100 XP

Explore supported APIs

3 minutes

Azure Cosmos DB offers multiple database APIs, which include:

- Azure Cosmos DB for NoSQL
- Azure Cosmos DB for MongoDB
- Azure Cosmos DB for PostgreSQL
- Azure Cosmos DB for Apache Cassandra
- Azure Cosmos DB for Table
- Azure Cosmos DB for Apache Gremlin

By using these APIs, you can model real world data using documents, key-value, graph, and column-family data models. These APIs allow your applications to treat Azure Cosmos DB as if it were various other databases technologies, without the overhead of management, and scaling approaches.

Considerations when choosing an API

API for NoSQL is native to Azure Cosmos DB.

API for MongoDB, PostgreSQL, Cassandra, Gremlin, and Table implement the wire protocol of open-source database engines. These APIs are best suited if the following conditions are true:

- If you have existing MongoDB, PostgreSQL Cassandra, or Gremlin applications
- If you don't want to rewrite your entire data access layer
- If you want to use the open-source developer ecosystem, client-drivers, expertise, and resources for your database

API for NoSQL

The Azure Cosmos DB API for NoSQL stores data in document format. It offers the best end-to-end experience as we have full control over the interface, service, and the SDK client libraries. Any new feature that is rolled out to Azure Cosmos DB is first available on API for NoSQL accounts. NoSQL accounts provide support for querying items using the Structured Query Language (SQL) syntax.

API for MongoDB

The Azure Cosmos DB API for MongoDB stores data in a document structure, via BSON format. It's compatible with MongoDB wire protocol; however, it doesn't use any native MongoDB related code. The API for MongoDB is a great choice if you want to use the broader MongoDB ecosystem and skills, without compromising on using Azure Cosmos DB features.

API for PostgreSQL

Azure Cosmos DB for PostgreSQL is a managed service for running PostgreSQL at any scale, with the Citus open source superpower of distributed tables. It stores data either on a single node, or distributed in a multi-node configuration.

API for Apache Cassandra

The Azure Cosmos DB API for Cassandra stores data in column-oriented schema. Apache Cassandra offers a highly distributed, horizontally scaling approach to storing large volumes of data while offering a flexible approach to a column-oriented schema. API for Cassandra in Azure Cosmos DB aligns with this philosophy to approaching distributed NoSQL databases. This API for Cassandra is wire protocol compatible with native Apache Cassandra.

API for Apache Gremlin

The Azure Cosmos DB API for Gremlin allows users to make graph queries and stores data as edges and vertices.

Use the API for Gremlin for scenarios:

- Involving dynamic data
- Involving data with complex relations
- Involving data that is too complex to be modeled with relational databases
- If you want to use the existing Gremlin ecosystem and skills

API for Table

The Azure Cosmos DB API for Table stores data in key/value format. If you're currently using Azure Table storage, you may see some limitations in latency, scaling, throughput, global distribution, index management, low query performance. API for Table overcomes these limitations and it's recommended to migrate your app if you want to use the benefits of Azure Cosmos DB. API for Table only supports OLTP scenarios.

Next unit: Discover request units

Continue >