**Daily activity Day-2 (04/08/2023)**

**ANSWERS**

**1)**

1)API Referred:

i)Facebook: Graph API

ii)LinkedIn: Various APIs, such as LinkedIn Marketing Developer Platform, LinkedIn Pages API, and LinkedIn Share API.

2)Facebook Graph API: The Graph API is a RESTful API provided by Facebook that allows developers to interact with and retrieve data from the Facebook platform. It enables you to access various features like user profiles, posts, photos, pages, groups, events, and more.

LinkedIn APIs: LinkedIn provides several APIs that cater to different use cases. The LinkedIn Marketing Developer Platform is designed for advertising and allows you to manage LinkedIn ad campaigns programmatically.

3)Parameters and Endpoints:

Facebook Graph API: The Graph API has various endpoints, each serving a specific purpose. For example, to get basic information about a user, you can use the following endpoint.

Parameters can be included in the request URL or as query parameters to filter or customize the data being retrieved.

LinkedIn APIs: Each LinkedIn API has its own set of endpoints and parameters. For example, to post content on a LinkedIn page using the LinkedIn Share API, you would typically make a request to the following endpoint.

Parameters would be included in the request body, specifying the content to be shared, the visibility settings, and other relevant details.

4) Business Utilities:

Facebook Graph API: The Graph API allows businesses to integrate Facebook features into their applications and websites. This integration can be used to enable social login, retrieve user data for personalized experiences, display user posts and photos, promote content, and track social interactions.

LinkedIn APIs: The LinkedIn APIs cater to businesses looking to leverage LinkedIn's professional network. The LinkedIn Marketing Developer Platform helps businesses manage their advertising campaigns efficiently.

6)Read and explain : What is an endpoint [NOT API endpoints --> Queue endpoint + topic endpoint]

Answer:

1)Queue Endpoint:

The queue endpoint ensures that messages are processed in the order they are received, providing a first-in-first-out (FIFO) behavior. Consumers pull messages from the queue endpoint and process them at their own pace.

2)Topic Endpoint:

In contrast, a topic endpoint is associated with the concept of publish/subscribe messaging. In a topic-based messaging system, messages are published to a topic without knowing which specific consumers will receive them.

7. What is High Availability Architecture? what happens in High Availability Architecture

Answer:

In a High Availability Architecture, several key principles and practices are employed to achieve the desired level of availability:

* i)Redundancy: Redundancy involves duplicating critical components and resources of the system, such as servers, databases, networking equipment, and power supplies.
* ii)Load Balancing: Load balancing distributes incoming requests or traffic across multiple servers or resources to prevent overloading a single server and to optimize resource utilization.
* iii)Failover: Failover is the process of automatically switching to a redundant system or component when a failure is detected.
* iv)Replication: Data replication involves maintaining multiple copies of data in different locations. This ensures that data is always accessible, even if one copy becomes inaccessible due to a failure.
* v)Monitoring and Alerting: High Availability Architectures include robust monitoring and alerting systems that continuously monitor the health and performance of the components.
* vi)Scalability: HA Architecture typically involves the ability to scale the system horizontally by adding more resources or servers to accommodate increased demand or load.
* vii)Geographical Distribution: In some cases, High Availability Architectures employ geographically distributed data centers or server clusters to ensure resilience against regional failures or disasters.

8. Write a NodeJS code for :

a. Create a dummy calculator :: Addition/Subtraction/Division [check for division by 0]

b. parse a sample XML file using NodeJS and display the response in browser

Answer:

a)Node.js Dummy Calculator:

const express = require('express');

const app = express();

const port = 3000;

app.use(express.json());

// Addition

app.post('/add', (req, res) => {

const { num1, num2 } = req.body;

if (isNaN(num1) || isNaN(num2)) {

return res.status(400).json({ error: 'Invalid input. Please provide valid numbers.' });

}

const result = num1 + num2;

res.json({ result });

});

// Subtraction

app.post('/subtract', (req, res) => {

const { num1, num2 } = req.body;

if (isNaN(num1) || isNaN(num2)) {

return res.status(400).json({ error: 'Invalid input. Please provide valid numbers.' });

}

const result = num1 - num2;

res.json({ result });

});

// Division

app.post('/divide', (req, res) => {

const { num1, num2 } = req.body;

if (isNaN(num1) || isNaN(num2)) {

return res.status(400).json({ error: 'Invalid input. Please provide valid numbers.' });

}

if (num2 === 0) {

return res.status(400).json({ error: 'Cannot divide by 0.' });

}

const result = num1 / num2;

res.json({ result });

});

app.listen(port, () => {

console.log(`Calculator app listening at http://localhost:${port}`);

});

* b)Node.js XML Parsing and Display in Browser:
* const express = require('express');
* const xml2js = require('xml2js');
* const app = express();
* const port = 3001;
* const xmlData = `
* <?xml version="1.0" encoding="UTF-8"?>
* <book>
* <title>Sample Book</title>
* <author>John Doe</author>
* <published>2023-08-06</published>
* </book>
* `;
* app.get('/', (req, res) => {
* xml2js.parseString(xmlData, (err, result) => {
* if (err) {
* console.error('Error parsing XML:', err);
* return res.status(500).send('Error parsing XML.');
* }
* res.json(result);
* });
* });
* app.listen(port, () => {
* console.log(`XML parser app listening at http://localhost:${port}`);
* });
* node app.js
* Visit http://localhost:3001 in your browser, and you should see the parsed XML data displayed as JSON.

9. Create a new Solace free a/c --> Create a new EBS service :: Deploy it in AWS --> Ohio region + Custom name for Message-VPN and Cluster

Answer:

Completed