

# AWS DynamoDB Programmatic Tutorial

Narges Mehran, MSc.

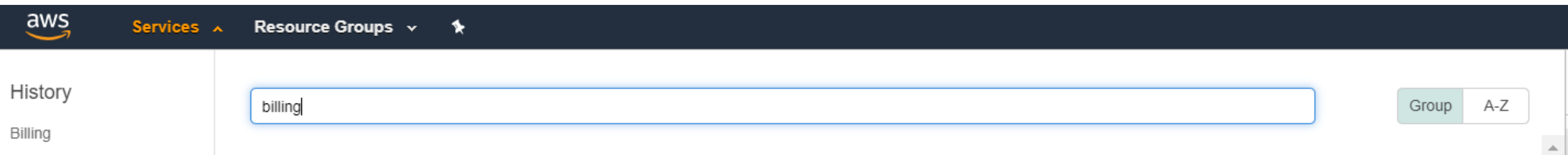
Dr. Dragi Kimovski

Current Topics in Distributed Systems: Internet of  
Things and Cloud Computing,

SS2020

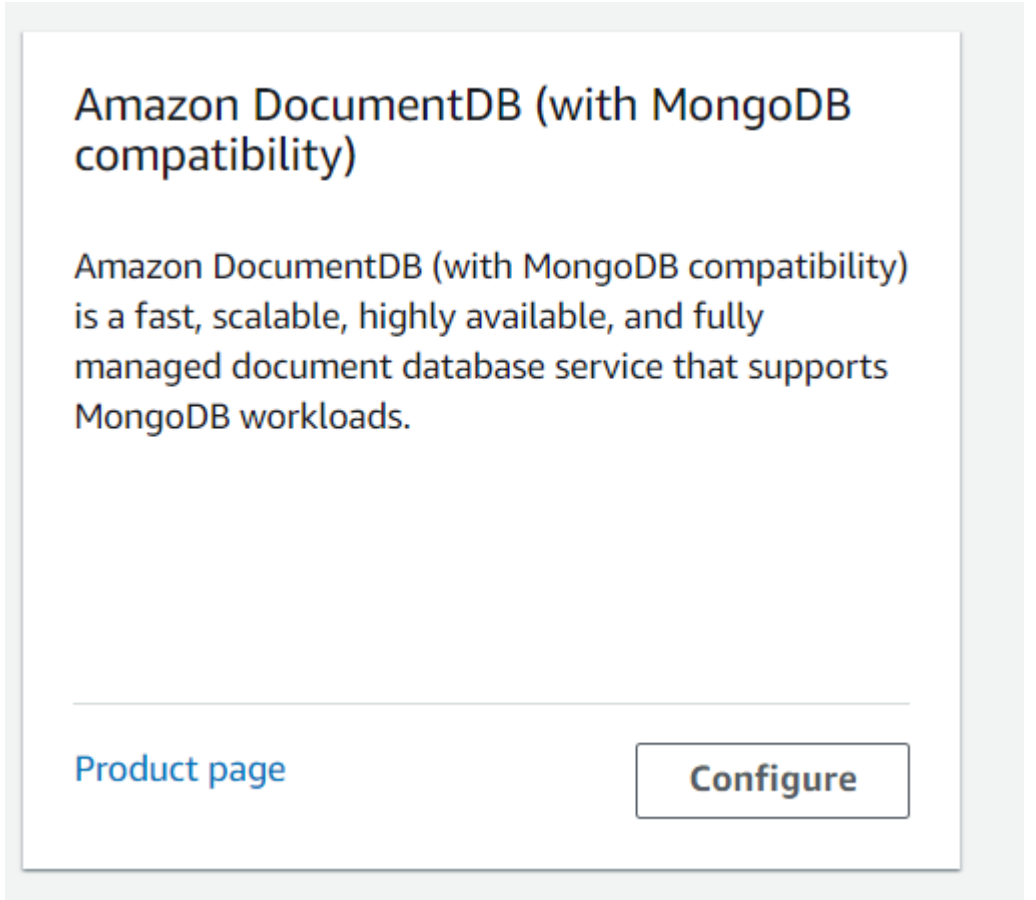
# Reminder

- In the future, please check your billing whenever you run your AWS Java code.



# AWS Pricing Calculator

- Estimate the amount of money by
  - <https://calculator.aws/#/addService>
  - A [document-oriented database](#) is designed for storing, retrieving, and managing document-oriented, or semi structured, information.
    - Document-oriented databases are one of the main categories of NoSQL databases.
  - You may select: “db r4.large” instance.



Amazon DocumentDB (with MongoDB compatibility)

Amazon DocumentDB (with MongoDB compatibility) is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads.

---

[Product page](#) [Configure](#)

# Database systems

- A **database** is an organized collection of data, generally stored and accessed electronically from a computer system
- A user's application can communicate with database to put and get data
- Data is accessed not on a file level, but e.g., on a record level
- Databases make it easy to search by an index for the data
- Customer can operate databases on Cloud instances
  - and some database services offer redundancy options
  - so the data will be replicated to a second instance to make a fault tolerance system.

# Amazon DynamoDB

- [Amazon DynamoDB](#) is a fully managed proprietary [NoSQL database](#) service that supports [key-value](#) and document data structures and is offered by [Amazon.com](#) as part of the [Amazon Web Services](#) portfolio
- DynamoDB exposes a similar data model and derives its name from [Dynamo](#), but has a different underlying implementation
- Dynamo had a multi-master design requiring the client to resolve version conflicts and DynamoDB uses synchronous replication across multiple [data centers](#) for high durability and availability
- DynamoDB was announced by Amazon CTO [Werner Vogels](#) on January 18, 2012, and is presented as an evolution of [Amazon SimpleDB](#) solution

# Amazon DynamoDB (cont.)

- Amazon DynamoDB provides low-level API actions for
  - managing database tables and indexes,
  - and for creating, reading, updating and deleting data.

[https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API\\_Operations.html](https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_Operations.html)

# Tables in DynamoDB

- Tables are the containers for all items in a DynamoDB database
- Before you can add or remove data from DynamoDB, you must create a table.
- For each table, you must define:
  - A *name* for the table that is unique for your account and region.
  - A *primary key* for which every value must be unique; no two items in your table own the same primary key value.  
A primary key can be *simple*, consisting of a single partition (HASH) key, or *composite*, consisting of a partition and a sort (RANGE) key.

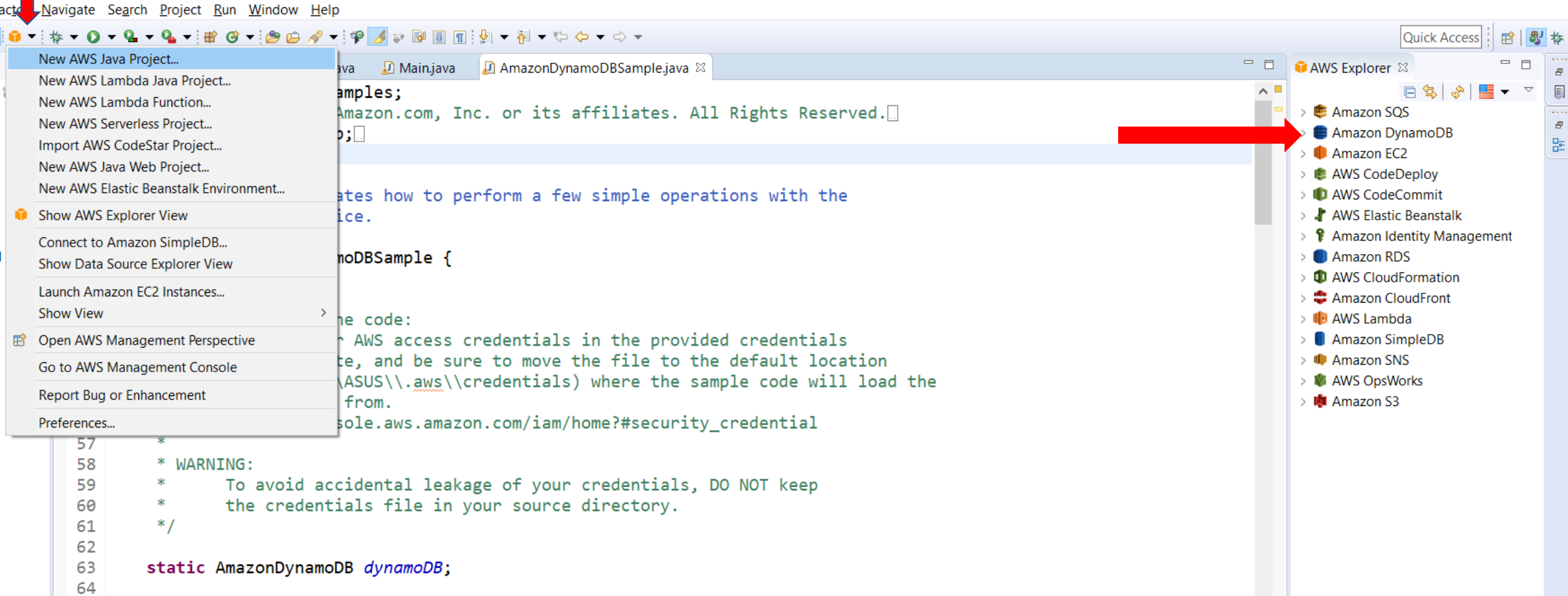
# Amazon DynamoDB (cont.)

## Amazon DynamoDB Stream:

- DynamoDB also provides API actions for accessing and processing stream records
- For example, *DescribeStream* can be called at a maximum rate of 10 times per second:
  - **Stream** has *StartingSequenceNumber*, and *EndingSequenceNumber*
  - Data stream size with the order of *KiloByte*
    - ✓ <https://aws.amazon.com/streaming-data/>



# New AWS JAVA project



The screenshot shows an IDE interface with a menu open on the left and the AWS Explorer view on the right. A red arrow points to the 'New AWS Java Project...' option in the menu. Another red arrow points to the 'Amazon DynamoDB' entry in the AWS Explorer list.

**Menu Options:**

- New AWS Java Project...
- New AWS Lambda Java Project...
- New AWS Lambda Function...
- New AWS Serverless Project...
- Import AWS CodeStar Project...
- New AWS Java Web Project...
- New AWS Elastic Beanstalk Environment...
- Show AWS Explorer View
- Connect to Amazon SimpleDB...
- Show Data Source Explorer View
- Launch Amazon EC2 Instances...
- Show View
- Open AWS Management Perspective
- Go to AWS Management Console
- Report Bug or Enhancement
- Preferences...

**AWS Explorer List:**

- Amazon SQS
- Amazon DynamoDB
- Amazon EC2
- AWS CodeDeploy
- AWS CodeCommit
- AWS Elastic Beanstalk
- Amazon Identity Management
- Amazon RDS
- AWS CloudFormation
- Amazon CloudFront
- AWS Lambda
- Amazon SimpleDB
- Amazon SNS
- AWS OpsWorks
- Amazon S3

**Code Snippets:**


```
samples;  
Amazon.com, Inc. or its affiliates. All Rights Reserved.  
;  
  
ates how to perform a few simple operations with the  
ice.  
  
noDBSample {  
  
ne code:  
r AWS access credentials in the provided credentials  
te, and be sure to move the file to the default location  
(ASUS\\.aws\\credentials) where the sample code will load the  
from.  
sole.aws.amazon.com/iam/home?#security_credential  
  
57 *  
58 * WARNING:  
59 * To avoid accidental leakage of your credentials, DO NOT keep  
60 * the credentials file in your source directory.  
61 */  
62  
63 static AmazonDynamoDB dynamoDB;  
64
```

# Creating AWS Java project and checking/configuring AWS Credential

New AWS Java Project

Create an AWS Java project

Create a new AWS Java project in the workspace

Project name:  

Maven configuration

Group ID:

Artifact ID:

Version:


Package name:

AWS Credentials

Select profile:  [Configure AWS profiles...](#)

AWS SDK for Java Samples

- ☐ Amazon DynamoDB Sample  
A sample Java program that makes requests to Amazon DynamoDB to store and query data.
- ☐ Amazon EC2 Spot Instance Advanced Sample  
Demonstrates persistent vs. one-time spot requests, launch groups, and availability groups
- ☐ Amazon EC2 Spot Instance Getting Started Sample  
Demonstrates how to set up requests for Spot Instances, how to determine when they have completed, and how to clean up afterwards.
- ☐ Amazon Kinesis Sample  
A demonstration of interacting with Amazon Kinesis using the AWS SDK for Java and the Amazon Kinesis Client library.
- ☐ Amazon S3 Sample  
A demonstration of accessing Amazon S3 buckets and objects using the AWS Java SDK.
- ☐ Amazon S3 Transfer Progress Sample  
A demonstration of tracking transfer progress for uploads to Amazon S3 using the AWS Java SDK.
- ☐ Amazon Simple Email Service JavaMail Sample  
Demonstrates how to send an email using the Amazon Simple Email Service with the AWS SDK for Java
- ☐ Amazon Simple Queue Service Sample  
A demonstration of accessing Amazon SQS queues and messages using the AWS Java SDK.
- ☐ AWS CloudFormation Sample



# Basic Amazon DynamoDB APIs

- Listing all of the tables associated with the current AWS account

```
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;  
[...]  
  
AmazonDynamoDB client = AmazonDynamoDBClientBuilder.standard().build();  
ListTablesRequest request = new ListTablesRequest(); // ?  
ListTablesResult response = client.listTables(request); // ?
```

- <https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/dynamodbv2/AmazonDynamoDB.html>

# Your Task?

- Create a Table
- Describe (get information about) a DynamoDB table
- Query a DynamoDB table
- List DynamoDB tables in a particular region
- Put an item in a DynamoDB table
  - Take the name of the table, a name (primary key value)
- Delete an item from a DynamoDB table
  - Take the table name and item (primary key: "Name") to delete.

# A sample data for your task

- A selection of the key statistics on *mental health*
- It depicts rates of mental ill health of men and women
- The statistics about homelessness and mental health

<https://www.mentalhealth.org.uk/statistics>

Thanks for your attention