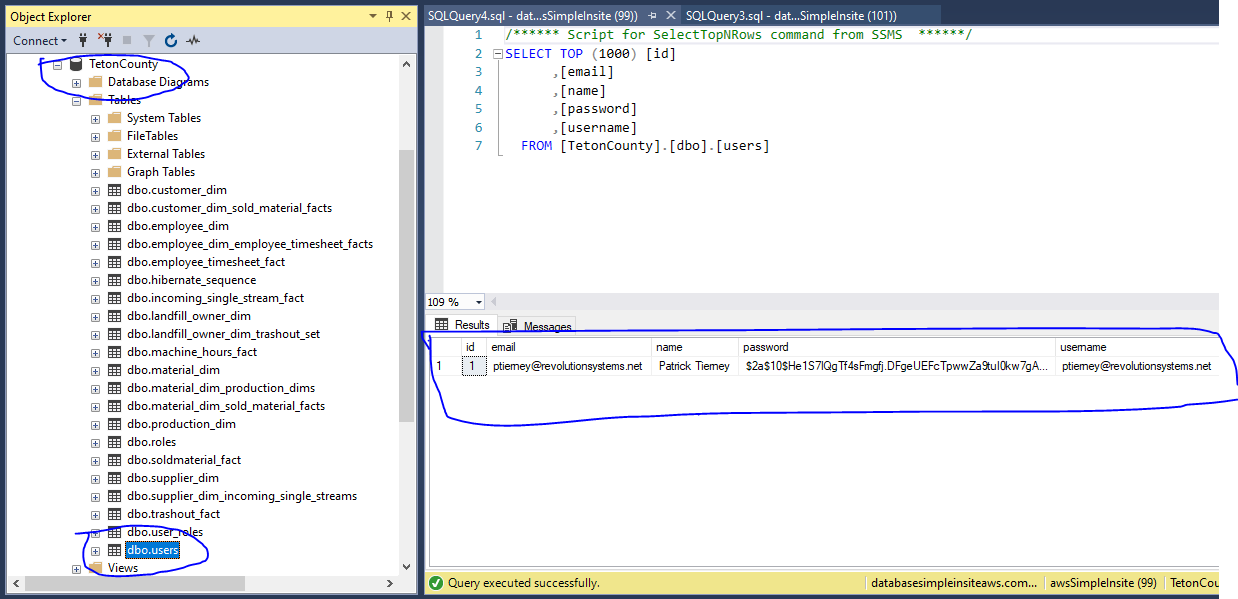
"password" : "Revolution@123"

}

* This creates an error – I need to provide the role of the user. In the code, I mentioned that the any user signed up should assign role – ROLE\_USER by default.
* So I need to add the roles in the dbo.roles table first with roles.



* Now, the POST endpoint to sign up user will work.

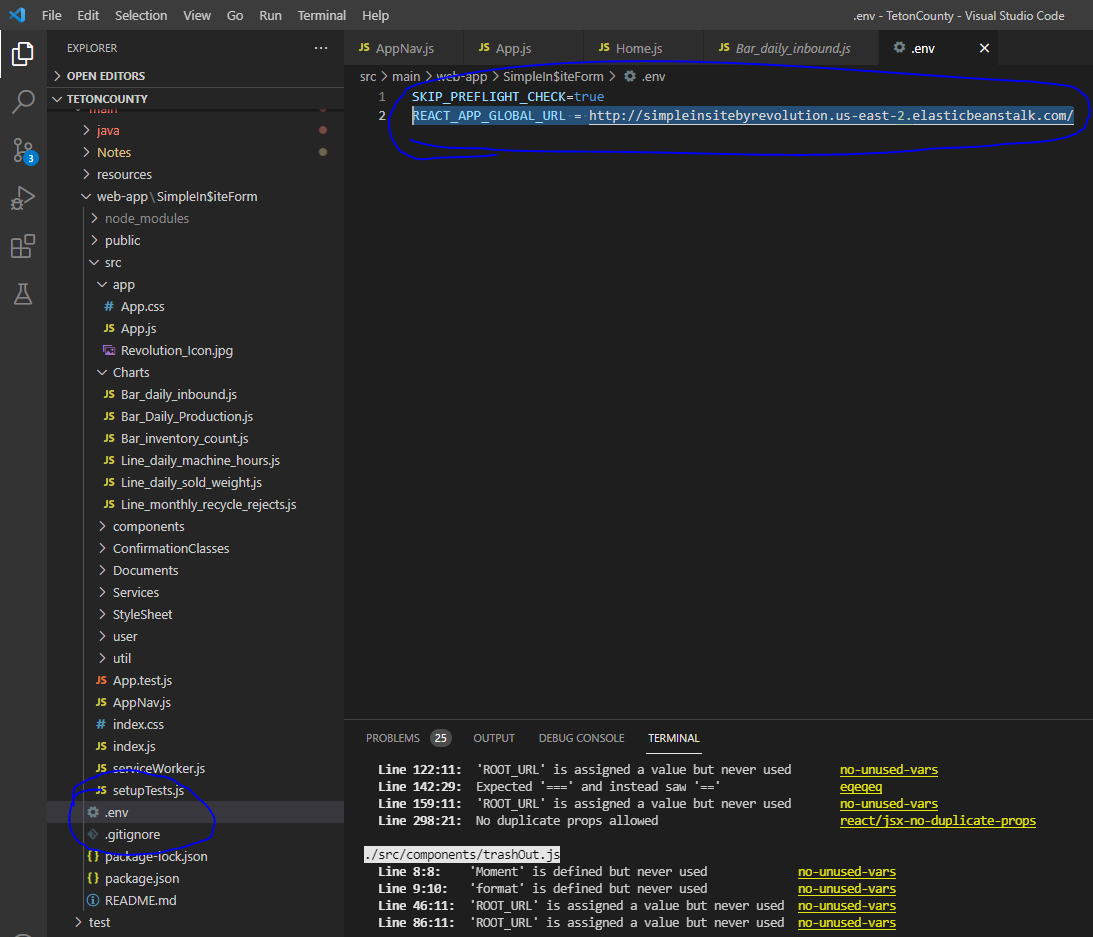


=======================================================================

Front End UI changes according to the customer:

1. Created a folder .env at the root directory. This folder works as a config folder to access the API endpoint URL. [NOTE – Defining the URL at the global level help to change the endpoint at one place and apply to everywhere. For eg, all the chart modules, reporting modules calls an API endpoint. When the code is deployed to the new AWS environment (EC2), I need to change the code at one place and all the module will reflect the correct API endpoint]

REACT\_APP\_GLOBAL\_URL = http://simpleinsitebyrevolution.us-east-2.elasticbeanstalk.com/



This var REACT\_APP\_GLOBAL\_URL can be accessed within all the modules like –

process.env.REACT\_APP\_GLOBAL\_URL + “SimpleIn$ite/dashboard/inbound/daily”

1. Set the Identity Column as Yes and Increment by 1 For all the Fact Table.
   * All the fact table’s Primary Key is set to Identity as NO. The React payload does NOT provide the primary key in the payload for the transactions. This throws the null primary key SQL error and unable to insert the record.
   * To solve this, go to Fact\_ table > Design > for the primary key > is\_Identity = Yes > Increment by 1,
2. For the DASHBOARD. I use the simple SQL Query. These queries point to Simple Insite table. Change the SQL query to call the API to appropriate table.

public interface Repo\_Production\_Material extends JpaRepository<Production\_dim,Long> {  
  
 public static final String *sql\_dash\_production\_daily* = " SELECT \*\n" +  
 "FROM [TetonCounty].[dbo].[CURRENT\_MONTH\_PRODUCTION] \n" +  
 "ORDER BY DATE ";  
  
 public static final String *sql\_dash\_production\_quarterly* = "SELECT Prod\_Month\_Name, Total\_Weight FROM [Simple\_Insite].[dbo].[QUARTER\_MONTH\_PRODUCTION]\n" +  
 "ORDER BY Prod\_Month";

* + Change the SQL Query to point to Teton County.
  + For Eg:
    - public static final String *sql\_dash\_production\_daily* = " SELECT \*\n" +  
       "FROM **[TetonCounty]**.[dbo].[CURRENT\_MONTH\_PRODUCTION] \n" +  
       "ORDER BY DATE ";

1. Stored Procedure and Views – No need to change.
   1. There are a few APIs that execute the Stored Procedure and View.
   2. As mentioned earlier, I used the RightClick on DB > Generate > Scripts > Views and SPs = To copy the SPs and Views from SimpleInsite Table.
   3. These tables does NOT mentioned the DB name OR schema. Thus, the query referes to the table within the DB and NO need to change them.
2. NPM run Build – command to package the React code.
3. S3 bucket deployment – Deploy the files under Web-app > Build > All the files.
   1. Go to S3 Bucket > Properties > Static Website > put “index.html” under index document > Put “error.html” in error document
4. To access the static website on a public domain; add the below bucket policy under permission

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowPublicReadAccess",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

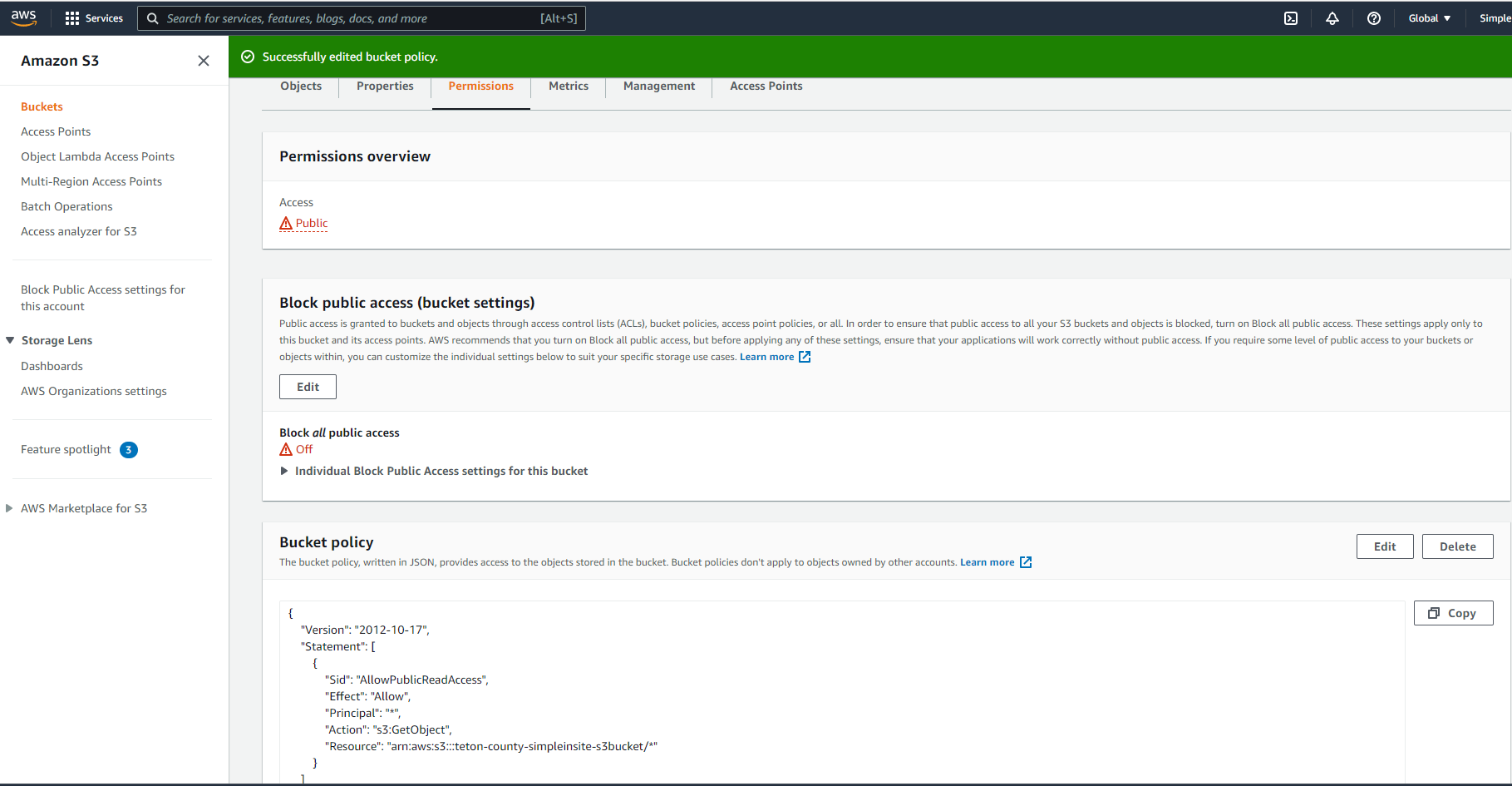
"Resource": "arn:aws:s3:::teton-county-simpleinsite-s3bucket/\*"

}

]

}

Make sure to add correct “Resource” . Copy it from top.



**ISSUE 1 – Inventory Count Graph is NOT working.**

The GET API endpoint to fetch the inventory count is “VIEW” created in SQL (VIEW is the separate table created from the actual tables). The VIEW is associated with the table so new project/database does NOT have the inventory as the view is NOT available.

public final String qry\_inventory\_count = " SELECT \*\n" +

" FROM INVENTORY\_COUNT\_V";

Error from Springboot service:

2022-01-06 22:17:43.029 DEBUG 13500 --- [nio-5000-exec-1] org.hibernate.SQL : SELECT \*

FROM INVENTORY\_COUNT\_V

Hibernate: SELECT \*

FROM INVENTORY\_COUNT\_V

2022-01-06 22:17:43.378 WARN 13500 --- [nio-5000-exec-1] o.h.engine.jdbc.spi.SqlExceptionHelper : SQL Error: 208, SQLState: S0002

2022-01-06 22:17:43.379 ERROR 13500 --- [nio-5000-exec-1] o.h.engine.jdbc.spi.SqlExceptionHelper : Invalid object name 'INVENTORY\_COUNT\_V'.

**Solution**

1. Use the simple SQL query to fetch the data.
2. Right click on master db > Tasks > Generate Scripts > Select Views and Stored Procedure 🡺 SSMS creates a script, run this script at the target server to get all the views and SPs. (Alternative way to copy Views and SPs to another DB)
   1. This option works fine. Copied all the views and SPs from one DB to multiple other DBs.

**Issue 2 – PORT number for two different EC2 instances.**

1. I deployed 2 projects on Elastic Beanstalk services,
   1. SimpleInsite [http://simpleinsite.us-east-2.elasticbeanstalk.com:5000/SimpleIn$ite/dashboard/soldweight]
   2. LakeshoreSimpleInsite [http://simpleinsitebyrevolution.us-east-2.elasticbeanstalk.com/SimpleIn$ite/dashboard/inventoryCount]

**Solution**

1. As seen from both the URL, to call an API for Simple Insite I need to mention the PORT number 5000. BUT for Lakeshore if I put the PORT number, the API does NOT work.
2. IMP – NO need to provide PORT number in the API URL while calling, the elastic beanstalk service automatically identify the PORT and provide the response. This should prevent error.