

Create table in dynamodb.

```
const params = {
  TableName : "book",
  AttributeDefinitions: [
    {
      AttributeName: 'title',
      AttributeType: 'S', // (S | N | B) for string, number, binary
    },
    {
      AttributeName: 'author',
      AttributeType: 'S',
    },
  ],
  KeySchema: [
    {
      AttributeName: 'title',
      KeyType: 'HASH',
    },
    {
      AttributeName: 'author',
      KeyType: 'RANGE',
    }
  ],
  ProvisionedThroughput: {
    ReadCapacityUnits: 1,
    WriteCapacityUnits:10,
  },
}

dynamodb.createTable(params,function(err,data){
  if(err){
    console.log(err)
  }else{
    console.log("table create..",data)
  }
})
```

Create Item in dynamodb

```
var params = {
  TableName: 'book',
  Item: {
    title: "Node",
    author : "vishwadatt"
  }
}

docClient.put(params, function(err, data) {
  if(err){
    console.log(err)
  }else{
    console.log("create item..".data)
  }
});
```

Get All Item in dynamodb

```
var params = {
  TableName: 'book',
```

```

    };
    docClient.scan(params, function(err, data) {
        if (err) ppJson(err); // an error occurred
        else ppJson(data); // successful response
    });

```

Get Item in dynamodb

```

var params = {
    TableName : "book",
    Key : {
        title : { "S" : "Node"},
        author : { "S" : "vishwadatt"}
    }
}

dynamodb.getItem(params,function(err,data){
    if(err){
        console.log(err)
    }else{
        console.log("data..",data)
    }
})

```

Delete Item in dynamodb

```

var params = {
    TableName : "book",
    Key : {
        title : { "S" : "Node"},
        author : { "S" : "vishwadatt"}
    }
};

dynamodb.deleteItem(params,function(err,data){
    if(err){
        console.log(err);
    }else{
        console.log("data..",data);
    }
});

```

Graphql

```

mutation{
  addBook(author : "tejas" title : "AI") {
    author
    title
  }
}

```

```

query{
  books {
    author
    title
  }
}

```

}
}