

Azure Cosmos DB support for Node.js source code

- [Objects](#)
- [What results can you expect?](#)
 - [Cosmos DB client creation](#)
 - [Database and collection creation](#)
 - [useSelectLink](#)
 - [useDeleteLink](#)



CAST supports **Azure Cosmos DB** via its [Node.js](#) extension. Details about how this support is provided for **Node.js source code** is discussed below.

Objects

Icon	Description
	Node.js Azure CosmosDB Client
	Node.js Azure CosmosDB Database
	Node.js Azure CosmosDB Collection
	Node.js Azure CosmosDB Unknown Client
	Node.js Azure CosmosDB Unknown Database
	Node.js CosmosDB Unknown Collection

What results can you expect?

Once the analysis/snapshot generation has completed, you can view the results in the normal manner (for example via CAST Enlighten). Some examples are shown below.

Cosmos DB client creation

```
const CosmosClient = require("@azure/cosmos");
const endpoint = '127.0.0.1:8060'

const client = new CosmosClient({ endpoint: endpoint, auth: { masterKey } });
```

Database and collection creation

```

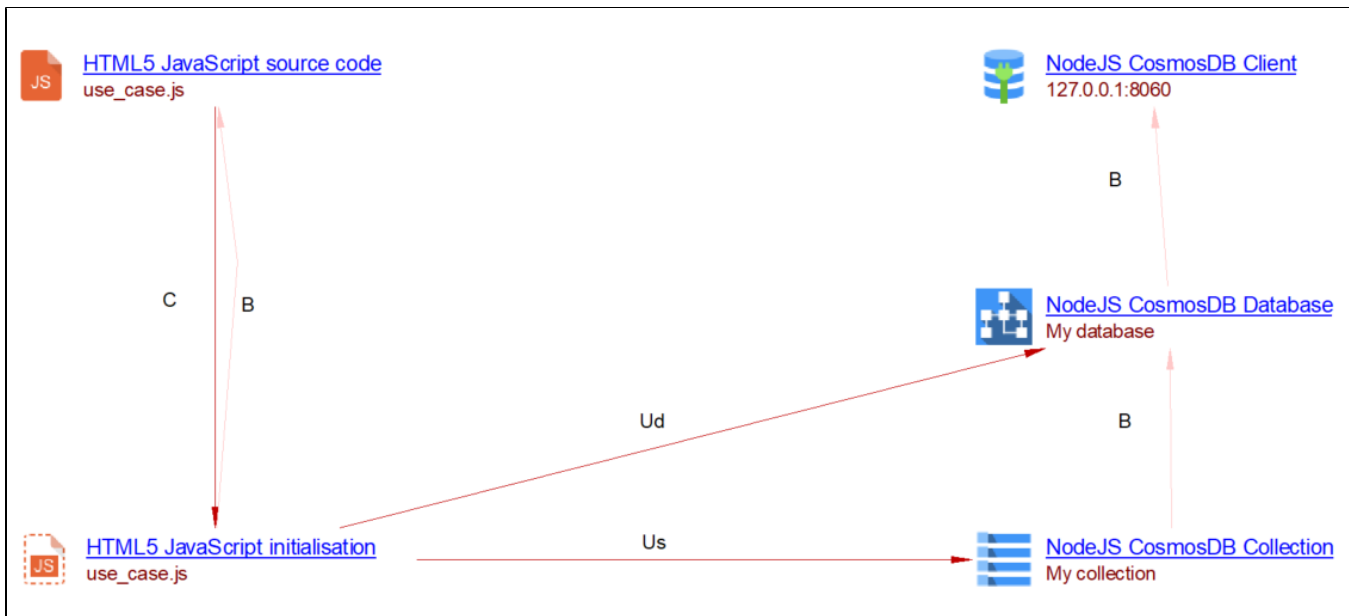
...

const databaseId = 'My database'
const collectionId = 'My collection'

// CREATE DATABASE
const dbResponse = client.databases.createIfNotExists({
  id: databaseId
});
database = dbResponse.database;

// CREATE COLLECTION
const coResponse = database.containers.createIfNotExists({
  id: collectionId
});
container = coResponse.container;

```



useSelectLink

These declarations create a *useSelectLink* from the source code to the collection

```

// QUERY
var param = {
  query: "SELECT * FROM root r WHERE r.completed=@completed",
  parameters: [
    {
      name: '@completed',
      value: false
    }
  ]
};

const { result: results } = container.items
  .query(param)
  .toArray();

```

useDeleteLink

This declaration create a *useDeleteLink* from the source code to the database

```
client.database(databaseId).delete()
```