aws

Welcome

Neptune Data API

API Version 2023-08-01

Neptune Data API: Welcome

Table of Contents

Welcome

The Amazon Neptune data API provides SDK support for more than 40 of Neptune's data operations, including data loading, query execution, data inquiry, and machine learning. It supports the Gremlin and openCypher query languages, and is available in all SDK languages. It automatically signs API requests and greatly simplifies integrating Neptune into your applications.

This document was last published on July 3, 2024.

Actions

The following actions are supported:

- CancelGremlinQuery
- CancelLoaderJob
- CancelMLDataProcessingJob
- CancelMLModelTrainingJob
- CancelMLModelTransformJob
- CancelOpenCypherQuery
- CreateMLEndpoint
- DeleteMLEndpoint
- DeletePropertygraphStatistics
- DeleteSparqlStatistics
- ExecuteFastReset
- ExecuteGremlinExplainQuery
- ExecuteGremlinProfileQuery
- ExecuteGremlinQuery
- ExecuteOpenCypherExplainQuery
- ExecuteOpenCypherQuery
- GetEngineStatus
- GetGremlinQueryStatus
- GetLoaderJobStatus
- GetMLDataProcessingJob
- GetMLEndpoint
- GetMLModelTrainingJob
- GetMLModelTransformJob
- GetOpenCypherQueryStatus
- GetPropertygraphStatistics
- GetPropertygraphStream
- GetPropertygraphSummary

- GetRDFGraphSummary
- GetSparqlStatistics
- GetSparqlStream
- ListGremlinQueries
- ListLoaderJobs
- ListMLDataProcessingJobs
- ListMLEndpoints
- ListMLModelTrainingJobs
- ListMLModelTransformJobs
- ListOpenCypherQueries
- ManagePropertygraphStatistics
- ManageSparqlStatistics
- StartLoaderJob
- StartMLDataProcessingJob
- StartMLModelTrainingJob
- StartMLModelTransformJob

CancelGremlinQuery

Cancels a Gremlin query. See Gremlin query cancellation for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelQuery IAM action in that cluster.

Request Syntax

```
DELETE /gremlin/status/queryId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

queryId

The unique identifier that identifies the query to be canceled.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

The status of the cancelation

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CancelLoaderJob

Cancels a specified load job. This is an HTTP DELETE request. See Neptune Loader Get-Status API for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelLoaderJob IAM action in that cluster..

Request Syntax

```
DELETE /loader/loadId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

loadId

The ID of the load job to be deleted.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

The cancellation status.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

BulkLoadIdNotFoundException

Raised when a specified bulk-load job ID cannot be found.

HTTP Status Code: 404

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InternalFailureException

Raised when the processing of the request failed unexpectedly.

HTTP Status Code: 500

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

LoadUrlAccessDeniedException

Raised when access is denied to a specified load URL.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CancelMLDataProcessingJob

Cancels a Neptune ML data processing job. See The `dataprocessing` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelMLDataProcessingJob IAM action in that cluster.

Request Syntax

```
DELETE /ml/dataprocessing/id?clean=clean&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

clean

If set to TRUE, this flag specifies that all Neptune ML S3 artifacts should be deleted when the job is stopped. The default is FALSE.

id

The unique identifier of the data-processing job.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

The status of the cancellation request.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CancelMLModelTrainingJob

Cancels a Neptune ML model training job. See Model training using the `modeltraining` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelMLModelTrainingJob IAM action in that cluster.

Request Syntax

```
DELETE /ml/modeltraining/id?clean=clean&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

clean

If set to TRUE, this flag specifies that all Amazon S3 artifacts should be deleted when the job is stopped. The default is FALSE.

id

The unique identifier of the model-training job to be canceled.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

The status of the cancellation.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CancelMLModelTransformJob

Cancels a specified model transform job. See Use a trained model to generate new model artifacts.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelMLModelTransformJob IAM action in that cluster.

Request Syntax

```
DELETE /ml/modeltransform/id?clean=clean&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

clean

If this flag is set to TRUE, all Neptune ML S3 artifacts should be deleted when the job is stopped. The default is FALSE.

id

The unique ID of the model transform job to be canceled.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

the status of the cancelation.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CancelOpenCypherQuery

Cancels a specified openCypher query. See Neptune openCypher status endpoint for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CancelQuery IAM action in that cluster.

Request Syntax

```
DELETE /opencypher/status/queryId?silent=silent HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

queryId

The unique ID of the openCypher query to cancel.

Required: Yes

silent

If set to TRUE, causes the cancelation of the openCypher query to happen silently.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": boolean,
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

The cancelation payload for the openCypher query.

Type: Boolean

status

The cancellation status of the openCypher query.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidNumericDataException

Raised when invalid numerical data is encountered when servicing a request.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateMLEndpoint

Creates a new Neptune ML inference endpoint that lets you query one specific model that the model-training process constructed. See Managing inference endpoints using the endpoints command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:CreateMLEndpoint IAM action in that cluster.

Request Syntax

```
POST /ml/endpoints HTTP/1.1
Content-type: application/json

{
   "id": "string",
   "instanceCount": number,
   "instanceType": "string",
   "mlModelTrainingJobId": "string",
   "mlModelTransformJobId": "string",
   "modelName": "string",
   "neptuneIamRoleArn": "string",
   "update": boolean,
   "volumeEncryptionKMSKey": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

id

A unique identifier for the new inference endpoint. The default is an autogenerated timestamped name.

Type: String

Required: No

instanceCount

The minimum number of Amazon EC2 instances to deploy to an endpoint for prediction. The default is 1

Type: Integer

Required: No

instanceType

The type of Neptune ML instance to use for online servicing. The default is `ml.m5.xlarge`. Choosing the ML instance for an inference endpoint depends on the task type, the graph size, and your budget.

Type: String

Required: No

mlModelTrainingJobId

The job Id of the completed model-training job that has created the model that the inference endpoint will point to. You must supply either the `mlModelTrainingJobId` or the `mlModelTransformJobId`.

Type: String

Required: No

mlModelTransformJobId

The job Id of the completed model-transform job. You must supply either the `mlModelTrainingJobId` or the `mlModelTransformJobId`.

Type: String

Required: No

modelName

Model type for training. By default the Neptune ML model is automatically based on the `modelType` used in data processing, but you can specify a different model type here. The default is `rgcn` for heterogeneous graphs and `kge` for knowledge graphs. The only valid

value for heterogeneous graphs is `rgcn`. Valid values for knowledge graphs are: `kge`, `transe`, `distmult`, and `rotate`.

Type: String

Required: No

neptuneIamRoleArn

The ARN of an IAM role providing Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will be thrown.

Type: String

Required: No

update

If set to `true`, `update` indicates that this is an update request. The default is `false`. You must supply either the `mlModelTrainingJobId` or the `mlModelTransformJobId`.

Type: Boolean

Required: No

volumeEncryptionKMSKey

The Amazon Key Management Service (Amazon KMS) key that SageMaker uses to encrypt data on the storage volume attached to the ML compute instances that run the training job. The default is None.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "creationTimeInMillis": number,
   "id": "string"
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The ARN for the new inference endpoint.

Type: String

creationTimeInMillis

The endpoint creation time, in milliseconds.

Type: Long

id

The unique ID of the new inference endpoint.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteMLEndpoint

Cancels the creation of a Neptune ML inference endpoint. See Managing inference endpoints using the endpoints command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:DeleteMLEndpoint IAM action in that cluster.

Request Syntax

```
DELETE /ml/endpoints/id?clean=clean&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

clean

If this flag is set to TRUE, all Neptune ML S3 artifacts should be deleted when the job is stopped. The default is FALSE.

id

The unique identifier of the inference endpoint.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role providing Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will be thrown.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

status

The status of the cancellation.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeletePropertygraphStatistics

Deletes statistics for Gremlin and openCypher (property graph) data.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:DeleteStatistics IAM action in that cluster.

Request Syntax

```
DELETE /propertygraph/statistics HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 statusCode
Content-type: application/json

{
   "payload": {
      "active": boolean,
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back the following HTTP response.

statusCode

The HTTP response code: 200 if the delete was successful, or 204 if there were no statistics to delete.

The following data is returned in JSON format by the service.

payload

The deletion payload.

Type: DeleteStatisticsValueMap object

status

The cancel status.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSparqlStatistics

Deletes SPARQL statistics

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:DeleteStatistics IAM action in that cluster.

Request Syntax

```
DELETE /sparql/statistics HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 statusCode
Content-type: application/json

{
   "payload": {
      "active": boolean,
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back the following HTTP response.

statusCode

The HTTP response code: 200 if the delete was successful, or 204 if there were no statistics to delete.

The following data is returned in JSON format by the service.

payload

The deletion payload.

Type: DeleteStatisticsValueMap object

status

The cancel status.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteFastReset

The fast reset REST API lets you reset a Neptune graph quicky and easily, removing all of its data.

Neptune fast reset is a two-step process. First you call `ExecuteFastReset` with `action` set to `initiateDatabaseReset`. This returns a UUID token which you then include when calling `ExecuteFastReset` again with `action` set to `performDatabaseReset`. See Empty an Amazon Neptune DB cluster using the fast reset API.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ResetDatabase IAM action in that cluster.

Request Syntax

```
POST /system HTTP/1.1
Content-type: application/json

{
   "action": "string",
   "token": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

action

The fast reset action. One of the following values:

- **`initiateDatabaseReset`** – This action generates a unique token needed to actually perform the fast reset.
- **`performDatabaseReset`** – This action uses the token generated by the `initiateDatabaseReset` action to actually perform the fast reset.

Type: String

Valid Values: `initiateDatabaseReset | performDatabaseReset`

Required: Yes

token

The fast-reset token to initiate the reset.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": {
      "token": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

The `payload` is only returned by the `initiateDatabaseReset` action, and contains the unique token to use with the `performDatabaseReset` action to make the reset occur.

Type: FastResetToken object

status

The `status` is only returned for the `performDatabaseReset` action, and indicates whether or not the fast reset rquest is accepted.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MethodNotAllowedException

Raised when the HTTP method used by a request is not supported by the endpoint being used.

HTTP Status Code: 405

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

ServerShutdownException

Raised when the server shuts down while processing a request.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteGremlinExplainQuery

Executes a Gremlin Explain query.

Amazon Neptune has added a Gremlin feature named `explain` that provides is a self-service tool for understanding the execution approach being taken by the Neptune engine for the query. You invoke it by adding an `explain` parameter to an HTTP call that submits a Gremlin query.

The explain feature provides information about the logical structure of query execution plans. You can use this information to identify potential evaluation and execution bottlenecks and to tune your query, as explained in Tuning Gremlin queries. You can also use query hints to improve query execution plans.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows one of the following IAM actions in that cluster, depending on the query:

- neptune-db:ReadDataViaQuery
- neptune-db:WriteDataViaQuery
- neptune-db:DeleteDataViaQuery

Note that the neptune-db:QueryLanguage:Gremlin IAM condition key can be used in the policy document to restrict the use of Gremlin queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
POST /gremlin/explain HTTP/1.1
Content-type: application/json

{
   "gremlin": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

gremlin

The Gremlin explain query string.

Type: String

Required: Yes

Response Syntax

```
HTTP/1.1 200

output
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following as the HTTP body.

output

A text blob containing the Gremlin explain result, as described in Tuning Gremlin queries.

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

CancelledByUserException

Raised when a user cancelled a request.

HTTP Status Code: 500

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MalformedQueryException

Raised when a query is submitted that is syntactically incorrect or does not pass additional validation.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

QueryLimitExceededException

Raised when the number of active queries exceeds what the server can process. The query in question can be retried when the system is less busy.

HTTP Status Code: 500

QueryLimitException

Raised when the size of a query exceeds the system limit.

HTTP Status Code: 400

QueryTooLargeException

Raised when the body of a query is too large.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteGremlinProfileQuery

Executes a Gremlin Profile query, which runs a specified traversal, collects various metrics about the run, and produces a profile report as output. See Gremlin profile API in Neptune for details.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ReadDataViaQuery IAM action in that cluster.

Note that the neptune-db:QueryLanguage:Gremlin IAM condition key can be used in the policy document to restrict the use of Gremlin queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
POST /gremlin/profile HTTP/1.1
Content-type: application/json

{
   "profile.chop": number,
   "gremlin": "string",
   "profile.indexOps": boolean,
   "profile.results": boolean,
   "profile.serializer": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

profile.chop

If non-zero, causes the results string to be truncated at that number of characters. If set to zero, the string contains all the results.

Type: Integer

Required: No

gremlin

The Gremlin query string to profile.

Type: String

Required: Yes

profile.indexOps

If this flag is set to TRUE, the results include a detailed report of all index operations that took place during query execution and serialization.

Type: Boolean

Required: No

profile.results

If this flag is set to TRUE, the query results are gathered and displayed as part of the profile report. If FALSE, only the result count is displayed.

Type: Boolean

Required: No

profile.serializer

If non-null, the gathered results are returned in a serialized response message in the format specified by this parameter. See Gremlin profile API in Neptune for more information.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200

output
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following as the HTTP body.

output

A text blob containing the Gremlin Profile result. See Gremlin profile API in Neptune for details.

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

CancelledByUserException

Raised when a user cancelled a request.

HTTP Status Code: 500

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MalformedQueryException

Raised when a query is submitted that is syntactically incorrect or does not pass additional validation.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

QueryLimitExceededException

Raised when the number of active queries exceeds what the server can process. The query in question can be retried when the system is less busy.

HTTP Status Code: 500

QueryLimitException

Raised when the size of a query exceeds the system limit.

HTTP Status Code: 400

QueryTooLargeException

Raised when the body of a query is too large.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteGremlinQuery

This commands executes a Gremlin query. Amazon Neptune is compatible with Apache TinkerPop3 and Gremlin, so you can use the Gremlin traversal language to query the graph, as described under The Graph in the Apache TinkerPop3 documentation. More details can also be found in Accessing a Neptune graph with Gremlin.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that enables one of the following IAM actions in that cluster, depending on the query:

- neptune-db:ReadDataViaQuery
- neptune-db:WriteDataViaQuery
- neptune-db:DeleteDataViaQuery

Note that the neptune-db:QueryLanguage:Gremlin IAM condition key can be used in the policy document to restrict the use of Gremlin queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
POST /gremlin HTTP/1.1
accept: serializer
Content-type: application/json

{
   "gremlin": "string"
}
```

URI Request Parameters

The request uses the following URI parameters.

serializer

If non-null, the query results are returned in a serialized response message in the format specified by this parameter. See the GraphSON section in the TinkerPop documentation for a list of the formats that are currently supported.

Request Body

The request accepts the following data in JSON format.

gremlin

Using this API, you can run Gremlin queries in string format much as you can using the HTTP endpoint. The interface is compatible with whatever Gremlin version your DB cluster is using (see the Tinkerpop client section to determine which Gremlin releases your engine version supports).

Type: String

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "meta": JSON value,
   "requestId": "string",
   "result": JSON value,
   "status": {
      "attributes": JSON value,
      "code": number,
      "message": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

meta

Metadata about the Gremlin query.

Type: JSON value

requestId

The unique identifier of the Gremlin query.

Type: String

result

The Gremlin query output from the server.

Type: JSON value

status

The status of the Gremlin query.

Type: GremlinQueryStatusAttributes object

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

CancelledByUserException

Raised when a user cancelled a request.

HTTP Status Code: 500

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MalformedQueryException

Raised when a query is submitted that is syntactically incorrect or does not pass additional validation.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

QueryLimitExceededException

Raised when the number of active queries exceeds what the server can process. The query in question can be retried when the system is less busy.

HTTP Status Code: 500

QueryLimitException

Raised when the size of a query exceeds the system limit.

HTTP Status Code: 400

QueryTooLargeException

Raised when the body of a query is too large.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteOpenCypherExplainQuery

Executes an openCypher `explain` request. See The openCypher explain feature for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ReadDataViaQuery IAM action in that cluster.

Note that the neptune-db:QueryLanguage:OpenCypher IAM condition key can be used in the policy document to restrict the use of openCypher queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
POST /opencypher/explain HTTP/1.1
Content-type: application/json

{
   "explain": "string",
   "query": "string",
   "parameters": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

explain

The openCypher `explain` mode. Can be one of: `static`, `dynamic`, or `details`.

Type: String

Valid Values: `static | dynamic | details`

Required: Yes

query

The openCypher query string.

Type: String

Required: Yes

parameters

The openCypher query parameters.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200

results
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following as the HTTP body.

results

A text blob containing the openCypher `explain` results.

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

CancelledByUserException

Raised when a user cancelled a request.

HTTP Status Code: 500

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidNumericDataException

Raised when invalid numerical data is encountered when servicing a request.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MalformedQueryException

Raised when a query is submitted that is syntactically incorrect or does not pass additional validation.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

QueryLimitExceededException

Raised when the number of active queries exceeds what the server can process. The query in question can be retried when the system is less busy.

HTTP Status Code: 500

QueryLimitException

Raised when the size of a query exceeds the system limit.

HTTP Status Code: 400

QueryTooLargeException

Raised when the body of a query is too large.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteOpenCypherQuery

Executes an openCypher query. See Accessing the Neptune Graph with openCypher for more information.

Neptune supports building graph applications using openCypher, which is currently one of the most popular query languages among developers working with graph databases. Developers, business analysts, and data scientists like openCypher's declarative, SQL-inspired syntax because it provides a familiar structure in which to querying property graphs.

The openCypher language was originally developed by Neo4j, then open-sourced in 2015 and contributed to the openCypher project under an Apache 2 open-source license.

Note that when invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows one of the following IAM actions in that cluster, depending on the query:

- neptune-db:ReadDataViaQuery
- neptune-db:WriteDataViaQuery
- neptune-db:DeleteDataViaQuery

Note also that the neptune-db:QueryLanguage:OpenCypher IAM condition key can be used in the policy document to restrict the use of openCypher queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
POST /opencypher HTTP/1.1
Content-type: application/json

{
   "query": "string",
   "parameters": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

query

The openCypher query string to be executed.

Type: String

Required: Yes

parameters

The openCypher query parameters for query execution. See Examples of openCypher parameterized queries for more information.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "results": JSON value
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

results

The openCypherquery results.

Type: JSON value

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

CancelledByUserException

Raised when a user cancelled a request.

HTTP Status Code: 500

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidNumericDataException

Raised when invalid numerical data is encountered when servicing a request.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MalformedQueryException

Raised when a query is submitted that is syntactically incorrect or does not pass additional validation.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

QueryLimitExceededException

Raised when the number of active queries exceeds what the server can process. The query in question can be retried when the system is less busy.

HTTP Status Code: 500

QueryLimitException

Raised when the size of a query exceeds the system limit.

HTTP Status Code: 400

QueryTooLargeException

Raised when the body of a query is too large.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetEngineStatus

Retrieves the status of the graph database on the host.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetEngineStatus IAM action in that cluster.

Request Syntax

```
GET /status HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "dbEngineVersion": "string",
   "dfeQueryEngine": "string",
   "features": {
      "string" : JSON value
   },
   "gremlin": {
      "version": "string"
   },
   "labMode": {
      "string" : "string"
   },
   "opencypher": {
      "version": "string"
   },
   "role": "string",
```

```
   "rollingBackTrxCount": number,
   "rollingBackTrxEarliestStartTime": "string",
   "settings": {
      "string" : "string"
   },
   "sparql": {
      "version": "string"
   },
   "startTime": "string",
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

dbEngineVersion

Set to the Neptune engine version running on your DB cluster. If this engine version has been manually patched since it was released, the version number is prefixed by `Patch-`.

Type: String

dfeQueryEngine

Set to `enabled` if the DFE engine is fully enabled, or to `viaQueryHint` (the default) if the DFE engine is only used with queries that have the `useDFE` query hint set to `true`.

Type: String

features

Contains status information about the features enabled on your DB cluster.

Type: String to JSON value map

gremlin

Contains information about the Gremlin query language available on your cluster. Specifically, it contains a version field that specifies the current TinkerPop version being used by the engine.

Type: QueryLanguageVersion object

labMode

Contains Lab Mode settings being used by the engine.

Type: String to string map

opencypher

Contains information about the openCypher query language available on your cluster. Specifically, it contains a version field that specifies the current operCypher version being used by the engine.

Type: QueryLanguageVersion object

role

Set to `reader` if the instance is a read-replica, or to `writer` if the instance is the primary instance.

Type: String

rollingBackTrxCount

If there are transactions being rolled back, this field is set to the number of such transactions. If there are none, the field doesn't appear at all.

Type: Integer

rollingBackTrxEarliestStartTime

Set to the start time of the earliest transaction being rolled back. If no transactions are being rolled back, the field doesn't appear at all.

Type: String

settings

Contains information about the current settings on your DB cluster. For example, contains the current cluster query timeout setting (`clusterQueryTimeoutInMs`).

Type: String to string map

sparql

Contains information about the SPARQL query language available on your cluster. Specifically, it contains a version field that specifies the current SPARQL version being used by the engine.

Type: QueryLanguageVersion object

startTime

Set to the UTC time at which the current server process started.

Type: String

status

Set to `healthy` if the instance is not experiencing problems. If the instance is recovering from a crash or from being rebooted and there are active transactions running from the latest server shutdown, status is set to `recovery`.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InternalFailureException

Raised when the processing of the request failed unexpectedly.

HTTP Status Code: 500

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetGremlinQueryStatus

Gets the status of a specified Gremlin query.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetQueryStatus IAM action in that cluster.

Note that the neptune-db:QueryLanguage:Gremlin IAM condition key can be used in the policy document to restrict the use of Gremlin queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /gremlin/status/queryId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

queryId

The unique identifier that identifies the Gremlin query.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "queryEvalStats": {
      "cancelled": boolean,
      "elapsed": number,
```

```
      "subqueries": JSON value,
      "waited": number
   },
   "queryId": "string",
   "queryString": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

queryEvalStats

The evaluation status of the Gremlin query.

Type: QueryEvalStats object

queryId

The ID of the query for which status is being returned.

Type: String

queryString

The Gremlin query string.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetLoaderJobStatus

Gets status information about a specified load job. Neptune keeps track of the most recent 1,024 bulk load jobs, and stores the last 10,000 error details per job.

See Neptune Loader Get-Status API for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetLoaderJobStatus IAM action in that cluster..

Request Syntax

```
GET /loader/loadId?details=details&errors=errors&errorsPerPage=errorsPerPage&page=page
 HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

details

Flag indicating whether or not to include details beyond the overall status (TRUE or FALSE; the default is FALSE).

errors

Flag indicating whether or not to include a list of errors encountered (TRUE or FALSE; the default is FALSE).

The list of errors is paged. The page and errorsPerPage parameters allow you to page through all the errors.

errorsPerPage

The number of errors returned in each page (a positive integer; the default is 10). Only valid when the errors parameter set to TRUE.

Valid Range: Minimum value of 1.

loadId

The load ID of the load job to get the status of.

Required: Yes

page

The error page number (a positive integer; the default is 1). Only valid when the `errors` parameter is set to TRUE.

Valid Range: Minimum value of 1.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": JSON value,
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

Status information about the load job, in a layout that could look like this:

Type: JSON value

status

The HTTP response code for the request.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

BulkLoadIdNotFoundException

Raised when a specified bulk-load job ID cannot be found.

HTTP Status Code: 404

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InternalFailureException

Raised when the processing of the request failed unexpectedly.

HTTP Status Code: 500

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

LoadUrlAccessDeniedException

Raised when access is denied to a specified load URL.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMLDataProcessingJob

Retrieves information about a specified data processing job. See The `dataprocessing` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:neptune-db:GetMLDataProcessingJobStatus IAM action in that cluster.

Request Syntax

```
GET /ml/dataprocessing/id?neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

id

The unique identifier of the data-processing job to be retrieved.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "id": "string",
   "processingJob": {
```

```
      "arn": "string",
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

id

The unique identifier of this data-processing job.

Type: String

processingJob

Definition of the data processing job.

Type: MlResourceDefinition object

status

Status of the data processing job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMLEndpoint

Retrieves details about an inference endpoint. See Managing inference endpoints using the endpoints command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetMLEndpointStatus IAM action in that cluster.

Request Syntax

```
GET /ml/endpoints/id?neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

id

The unique identifier of the inference endpoint.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "endpoint": {
```

```
      "arn": "string",
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "endpointConfig": {
      "arn": "string",
      "name": "string"
   },
   "id": "string",
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

endpoint

The endpoint definition.

Type: MlResourceDefinition object

endpointConfig

The endpoint configuration

Type: MlConfigDefinition object

id

The unique identifier of the inference endpoint.

Type: String

status

The status of the inference endpoint.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMLModelTrainingJob

Retrieves information about a Neptune ML model training job. See Model training using the `modeltraining` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetMLModelTrainingJobStatus IAM action in that cluster.

Request Syntax

```
GET /ml/modeltraining/id?neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

id

The unique identifier of the model-training job to retrieve.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "hpoJob": {
      "arn": "string",
```

```
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "id": "string",
   "mlModels": [
      {
         "arn": "string",
         "name": "string"
      }
   ],
   "modelTransformJob": {
      "arn": "string",
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "processingJob": {
      "arn": "string",
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

hpoJob

The HPO job.

Type: MlResourceDefinition object

id

The unique identifier of this model-training job.

Type: String

mlModels

A list of the configurations of the ML models being used.

Type: Array of MlConfigDefinition objects

modelTransformJob

The model transform job.

Type: MlResourceDefinition object

processingJob

The data processing job.

Type: MlResourceDefinition object

status

The status of the model training job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMLModelTransformJob

Gets information about a specified model transform job. See Use a trained model to generate new model artifacts.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetMLModelTransformJobStatus IAM action in that cluster.

Request Syntax

```
GET /ml/modeltransform/id?neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

id

The unique identifier of the model-transform job to be reetrieved.

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "baseProcessingJob": {
      "arn": "string",
```

```
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "id": "string",
   "models": [
      {
         "arn": "string",
         "name": "string"
      }
   ],
   "remoteModelTransformJob": {
      "arn": "string",
      "cloudwatchLogUrl": "string",
      "failureReason": "string",
      "name": "string",
      "outputLocation": "string",
      "status": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

baseProcessingJob

The base data processing job.

Type: MlResourceDefinition object

id

The unique identifier of the model-transform job to be retrieved.

Type: String

models

A list of the configuration information for the models being used.

Type: Array of MlConfigDefinition objects

remoteModelTransformJob

The remote model transform job.

Type: MlResourceDefinition object

status

The status of the model-transform job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetOpenCypherQueryStatus

Retrieves the status of a specified openCypher query.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetQueryStatus IAM action in that cluster.

Note that the neptune-db:QueryLanguage:OpenCypher IAM condition key can be used in the policy document to restrict the use of openCypher queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /opencypher/status/queryId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

queryId

The unique ID of the openCypher query for which to retrieve the query status.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "queryEvalStats": {
      "cancelled": boolean,
      "elapsed": number,
      "subqueries": JSON value,
```

```
      "waited": number
   },
   "queryId": "string",
   "queryString": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

queryEvalStats

The openCypher query evaluation status.

Type: QueryEvalStats object

queryId

The unique ID of the query for which status is being returned.

Type: String

queryString

The openCypher query string.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidNumericDataException

Raised when invalid numerical data is encountered when servicing a request.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetPropertygraphStatistics

Gets property graph statistics (Gremlin and openCypher).

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetStatisticsStatus IAM action in that cluster.

Request Syntax

```
GET /propertygraph/statistics HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": {
      "active": boolean,
      "autoCompute": boolean,
      "date": "string",
      "note": "string",
      "signatureInfo": {
         "instanceCount": number,
         "predicateCount": number,
         "signatureCount": number
      },
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

Statistics for property-graph data.

Type: Statistics object

status

The HTTP return code of the request. If the request succeeded, the code is 200. See Common error codes for DFE statistics request for a list of common errors.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetPropertygraphStream

Gets a stream for a property graph.

With the Neptune Streams feature, you can generate a complete sequence of change-log entries that record every change made to your graph data as it happens. `GetPropertygraphStream` lets you collect these change-log entries for a property graph.

The Neptune streams feature needs to be enabled on your Neptune DBcluster. To enable streams, set the neptune_streams DB cluster parameter to 1.

See Capturing graph changes in real time using Neptune streams.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetStreamRecords IAM action in that cluster.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that enables one of the following IAM actions, depending on the query:

Note that you can restrict property-graph queries using the following IAM context keys:

- neptune-db:QueryLanguage:Gremlin
- neptune-db:QueryLanguage:OpenCypher

See Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /propertygraph/stream?
commitNum=commitNum&iteratorType=iteratorType&limit=limit&opNum=opNum HTTP/1.1
Accept-Encoding: encoding
```

URI Request Parameters

The request uses the following URI parameters.

commitNum

The commit number of the starting record to read from the change-log stream. This parameter is required when `iteratorType` is`AT_SEQUENCE_NUMBER` or `AFTER_SEQUENCE_NUMBER`, and ignored when `iteratorType` is `TRIM_HORIZON` or `LATEST`.

encoding

If set to TRUE, Neptune compresses the response using gzip encoding.

Valid Values: `gzip`

iteratorType

Can be one of:

- `AT_SEQUENCE_NUMBER` – Indicates that reading should start from the event sequence number specified jointly by the `commitNum` and `opNum` parameters.
- `AFTER_SEQUENCE_NUMBER` – Indicates that reading should start right after the event sequence number specified jointly by the `commitNum` and `opNum` parameters.
- `TRIM_HORIZON` – Indicates that reading should start at the last untrimmed record in the system, which is the oldest unexpired (not yet deleted) record in the change-log stream.
- `LATEST` – Indicates that reading should start at the most recent record in the system, which is the latest unexpired (not yet deleted) record in the change-log stream.

Valid Values: `AT_SEQUENCE_NUMBER | AFTER_SEQUENCE_NUMBER | TRIM_HORIZON | LATEST`

limit

Specifies the maximum number of records to return. There is also a size limit of 10 MB on the response that can't be modified and that takes precedence over the number of records specified in the `limit` parameter. The response does include a threshold-breaching record if the 10 MB limit was reached.

The range for `limit` is 1 to 100,000, with a default of 10.

Valid Range: Minimum value of 1. Maximum value of 100000.

opNum

The operation sequence number within the specified commit to start reading from in the change-log stream data. The default is 1.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "format": "string",
   "lastEventId": {
      "string" : "string"
   },
   "lastTrxTimestamp": number,
   "records": [
      {
         "commitTimestamp": number,
         "data": {
            "from": "string",
            "id": "string",
            "key": "string",
            "to": "string",
            "type": "string",
            "value": JSON value
         },
         "eventId": {
            "string" : "string"
         },
         "isLastOp": boolean,
         "op": "string"
      }
   ],
   "totalRecords": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

format

Serialization format for the change records being returned. Currently, the only supported value is `PG_JSON`.

Type: String

lastEventId

Sequence identifier of the last change in the stream response.

An event ID is composed of two fields: a `commitNum`, which identifies a transaction that changed the graph, and an `opNum`, which identifies a specific operation within that transaction:

Type: String to string map

lastTrxTimestamp

The time at which the commit for the transaction was requested, in milliseconds from the Unix epoch.

Type: Long

records

An array of serialized change-log stream records included in the response.

Type: Array of PropertygraphRecord objects

totalRecords

The total number of records in the response.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

ExpiredStreamException

Raised when a request attempts to access an stream that has expired.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

StreamRecordsNotFoundException

Raised when stream records requested by a query cannot be found.

HTTP Status Code: 404

ThrottlingException

Raised when the rate of requests exceeds the maximum throughput. Requests can be retried after encountering this exception.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetPropertygraphSummary

Gets a graph summary for a property graph.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetGraphSummary IAM action in that cluster.

Request Syntax

```
GET /propertygraph/statistics/summary?mode=mode HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

mode

Mode can take one of two values: `BASIC` (the default), and `DETAILED`.

Valid Values: `basic | detailed`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 statusCode
Content-type: application/json

{
   "payload": {
      "graphSummary": {
         "edgeLabels": [ "string" ],
         "edgeProperties": [
            {
               "string" : number
            }
```

```
         ],
         "edgeStructures": [
            {
               "count": number,
               "edgeProperties": [ "string" ]
            }
         ],
         "nodeLabels": [ "string" ],
         "nodeProperties": [
            {
               "string" : number
            }
         ],
         "nodeStructures": [
            {
               "count": number,
               "distinctOutgoingEdgeLabels": [ "string" ],
               "nodeProperties": [ "string" ]
            }
         ],
         "numEdgeLabels": number,
         "numEdgeProperties": number,
         "numEdges": number,
         "numNodeLabels": number,
         "numNodeProperties": number,
         "numNodes": number,
         "totalEdgePropertyValues": number,
         "totalNodePropertyValues": number
      },
      "lastStatisticsComputationTime": "string",
      "version": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back the following HTTP response.

statusCode

The HTTP return code of the request. If the request succeeded, the code is 200.

The following data is returned in JSON format by the service.

payload

Payload containing the property graph summary response.

Type: PropertygraphSummaryValueMap object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetRDFGraphSummary

Gets a graph summary for an RDF graph.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetGraphSummary IAM action in that cluster.

Request Syntax

```
GET /rdf/statistics/summary?mode=mode HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

mode

Mode can take one of two values: `BASIC` (the default), and `DETAILED`.

Valid Values: `basic | detailed`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 statusCode
Content-type: application/json

{
   "payload": {
      "graphSummary": {
         "classes": [ "string" ],
         "numClasses": number,
         "numDistinctPredicates": number,
         "numDistinctSubjects": number,
         "numQuads": number,
         "predicates": [
```

```
            {
               "string" : number
            }
         ],
         "subjectStructures": [
            {
               "count": number,
               "predicates": [ "string" ]
            }
         ]
      },
      "lastStatisticsComputationTime": "string",
      "version": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back the following HTTP response.

statusCode

The HTTP return code of the request. If the request succeeded, the code is 200.

The following data is returned in JSON format by the service.

payload

Payload for an RDF graph summary response

Type: RDFGraphSummaryValueMap object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSparqlStatistics

Gets RDF statistics (SPARQL).

Request Syntax

```
GET /sparql/statistics HTTP/1.1
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": {
      "active": boolean,
      "autoCompute": boolean,
      "date": "string",
      "note": "string",
      "signatureInfo": {
         "instanceCount": number,
         "predicateCount": number,
         "signatureCount": number
      },
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

Statistics for RDF data.

Type: Statistics object

status

The HTTP return code of the request. If the request succeeded, the code is 200. See Common error codes for DFE statistics request for a list of common errors.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetStatisticsStatus IAM action in that cluster.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSparqlStream

Gets a stream for an RDF graph.

With the Neptune Streams feature, you can generate a complete sequence of change-log entries that record every change made to your graph data as it happens. `GetSparqlStream` lets you collect these change-log entries for an RDF graph.

The Neptune streams feature needs to be enabled on your Neptune DBcluster. To enable streams, set the neptune_streams DB cluster parameter to 1.

See Capturing graph changes in real time using Neptune streams.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetStreamRecords IAM action in that cluster.

Note that the neptune-db:QueryLanguage:Sparql IAM condition key can be used in the policy document to restrict the use of SPARQL queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /sparql/stream?
commitNum=commitNum&iteratorType=iteratorType&limit=limit&opNum=opNum HTTP/1.1
Accept-Encoding: encoding
```

URI Request Parameters

The request uses the following URI parameters.

commitNum

The commit number of the starting record to read from the change-log stream. This parameter is required when `iteratorType` is`AT_SEQUENCE_NUMBER` or `AFTER_SEQUENCE_NUMBER`, and ignored when `iteratorType` is `TRIM_HORIZON` or `LATEST`.

encoding

If set to TRUE, Neptune compresses the response using gzip encoding.

Valid Values: `gzip`

iteratorType

Can be one of:

- `AT_SEQUENCE_NUMBER` – Indicates that reading should start from the event sequence number specified jointly by the `commitNum` and `opNum` parameters.
- `AFTER_SEQUENCE_NUMBER` – Indicates that reading should start right after the event sequence number specified jointly by the `commitNum` and `opNum` parameters.
- `TRIM_HORIZON` – Indicates that reading should start at the last untrimmed record in the system, which is the oldest unexpired (not yet deleted) record in the change-log stream.
- `LATEST` – Indicates that reading should start at the most recent record in the system, which is the latest unexpired (not yet deleted) record in the change-log stream.

Valid Values: `AT_SEQUENCE_NUMBER | AFTER_SEQUENCE_NUMBER | TRIM_HORIZON | LATEST`

limit

Specifies the maximum number of records to return. There is also a size limit of 10 MB on the response that can't be modified and that takes precedence over the number of records specified in the `limit` parameter. The response does include a threshold-breaching record if the 10 MB limit was reached.

The range for `limit` is 1 to 100,000, with a default of 10.

Valid Range: Minimum value of 1. Maximum value of 100000.

opNum

The operation sequence number within the specified commit to start reading from in the change-log stream data. The default is 1.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "format": "string",
   "lastEventId": {
      "string" : "string"
   },
   "lastTrxTimestamp": number,
   "records": [
      {
         "commitTimestamp": number,
         "data": {
            "stmt": "string"
         },
         "eventId": {
            "string" : "string"
         },
         "isLastOp": boolean,
         "op": "string"
      }
   ],
   "totalRecords": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

format

Serialization format for the change records being returned. Currently, the only supported value is NQUADS.

Type: String

lastEventId

Sequence identifier of the last change in the stream response.

An event ID is composed of two fields: a `commitNum`, which identifies a transaction that changed the graph, and an `opNum`, which identifies a specific operation within that transaction:

Type: String to string map

lastTrxTimestamp

The time at which the commit for the transaction was requested, in milliseconds from the Unix epoch.

Type: Long

records

An array of serialized change-log stream records included in the response.

Type: Array of SparqlRecord objects

totalRecords

The total number of records in the response.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

ExpiredStreamException

Raised when a request attempts to access an stream that has expired.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MemoryLimitExceededException

Raised when a request fails because of insufficient memory resources. The request can be retried.

HTTP Status Code: 500

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

StreamRecordsNotFoundException

Raised when stream records requested by a query cannot be found.

HTTP Status Code: 404

ThrottlingException

Raised when the rate of requests exceeds the maximum throughput. Requests can be retried after encountering this exception.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListGremlinQueries

Lists active Gremlin queries. See Gremlin query status API for details about the output.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetQueryStatus IAM action in that cluster.

Note that the neptune-db:QueryLanguage:Gremlin IAM condition key can be used in the policy document to restrict the use of Gremlin queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /gremlin/status?includeWaiting=includeWaiting HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

includeWaiting

If set to TRUE, the list returned includes waiting queries. The default is FALSE;

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "acceptedQueryCount": number,
   "queries": [
      {
         "queryEvalStats": {
            "cancelled": boolean,
            "elapsed": number,
```

```
            "subqueries": JSON value,
            "waited": number
         },
         "queryId": "string",
         "queryString": "string"
      }
   ],
   "runningQueryCount": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

acceptedQueryCount

The number of queries that have been accepted but not yet completed, including queries in the queue.

Type: Integer

queries

A list of the current queries.

Type: Array of GremlinQueryStatus objects

runningQueryCount

The number of Gremlin queries currently running.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListLoaderJobs

Retrieves a list of the `loadIds` for all active loader jobs.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ListLoaderJobs IAM action in that cluster..

Request Syntax

```
GET /loader?includeQueuedLoads=includeQueuedLoads&limit=limit HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

includeQueuedLoads

An optional parameter that can be used to exclude the load IDs of queued load requests when requesting a list of load IDs by setting the parameter to FALSE. The default value is TRUE.

limit

The number of load IDs to list. Must be a positive integer greater than zero and not more than 100 (which is the default).

Valid Range: Minimum value of 1. Maximum value of 100.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": {
      "loadIds": [ "string" ]
```

```
    },
    "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

The requested list of job IDs.

Type: LoaderIdResult object

status

Returns the status of the job list request.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

BulkLoadIdNotFoundException

Raised when a specified bulk-load job ID cannot be found.

HTTP Status Code: 404

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InternalFailureException

Raised when the processing of the request failed unexpectedly.

HTTP Status Code: 500

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

LoadUrlAccessDeniedException

Raised when access is denied to a specified load URL.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMLDataProcessingJobs

Returns a list of Neptune ML data processing jobs. See Listing active data-processing jobs using the Neptune ML dataprocessing command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ListMLDataProcessingJobs IAM action in that cluster.

Request Syntax

```
GET /ml/dataprocessing?maxItems=maxItems&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxItems

The maximum number of items to return (from 1 to 1024; the default is 10).

Valid Range: Minimum value of 1. Maximum value of 1024.

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ids": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ids

A page listing data processing job IDs.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMLEndpoints

Lists existing inference endpoints. See Managing inference endpoints using the endpoints command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ListMLEndpoints IAM action in that cluster.

Request Syntax

```
GET /ml/endpoints?maxItems=maxItems&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxItems

The maximum number of items to return (from 1 to 1024; the default is 10.

Valid Range: Minimum value of 1. Maximum value of 1024.

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ids": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ids

A page from the list of inference endpoint IDs.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMLModelTrainingJobs

Lists Neptune ML model-training jobs. See Model training using the `modeltraining` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:neptune-db:ListMLModelTrainingJobs IAM action in that cluster.

Request Syntax

```
GET /ml/modeltraining?maxItems=maxItems&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxItems

The maximum number of items to return (from 1 to 1024; the default is 10).

Valid Range: Minimum value of 1. Maximum value of 1024.

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ids": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ids

A page of the list of model training job IDs.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMLModelTransformJobs

Returns a list of model transform job IDs. See Use a trained model to generate new model artifacts.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ListMLModelTransformJobs IAM action in that cluster.

Request Syntax

```
GET /ml/modeltransform?maxItems=maxItems&neptuneIamRoleArn=neptuneIamRoleArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxItems

The maximum number of items to return (from 1 to 1024; the default is 10).

Valid Range: Minimum value of 1. Maximum value of 1024.

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ids": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ids

A page from the list of model transform IDs.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListOpenCypherQueries

Lists active openCypher queries. See Neptune openCypher status endpoint for more information.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:GetQueryStatus IAM action in that cluster.

Note that the neptune-db:QueryLanguage:OpenCypher IAM condition key can be used in the policy document to restrict the use of openCypher queries (see Condition keys available in Neptune IAM data-access policy statements).

Request Syntax

```
GET /opencypher/status?includeWaiting=includeWaiting HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

includeWaiting

When set to TRUE and other parameters are not present, causes status information to be returned for waiting queries as well as for running queries.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "acceptedQueryCount": number,
   "queries": [
      {
         "queryEvalStats": {
            "cancelled": boolean,
```

```
            "elapsed": number,
            "subqueries": JSON value,
            "waited": number
         },
         "queryId": "string",
         "queryString": "string"
      }
   ],
   "runningQueryCount": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

acceptedQueryCount

The number of queries that have been accepted but not yet completed, including queries in the queue.

Type: Integer

queries

A list of current openCypher queries.

Type: Array of GremlinQueryStatus objects

runningQueryCount

The number of currently running openCypher queries.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConcurrentModificationException

Raised when a request attempts to modify data that is concurrently being modified by another process.

HTTP Status Code: 500

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

FailureByQueryException

Raised when a request fails.

HTTP Status Code: 500

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidNumericDataException

Raised when invalid numerical data is encountered when servicing a request.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

ParsingException

Raised when a parsing issue is encountered.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

TimeLimitExceededException

Raised when the an operation exceeds the time limit allowed for it.

HTTP Status Code: 500

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ManagePropertygraphStatistics

Manages the generation and use of property graph statistics.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ManageStatistics IAM action in that cluster.

Request Syntax

```
POST /propertygraph/statistics HTTP/1.1
Content-type: application/json

{
   "mode": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

mode

The statistics generation mode. One of: `DISABLE_AUTOCOMPUTE`, `ENABLE_AUTOCOMPUTE`, or `REFRESH`, the last of which manually triggers DFE statistics generation.

Type: String

Valid Values: `disableAutoCompute | enableAutoCompute | refresh`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "payload": {
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

This is only returned for refresh mode.

Type: RefreshStatisticsIdMap object

status

The HTTP return code of the request. If the request succeeded, the code is 200.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ManageSparqlStatistics

Manages the generation and use of RDF graph statistics.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:ManageStatistics IAM action in that cluster.

Request Syntax

```
POST /sparql/statistics HTTP/1.1
Content-type: application/json

{
   "mode": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

mode

The statistics generation mode. One of: `DISABLE_AUTOCOMPUTE`, `ENABLE_AUTOCOMPUTE`, or `REFRESH`, the last of which manually triggers DFE statistics generation.

Type: String

Valid Values: `disableAutoCompute | enableAutoCompute | refresh`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "payload": {
      "statisticsId": "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

This is only returned for refresh mode.

Type: RefreshStatisticsIdMap object

status

The HTTP return code of the request. If the request succeeded, the code is 200.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Raised in case of an authentication or authorization failure.

HTTP Status Code: 403

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

ReadOnlyViolationException

Raised when a request attempts to write to a read-only resource.

HTTP Status Code: 400

StatisticsNotAvailableException

Raised when statistics needed to satisfy a request are not available.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartLoaderJob

Starts a Neptune bulk loader job to load data from an Amazon S3 bucket into a Neptune DB instance. See Using the Amazon Neptune Bulk Loader to Ingest Data.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:StartLoaderJob IAM action in that cluster.

Request Syntax

```
POST /loader HTTP/1.1
Content-type: application/json

{
   "dependencies": [ "string" ],
   "failOnError": boolean,
   "format": "string",
   "iamRoleArn": "string",
   "mode": "string",
   "parallelism": "string",
   "parserConfiguration": {
      "string" : "string"
   },
   "queueRequest": boolean,
   "region": "string",
   "source": "string",
   "updateSingleCardinalityProperties": boolean,
   "userProvidedEdgeIds": boolean
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

dependencies

This is an optional parameter that can make a queued load request contingent on the successful completion of one or more previous jobs in the queue.

Neptune can queue up as many as 64 load requests at a time, if their `queueRequest` parameters are set to `"TRUE"`. The `dependencies` parameter lets you make execution of such a queued request dependent on the successful completion of one or more specified previous requests in the queue.

For example, if load `Job-A` and `Job-B` are independent of each other, but load `Job-C` needs `Job-A` and `Job-B` to be finished before it begins, proceed as follows:

1. Submit `load-job-A` and `load-job-B` one after another in any order, and save their load-ids.
2. Submit `load-job-C` with the load-ids of the two jobs in its `dependencies` field:

Because of the `dependencies` parameter, the bulk loader will not start `Job-C` until `Job-A` and `Job-B` have completed successfully. If either one of them fails, Job-C will not be executed, and its status will be set to `LOAD_FAILED_BECAUSE_DEPENDENCY_NOT_SATISFIED`.

You can set up multiple levels of dependency in this way, so that the failure of one job will cause all requests that are directly or indirectly dependent on it to be cancelled.

Type: Array of strings

Required: No

failOnError

`failOnError` – A flag to toggle a complete stop on an error.

Allowed values: `"TRUE"`, `"FALSE"`.

Default value: `"TRUE"`.

When this parameter is set to `"FALSE"`, the loader tries to load all the data in the location specified, skipping any entries with errors.

When this parameter is set to `"TRUE"`, the loader stops as soon as it encounters an error. Data loaded up to that point persists.

Type: Boolean

Required: No

format

The format of the data. For more information about data formats for the Neptune `Loader` command, see Load Data Formats.

Allowed values

- **`csv`** for the Gremlin CSV data format.
- **`opencypher`** for the openCypher CSV data format.
- **`ntriples`** for the N-Triples RDF data format.
- **`nquads`** for the N-Quads RDF data format.
- **`rdfxml`** for the RDF\XML RDF data format.
- **`turtle`** for the Turtle RDF data format.

Type: String

Valid Values: `csv | opencypher | ntriples | nquads | rdfxml | turtle`

Required: Yes

iamRoleArn

The Amazon Resource Name (ARN) for an IAM role to be assumed by the Neptune DB instance for access to the S3 bucket. The IAM role ARN provided here should be attached to the DB cluster (see Adding the IAM Role to an Amazon Neptune Cluster.

Type: String

Required: Yes

mode

The load job mode.

Allowed values: `RESUME`, `NEW`, `AUTO`.

Default value: `AUTO`.

- `RESUME` – In RESUME mode, the loader looks for a previous load from this source, and if it finds one, resumes that load job. If no previous load job is found, the loader stops.

The loader avoids reloading files that were successfully loaded in a previous job. It only tries to process failed files. If you dropped previously loaded data from your Neptune cluster, that data is not reloaded in this mode. If a previous load job loaded all files from the same source successfully, nothing is reloaded, and the loader returns success.

- `NEW` – In NEW mode, the creates a new load request regardless of any previous loads. You can use this mode to reload all the data from a source after dropping previously loaded data from your Neptune cluster, or to load new data available at the same source.
- `AUTO` – In AUTO mode, the loader looks for a previous load job from the same source, and if it finds one, resumes that job, just as in `RESUME` mode.

  If the loader doesn't find a previous load job from the same source, it loads all data from the source, just as in `NEW` mode.

Type: String

Valid Values: `RESUME` | `NEW` | `AUTO`

Required: No

parallelism

The optional `parallelism` parameter can be set to reduce the number of threads used by the bulk load process.

Allowed values:

- `LOW` – The number of threads used is the number of available vCPUs divided by 8.
- `MEDIUM` – The number of threads used is the number of available vCPUs divided by 2.
- `HIGH` – The number of threads used is the same as the number of available vCPUs.
- `OVERSUBSCRIBE` – The number of threads used is the number of available vCPUs multiplied by 2. If this value is used, the bulk loader takes up all available resources.

  This does not mean, however, that the `OVERSUBSCRIBE` setting results in 100% CPU utilization. Because the load operation is I/O bound, the highest CPU utilization to expect is in the 60% to 70% range.

Default value: `HIGH`

The `parallelism` setting can sometimes result in a deadlock between threads when loading openCypher data. When this happens, Neptune returns the `LOAD_DATA_DEADLOCK` error. You

can generally fix the issue by setting `parallelism` to a lower setting and retrying the load command.

Type: String

Valid Values: `LOW | MEDIUM | HIGH | OVERSUBSCRIBE`

Required: No

parserConfiguration

`parserConfiguration` – An optional object with additional parser configuration values. Each of the child parameters is also optional:

- **`namedGraphUri`** – The default graph for all RDF formats when no graph is specified (for non-quads formats and NQUAD entries with no graph).

  The default is `https://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`.
- **`baseUri`** – The base URI for RDF/XML and Turtle formats.

  The default is `https://aws.amazon.com/neptune/default`.
- **`allowEmptyStrings`** – Gremlin users need to be able to pass empty string values("") as node and edge properties when loading CSV data. If `allowEmptyStrings` is set to `false` (the default), such empty strings are treated as nulls and are not loaded.

  If `allowEmptyStrings` is set to `true`, the loader treats empty strings as valid property values and loads them accordingly.

Type: String to string map

Required: No

queueRequest

This is an optional flag parameter that indicates whether the load request can be queued up or not.

You don't have to wait for one load job to complete before issuing the next one, because Neptune can queue up as many as 64 jobs at a time, provided that their `queueRequest` parameters are all set to `"TRUE"`. The queue order of the jobs will be first-in-first-out (FIFO).

If the `queueRequest` parameter is omitted or set to `"FALSE"`, the load request will fail if another load job is already running.

Allowed values: "TRUE", "FALSE".

Default value: "FALSE".

Type: Boolean

Required: No

region

The Amazon region of the S3 bucket. This must match the Amazon Region of the DB cluster.

Type: String

Valid Values: us-east-1 | us-east-2 | us-west-1 | us-west-2 | ca-central-1 | sa-east-1 | eu-north-1 | eu-west-1 | eu-west-2 | eu-west-3 | eu-central-1 | me-south-1 | af-south-1 | ap-east-1 | ap-northeast-1 | ap-northeast-2 | ap-southeast-1 | ap-southeast-2 | ap-south-1 | cn-north-1 | cn-northwest-1 | us-gov-west-1 | us-gov-east-1

Required: Yes

source

The source parameter accepts an S3 URI that identifies a single file, multiple files, a folder, or multiple folders. Neptune loads every data file in any folder that is specified.

The URI can be in any of the following formats.

- s3://(bucket_name)/(object-key-name)
- https://s3.amazonaws.com/(bucket_name)/(object-key-name)
- https://s3.us-east-1.amazonaws.com/(bucket_name)/(object-key-name)

The object-key-name element of the URI is equivalent to the prefix parameter in an S3 ListObjects API call. It identifies all the objects in the specified S3 bucket whose names begin with that prefix. That can be a single file or folder, or multiple files and/or folders.

The specified folder or folders can contain multiple vertex files and multiple edge files.

Type: String

Required: Yes

updateSingleCardinalityProperties

`updateSingleCardinalityProperties` is an optional parameter that controls how the bulk loader treats a new value for single-cardinality vertex or edge properties. This is not supported for loading openCypher data.

Allowed values: `"TRUE"`, `"FALSE"`.

Default value: `"FALSE"`.

By default, or when `updateSingleCardinalityProperties` is explicitly set to `"FALSE"`, the loader treats a new value as an error, because it violates single cardinality.

When `updateSingleCardinalityProperties` is set to `"TRUE"`, on the other hand, the bulk loader replaces the existing value with the new one. If multiple edge or single-cardinality vertex property values are provided in the source file(s) being loaded, the final value at the end of the bulk load could be any one of those new values. The loader only guarantees that the existing value has been replaced by one of the new ones.

Type: Boolean

Required: No

userProvidedEdgeIds

This parameter is required only when loading openCypher data that contains relationship IDs. It must be included and set to `True` when openCypher relationship IDs are explicitly provided in the load data (recommended).

When `userProvidedEdgeIds` is absent or set to `True`, an `:ID` column must be present in every relationship file in the load.

When `userProvidedEdgeIds` is present and set to `False`, relationship files in the load **must not** contain an `:ID` column. Instead, the Neptune loader automatically generates an ID for each relationship.

It's useful to provide relationship IDs explicitly so that the loader can resume loading after error in the CSV data have been fixed, without having to reload any relationships that have already been loaded. If relationship IDs have not been explicitly assigned, the loader cannot resume a failed load if any relationship file has had to be corrected, and must instead reload all the relationships.

Type: Boolean

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "payload": {
      "string" : "string"
   },
   "status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

payload

Contains a `loadId` name-value pair that provides an identifier for the load operation.

Type: String to string map

status

The HTTP return code indicating the status of the load job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

BulkLoadIdNotFoundException

Raised when a specified bulk-load job ID cannot be found.

HTTP Status Code: 404

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InternalFailureException

Raised when the processing of the request failed unexpectedly.

HTTP Status Code: 500

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

LoadUrlAccessDeniedException

Raised when access is denied to a specified load URL.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

S3Exception

Raised when there is a problem accessing Amazon S3.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

StartMLDataProcessingJob

Creates a new Neptune ML data processing job for processing the graph data exported from Neptune for training. See The `dataprocessing` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:StartMLModelDataProcessingJob IAM action in that cluster.

Request Syntax

```
POST /ml/dataprocessing HTTP/1.1
Content-type: application/json

{
   "configFileName": "string",
   "id": "string",
   "inputDataS3Location": "string",
   "modelType": "string",
   "neptuneIamRoleArn": "string",
   "previousDataProcessingJobId": "string",
   "processedDataS3Location": "string",
   "processingInstanceType": "string",
   "processingInstanceVolumeSizeInGB": number,
   "processingTimeOutInSeconds": number,
   "s3OutputEncryptionKMSKey": "string",
   "sagemakerIamRoleArn": "string",
   "securityGroupIds": [ "string" ],
   "subnets": [ "string" ],
   "volumeEncryptionKMSKey": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

configFileName

A data specification file that describes how to load the exported graph data for training. The file is automatically generated by the Neptune export toolkit. The default is `training-data-configuration.json`.

Type: String

Required: No

id

A unique identifier for the new job. The default is an autogenerated UUID.

Type: String

Required: No

inputDataS3Location

The URI of the Amazon S3 location where you want SageMaker to download the data needed to run the data processing job.

Type: String

Required: Yes

modelType

One of the two model types that Neptune ML currently supports: heterogeneous graph models (`heterogeneous`), and knowledge graph (`kge`). The default is none. If not specified, Neptune ML chooses the model type automatically based on the data.

Type: String

Required: No

neptuneIamRoleArn

The Amazon Resource Name (ARN) of an IAM role that SageMaker can assume to perform tasks on your behalf. This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

previousDataProcessingJobId

The job ID of a completed data processing job run on an earlier version of the data.

Type: String

Required: No

processedDataS3Location

The URI of the Amazon S3 location where you want SageMaker to save the results of a data processing job.

Type: String

Required: Yes

processingInstanceType

The type of ML instance used during data processing. Its memory should be large enough to hold the processed dataset. The default is the smallest ml.r5 type whose memory is ten times larger than the size of the exported graph data on disk.

Type: String

Required: No

processingInstanceVolumeSizeInGB

The disk volume size of the processing instance. Both input data and processed data are stored on disk, so the volume size must be large enough to hold both data sets. The default is 0. If not specified or 0, Neptune ML chooses the volume size automatically based on the data size.

Type: Integer

Required: No

processingTimeOutInSeconds

Timeout in seconds for the data processing job. The default is 86,400 (1 day).

Type: Integer

Required: No

s3OutputEncryptionKMSKey

The Amazon Key Management Service (Amazon KMS) key that SageMaker uses to encrypt the output of the processing job. The default is none.

Type: String

Required: No

sagemakerIamRoleArn

The ARN of an IAM role for SageMaker execution. This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

securityGroupIds

The VPC security group IDs. The default is None.

Type: Array of strings

Required: No

subnets

The IDs of the subnets in the Neptune VPC. The default is None.

Type: Array of strings

Required: No

volumeEncryptionKMSKey

The Amazon Key Management Service (Amazon KMS) key that SageMaker uses to encrypt data on the storage volume attached to the ML compute instances that run the training job. The default is None.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "creationTimeInMillis": number,
   "id": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The ARN of the data processing job.

Type: String

creationTimeInMillis

The time it took to create the new processing job, in milliseconds.

Type: Long

id

The unique ID of the new data processing job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartMLModelTrainingJob

Creates a new Neptune ML model training job. See Model training using the `modeltraining` command.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:StartMLModelTrainingJob IAM action in that cluster.

Request Syntax

```
POST /ml/modeltraining HTTP/1.1
Content-type: application/json

{
   "baseProcessingInstanceType": "string",
   "customModelTrainingParameters": {
      "sourceS3DirectoryPath": "string",
      "trainingEntryPointScript": "string",
      "transformEntryPointScript": "string"
   },
   "dataProcessingJobId": "string",
   "enableManagedSpotTraining": boolean,
   "id": "string",
   "maxHPONumberOfTrainingJobs": number,
   "maxHPOParallelTrainingJobs": number,
   "neptuneIamRoleArn": "string",
   "previousModelTrainingJobId": "string",
   "s3OutputEncryptionKMSKey": "string",
   "sagemakerIamRoleArn": "string",
   "securityGroupIds": [ "string" ],
   "subnets": [ "string" ],
   "trainingInstanceType": "string",
   "trainingInstanceVolumeSizeInGB": number,
   "trainingTimeOutInSeconds": number,
   "trainModelS3Location": "string",
   "volumeEncryptionKMSKey": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

baseProcessingInstanceType

The type of ML instance used in preparing and managing training of ML models. This is a CPU instance chosen based on memory requirements for processing the training data and model.

Type: String

Required: No

customModelTrainingParameters

The configuration for custom model training. This is a JSON object.

Type: CustomModelTrainingParameters object

Required: No

dataProcessingJobId

The job ID of the completed data-processing job that has created the data that the training will work with.

Type: String

Required: Yes

enableManagedSpotTraining

Optimizes the cost of training machine-learning models by using Amazon Elastic Compute Cloud spot instances. The default is `False`.

Type: Boolean

Required: No

id

A unique identifier for the new job. The default is An autogenerated UUID.

Type: String

Required: No

maxHPONumberOfTrainingJobs

Maximum total number of training jobs to start for the hyperparameter tuning job. The default is 2. Neptune ML automatically tunes the hyperparameters of the machine learning model. To obtain a model that performs well, use at least 10 jobs (in other words, set `maxHPONumberOfTrainingJobs` to 10). In general, the more tuning runs, the better the results.

Type: Integer

Required: No

maxHPOParallelTrainingJobs

Maximum number of parallel training jobs to start for the hyperparameter tuning job. The default is 2. The number of parallel jobs you can run is limited by the available resources on your training instance.

Type: Integer

Required: No

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

previousModelTrainingJobId

The job ID of a completed model-training job that you want to update incrementally based on updated data.

Type: String

Required: No

s3OutputEncryptionKMSKey

The Amazon Key Management Service (KMS) key that SageMaker uses to encrypt the output of the processing job. The default is none.

Type: String

Required: No

sagemakerIamRoleArn

The ARN of an IAM role for SageMaker execution.This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

securityGroupIds

The VPC security group IDs. The default is None.

Type: Array of strings

Required: No

subnets

The IDs of the subnets in the Neptune VPC. The default is None.

Type: Array of strings

Required: No

trainingInstanceType

The type of ML instance used for model training. All Neptune ML models support CPU, GPU, and multiGPU training. The default is `ml.p3.2xlarge`. Choosing the right instance type for training depends on the task type, graph size, and your budget.

Type: String

Required: No

trainingInstanceVolumeSizeInGB

The disk volume size of the training instance. Both input data and the output model are stored on disk, so the volume size must be large enough to hold both data sets. The default is 0. If not specified or 0, Neptune ML selects a disk volume size based on the recommendation generated in the data processing step.

Type: Integer

Required: No

trainingTimeOutInSeconds

Timeout in seconds for the training job. The default is 86,400 (1 day).

Type: Integer

Required: No

trainModelS3Location

The location in Amazon S3 where the model artifacts are to be stored.

Type: String

Required: Yes

volumeEncryptionKMSKey

The Amazon Key Management Service (KMS) key that SageMaker uses to encrypt data on the storage volume attached to the ML compute instances that run the training job. The default is None.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "creationTimeInMillis": number,
   "id": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The ARN of the new model training job.

Type: String

creationTimeInMillis

The model training job creation time, in milliseconds.

Type: Long

id

The unique ID of the new model training job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartMLModelTransformJob

Creates a new model transform job. See Use a trained model to generate new model artifacts.

When invoking this operation in a Neptune cluster that has IAM authentication enabled, the IAM user or role making the request must have a policy attached that allows the neptune-db:StartMLModelTransformJob IAM action in that cluster.

Request Syntax

```
POST /ml/modeltransform HTTP/1.1
Content-type: application/json

{
   "baseProcessingInstanceType": "string",
   "baseProcessingInstanceVolumeSizeInGB": number,
   "customModelTransformParameters": {
      "sourceS3DirectoryPath": "string",
      "transformEntryPointScript": "string"
   },
   "dataProcessingJobId": "string",
   "id": "string",
   "mlModelTrainingJobId": "string",
   "modelTransformOutputS3Location": "string",
   "neptuneIamRoleArn": "string",
   "s3OutputEncryptionKMSKey": "string",
   "sagemakerIamRoleArn": "string",
   "securityGroupIds": [ "string" ],
   "subnets": [ "string" ],
   "trainingJobName": "string",
   "volumeEncryptionKMSKey": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

baseProcessingInstanceType

The type of ML instance used in preparing and managing training of ML models. This is an ML compute instance chosen based on memory requirements for processing the training data and model.

Type: String

Required: No

baseProcessingInstanceVolumeSizeInGB

The disk volume size of the training instance in gigabytes. The default is 0. Both input data and the output model are stored on disk, so the volume size must be large enough to hold both data sets. If not specified or 0, Neptune ML selects a disk volume size based on the recommendation generated in the data processing step.

Type: Integer

Required: No

customModelTransformParameters

Configuration information for a model transform using a custom model. The `customModelTransformParameters` object contains the following fields, which must have values compatible with the saved model parameters from the training job:

Type: CustomModelTransformParameters object

Required: No

dataProcessingJobId

The job ID of a completed data-processing job. You must include either `dataProcessingJobId` and a `mlModelTrainingJobId`, or a `trainingJobName`.

Type: String

Required: No

id

A unique identifier for the new job. The default is an autogenerated UUID.

Type: String

Required: No

mlModelTrainingJobId

The job ID of a completed model-training job. You must include either `dataProcessingJobId` and a `mlModelTrainingJobId`, or a `trainingJobName`.

Type: String

Required: No

modelTransformOutputS3Location

The location in Amazon S3 where the model artifacts are to be stored.

Type: String

Required: Yes

neptuneIamRoleArn

The ARN of an IAM role that provides Neptune access to SageMaker and Amazon S3 resources. This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

s3OutputEncryptionKMSKey

The Amazon Key Management Service (KMS) key that SageMaker uses to encrypt the output of the processing job. The default is none.

Type: String

Required: No

sagemakerIamRoleArn

The ARN of an IAM role for SageMaker execution. This must be listed in your DB cluster parameter group or an error will occur.

Type: String

Required: No

securityGroupIds

The VPC security group IDs. The default is None.

Type: Array of strings

Required: No

subnets

The IDs of the subnets in the Neptune VPC. The default is None.

Type: Array of strings

Required: No

trainingJobName

The name of a completed SageMaker training job. You must include either `dataProcessingJobId` and a `mlModelTrainingJobId`, or a `trainingJobName`.

Type: String

Required: No

volumeEncryptionKMSKey

The Amazon Key Management Service (KMS) key that SageMaker uses to encrypt data on the storage volume attached to the ML compute instances that run the training job. The default is None.

Type: String

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "creationTimeInMillis": number,
   "id": "string"
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

arn

The ARN of the model transform job.

Type: String

creationTimeInMillis

The creation time of the model transform job, in milliseconds.

Type: Long

id

The unique ID of the new model transform job.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

BadRequestException

Raised when a request is submitted that cannot be processed.

HTTP Status Code: 400

ClientTimeoutException

Raised when a request timed out in the client.

HTTP Status Code: 408

ConstraintViolationException

Raised when a value in a request field did not satisfy required constraints.

HTTP Status Code: 400

IllegalArgumentException

Raised when an argument in a request is not supported.

HTTP Status Code: 400

InvalidArgumentException

Raised when an argument in a request has an invalid value.

HTTP Status Code: 400

InvalidParameterException

Raised when a parameter value is not valid.

HTTP Status Code: 400

MissingParameterException

Raised when a required parameter is missing.

HTTP Status Code: 400

MLResourceNotFoundException

Raised when a specified machine-learning resource could not be found.

HTTP Status Code: 404

PreconditionsFailedException

Raised when a precondition for processing a request is not satisfied.

HTTP Status Code: 400

TooManyRequestsException

Raised when the number of requests being processed exceeds the limit.

HTTP Status Code: 429

UnsupportedOperationException

Raised when a request attempts to initiate an operation that is not supported.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The Amazon NeptuneData API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
>
> The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- CustomModelTrainingParameters
- CustomModelTransformParameters
- DeleteStatisticsValueMap
- EdgeStructure
- FastResetToken
- GremlinQueryStatus
- GremlinQueryStatusAttributes
- LoaderIdResult
- MlConfigDefinition
- MlResourceDefinition
- NodeStructure
- PropertygraphData
- PropertygraphRecord
- PropertygraphSummary
- PropertygraphSummaryValueMap
- QueryEvalStats
- QueryLanguageVersion
- RDFGraphSummary
- RDFGraphSummaryValueMap
- RefreshStatisticsIdMap

- SparqlData
- SparqlRecord
- Statistics
- StatisticsSummary
- SubjectStructure

CustomModelTrainingParameters

Contains custom model training parameters. See Custom models in Neptune ML.

Contents

sourceS3DirectoryPath

The path to the Amazon S3 location where the Python module implementing your model is located. This must point to a valid existing Amazon S3 location that contains, at a minimum, a training script, a transform script, and a `model-hpo-configuration.json` file.

Type: String

Required: Yes

trainingEntryPointScript

The name of the entry point in your module of a script that performs model training and takes hyperparameters as command-line arguments, including fixed hyperparameters. The default is `training.py`.

Type: String

Required: No

transformEntryPointScript

The name of the entry point in your module of a script that should be run after the best model from the hyperparameter search has been identified, to compute the model artifacts necessary for model deployment. It should be able to run with no command-line arguments.The default is `transform.py`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CustomModelTransformParameters

Contains custom model transform parameters. See Use a trained model to generate new model artifacts.

Contents

sourceS3DirectoryPath

The path to the Amazon S3 location where the Python module implementing your model is located. This must point to a valid existing Amazon S3 location that contains, at a minimum, a training script, a transform script, and a `model-hpo-configuration.json` file.

Type: String

Required: Yes

transformEntryPointScript

The name of the entry point in your module of a script that should be run after the best model from the hyperparameter search has been identified, to compute the model artifacts necessary for model deployment. It should be able to run with no command-line arguments. The default is `transform.py`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteStatisticsValueMap

The payload for DeleteStatistics.

Contents

active

The current status of the statistics.

Type: Boolean

Required: No

statisticsId

The ID of the statistics generation run that is currently occurring.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EdgeStructure

An edge structure.

Contents

count

The number of edges that have this specific structure.

Type: Long

Required: No

edgeProperties

A list of edge properties present in this specific structure.

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FastResetToken

A structure containing the fast reset token used to initiate a fast reset.

Contents

token

A UUID generated by the database in the `initiateDatabaseReset` action, and then consumed by the `performDatabaseReset` to reset the database.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GremlinQueryStatus

Captures the status of a Gremlin query (see the Gremlin query status API page).

Contents

queryEvalStats

The query statistics of the Gremlin query.

Type: QueryEvalStats object

Required: No

queryId

The ID of the Gremlin query.

Type: String

Required: No

queryString

The query string of the Gremlin query.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GremlinQueryStatusAttributes

Contains status components of a Gremlin query.

Contents

attributes

Attributes of the Gremlin query status.

Type: JSON value

Required: No

code

The HTTP response code returned fro the Gremlin query request..

Type: Integer

Required: No

message

The status message.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LoaderIdResult

Contains a list of load IDs.

Contents

loadIds

A list of load IDs.

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MlConfigDefinition

Contains a Neptune ML configuration.

Contents

arn

The ARN for the configuration.

Type: String

Required: No

name

The configuration name.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MlResourceDefinition

Defines a Neptune ML resource.

Contents

arn

The resource ARN.

Type: String

Required: No

cloudwatchLogUrl

The CloudWatch log URL for the resource.

Type: String

Required: No

failureReason

The failure reason, in case of a failure.

Type: String

Required: No

name

The resource name.

Type: String

Required: No

outputLocation

The output location.

Type: String

Required: No

status

The resource status.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NodeStructure

A node structure.

Contents

count

Number of nodes that have this specific structure.

Type: Long

Required: No

distinctOutgoingEdgeLabels

A list of distinct outgoing edge labels present in this specific structure.

Type: Array of strings

Required: No

nodeProperties

A list of the node properties present in this specific structure.

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PropertygraphData

A Gremlin or openCypher change record.

Contents

id

The ID of the Gremlin or openCypher element.

Type: String

Required: Yes

key

The property name. For element labels, this is `label`.

Type: String

Required: Yes

type

The type of this Gremlin or openCypher element. Must be one of:

- **v1** - Vertex label for Gremlin, or node label for openCypher.
- **vp** - Vertex properties for Gremlin, or node properties for openCypher.
- **e** - Edge and edge label for Gremlin, or relationship and relationship type for openCypher.
- **ep** - Edge properties for Gremlin, or relationship properties for openCypher.

Type: String

Required: Yes

value

This is a JSON object that contains a value field for the value itself, and a datatype field for the JSON data type of that value:

Type: JSON value

Required: Yes

from

If this is an edge (type = `e`), the ID of the corresponding `from` vertex or source node.

Type: String

Required: No

to

If this is an edge (type = `e`), the ID of the corresponding `to` vertex or target node.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PropertygraphRecord

Structure of a property graph record.

Contents

commitTimestamp

The time at which the commit for the transaction was requested, in milliseconds from the Unix epoch.

Type: Long

Required: Yes

data

The serialized Gremlin or openCypher change record.

Type: PropertygraphData object

Required: Yes

eventId

The sequence identifier of the stream change record.

Type: String to string map

Required: Yes

op

The operation that created the change.

Type: String

Required: Yes

isLastOp

Only present if this operation is the last one in its transaction. If present, it is set to true. It is useful for ensuring that an entire transaction is consumed.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PropertygraphSummary

The graph summary API returns a read-only list of node and edge labels and property keys, along with counts of nodes, edges, and properties. See Graph summary response for a property graph (PG).

Contents

edgeLabels

A list of the distinct edge labels in the graph.

Type: Array of strings

Required: No

edgeProperties

A list of the distinct edge properties in the graph, along with the count of edges where each property is used.

Type: Array of string to long maps

Required: No

edgeStructures

This field is only present when the requested mode is `DETAILED`. It contains a list of edge structures.

Type: Array of EdgeStructure objects

Required: No

nodeLabels

A list of the distinct node labels in the graph.

Type: Array of strings

Required: No

nodeProperties

The number of distinct node properties in the graph.

Type: Array of string to long maps

Required: No

nodeStructures

This field is only present when the requested mode is `DETAILED`. It contains a list of node structures.

Type: Array of NodeStructure objects

Required: No

numEdgeLabels

The number of distinct edge labels in the graph.

Type: Long

Required: No

numEdgeProperties

The number of distinct edge properties in the graph.

Type: Long

Required: No

numEdges

The number of edges in the graph.

Type: Long

Required: No

numNodeLabels

The number of distinct node labels in the graph.

Type: Long

Required: No

numNodeProperties

A list of the distinct node properties in the graph, along with the count of nodes where each property is used.

Type: Long

Required: No

numNodes

The number of nodes in the graph.

Type: Long

Required: No

totalEdgePropertyValues

The total number of usages of all edge properties.

Type: Long

Required: No

totalNodePropertyValues

The total number of usages of all node properties.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PropertygraphSummaryValueMap

Payload for the property graph summary response.

Contents

graphSummary

The graph summary.

Type: PropertygraphSummary object

Required: No

lastStatisticsComputationTime

The timestamp, in ISO 8601 format, of the time at which Neptune last computed statistics.

Type: Timestamp

Required: No

version

The version of this graph summary response.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

QueryEvalStats

Structure to capture query statistics such as how many queries are running, accepted or waiting and their details.

Contents

cancelled

Set to TRUE if the query was cancelled, or FALSE otherwise.

Type: Boolean

Required: No

elapsed

The number of milliseconds the query has been running so far.

Type: Integer

Required: No

subqueries

The number of subqueries in this query.

Type: JSON value

Required: No

waited

Indicates how long the query waited, in milliseconds.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

QueryLanguageVersion

Structure for expressing the query language version.

Contents

version

The version of the query language.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RDFGraphSummary

The RDF graph summary API returns a read-only list of classes and predicate keys, along with counts of quads, subjects, and predicates.

Contents

classes

A list of the classes in the graph.

Type: Array of strings

Required: No

numClasses

The number of classes in the graph.

Type: Long

Required: No

numDistinctPredicates

The number of distinct predicates in the graph.

Type: Long

Required: No

numDistinctSubjects

The number of distinct subjects in the graph.

Type: Long

Required: No

numQuads

The number of quads in the graph.

Type: Long

Required: No

predicates

"A list of predicates in the graph, along with the predicate counts.

Type: Array of string to long maps

Required: No

subjectStructures

This field is only present when the request mode is `DETAILED`. It contains a list of subject structures.

Type: Array of SubjectStructure objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RDFGraphSummaryValueMap

Payload for an RDF graph summary response.

Contents

graphSummary

The graph summary of an RDF graph. See Graph summary response for an RDF graph.

Type: RDFGraphSummary object

Required: No

lastStatisticsComputationTime

The timestamp, in ISO 8601 format, of the time at which Neptune last computed statistics.

Type: Timestamp

Required: No

version

The version of this graph summary response.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RefreshStatisticsIdMap

Statistics for REFRESH mode.

Contents

statisticsId

The ID of the statistics generation run that is currently occurring.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SparqlData

Neptune logs are converted to SPARQL quads in the graph using the Resource Description Framework (RDF) N-QUADS language defined in the W3C RDF 1.1 N-Quads specification

Contents

stmt

Holds an N-QUADS statement expressing the changed quad.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SparqlRecord

A serialized SPARQL stream record capturing a change-log entry for the RDF graph.

Contents

commitTimestamp

The time at which the commit for the transaction was requested, in milliseconds from the Unix epoch.

Type: Long

Required: Yes

data

The serialized SPARQL change record. The serialization formats of each record are described in more detail in Serialization Formats in Neptune Streams.

Type: SparqlData object

Required: Yes

eventId

The sequence identifier of the stream change record.

Type: String to string map

Required: Yes

op

The operation that created the change.

Type: String

Required: Yes

isLastOp

Only present if this operation is the last one in its transaction. If present, it is set to true. It is useful for ensuring that an entire transaction is consumed.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Statistics

Contains statistics information. The DFE engine uses information about the data in your Neptune graph to make effective trade-offs when planning query execution. This information takes the form of statistics that include so-called characteristic sets and predicate statistics that can guide query planning. See Managing statistics for the Neptune DFE to use.

Contents

active

Indicates whether or not DFE statistics generation is enabled at all.

Type: Boolean

Required: No

autoCompute

Indicates whether or not automatic statistics generation is enabled.

Type: Boolean

Required: No

date

The UTC time at which DFE statistics have most recently been generated.

Type: Timestamp

Required: No

note

A note about problems in the case where statistics are invalid.

Type: String

Required: No

signatureInfo

A StatisticsSummary structure that contains:

- `signatureCount` - The total number of signatures across all characteristic sets.
- `instanceCount` - The total number of characteristic-set instances.
- `predicateCount` - The total number of unique predicates.

Type: StatisticsSummary object

Required: No

statisticsId

Reports the ID of the current statistics generation run. A value of -1 indicates that no statistics have been generated.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StatisticsSummary

Information about the characteristic sets generated in the statistics.

Contents

instanceCount

The total number of characteristic-set instances.

Type: Integer

Required: No

predicateCount

The total number of unique predicates.

Type: Integer

Required: No

signatureCount

The total number of signatures across all characteristic sets.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SubjectStructure

A subject structure.

Contents

count

Number of occurrences of this specific structure.

Type: Long

Required: No

predicates

A list of predicates present in this specific structure.

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 403

ExpiredTokenException

The security token included in the request is expired

HTTP Status Code: 403

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 403

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

MalformedHttpRequestException

Problems with the request at the HTTP level, e.g. we can't decompress the body according to the decompression algorithm specified by the content-encoding.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 401

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestAbortedException

Convenient exception that can be used when a request is aborted before a reply is sent back (e.g. client closed connection).

HTTP Status Code: 400

RequestEntityTooLargeException

Problems with the request at the HTTP level. The request entity is too large.

HTTP Status Code: 413

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

RequestTimeoutException

Problems with the request at the HTTP level. Reading the Request timed out.

HTTP Status Code: 408

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

UnrecognizedClientException

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

UnknownOperationException

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 404

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400