**Amazon DynamoDB** is a “Key-Value” store, [NoSQL](https://en.wikipedia.org/wiki/NoSQL) [database](https://en.wikipedia.org/wiki/Database) service that is offered by [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com) as part of the [Amazon Web Services](https://en.wikipedia.org/wiki/Amazon_Web_Services) portfolio.

Throughput-based pricing.

Data is eventually consistent by default

DynamoDb design in consistency and replication is similar to Cassandra and falls into AP in CAP theorem. But Dynamo is Document oriented and Cassandra is column oriented db.

In addition to the Amazon DynamoDB web service, AWS provides a downloadable version of DynamoDB that you can run locally.

**To Download the DynamoDb🡪 click below link**

<http://docs.aws.amazon.com/amazondynamodb/latest/gettingstartedguide/GettingStarted.Download.html>

To start DynamoDb run the below command – server starts by default in 8000

java -Djava.library.path=./DynamoDBLocal\_lib -jar DynamoDBLocal.jar -sharedDb –inMemory

If you interact with AWS programmatically using the SDKs, Command Line Interface (CLI), or APIs, you must provide access keys to verify who you are and whether you have permission to access the resources you're requesting

**DynamoDb FrontEnd GUI**

Install “AWS Explorer” in Eclipse and provide “access key” and “secret key” of AWS to connect to AWS Services online.

access key : AKIAJV76KTJECBWVBG3Q

secret key : 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

To view the local database data, change the country to “Local (localhost)” inside “Amazon DynamoDb” you can see all tables.

**To write a query in “JavaScript shell” and view the result**

**Goto:** [**http://localhost:8000/shell/#**](http://localhost:8000/shell/)

**Paste the below command**

var params = {

TableName: 'Movies',

Key: {

year: 2013,

title:'Rush'

}

};

docClient.get(params, function(err, data) {

if (err) ppJson(err);

else ppJson(data);

});

**create a table**

var params = {

TableName: 'StoreSettings',

KeySchema: [ // The type of of schema. Must start with a HASH type, with an optional second RANGE.

{ // Required HASH type attribute

AttributeName: 'id',

KeyType: 'HASH',

},

{ // Required HASH type attribute

AttributeName: 'store\_id',

KeyType: 'S',

}

],

AttributeDefinitions: [ // The names and types of all primary and index key attributes only

{

AttributeName: 'id',

AttributeType: 'N' // (S | N | B) for string, number, binary

},

{

AttributeName: 'store\_id',

AttributeType: 'S' // (S | N | B) for string, number, binary

}

],

ProvisionedThroughput: { // required provisioned throughput for the table

ReadCapacityUnits: 1,

WriteCapacityUnits: 1

}

};

dynamodb.createTable(params, function(err, data) {

if (err) ppJson(err); // an error occurred

else ppJson(data); // successful response

});

**put item**

var params = {

TableName: 'StoreSettings',

Item: {

id:2,

store\_id:'20'

}};

docClient.put(params, function(err, data) {

if (err) ppJson(err); // an error occurred

else ppJson(data); // successful response

});