# Introduction to DocumentDB: A NoSQL JSON Database

Azure DocumentDB is a fully managed NoSQL database service built for fast and predictable performance, high availability, automatic scaling, and ease of development. Its flexible data model, consistent low latencies, and rich query capabilities make it a great fit for web, mobile, gaming, and IoT, and many other applications that need seamless scale.

A quick way to learn about this JSON database and see it in action is to follow these three steps:

1. Watch the two minute [What is DocumentDB?](/documentation/videos/what-is-azure-documentdb/) video, which introduces the benefits of using DocumentDB.
2. Watch the three minute [Create DocumentDB on Azure](/documentation/videos/create-documentdb-on-azure/) video, which highlights how to get started with DocumentDB by using the Azure Portal.
3. Visit the [Query Playground](http://www.documentdb.com/sql/demo), where you can walk through different activities to learn about the rich querying functionality available in DocumentDB. Then, head over to the Sandbox tab and run your own custom SQL queries and experiment with DocumentDB.

Then, return to this article, where we’ll dig in deeper and you’ll learn the answers to the following questions:

* [What is DocumentDB and what value does it provide to modern applications?](#what-is-azure-documentdb)
* [How is my data managed in DocumentDB and how do I access it?](#data-management)
* [How do I develop applications using DocumentDB?](#develop)
* [What are my next steps to build a DocumentDB application?](#next-steps)

## What is Azure DocumentDB?

Modern applications produce, consume and respond quickly to very large volumes of data. These applications evolve very rapidly and so does the underlying data schema. In response to this, developers have increasingly chosen schema-free NoSQL document databases as simple, fast, scalable solutions to store and process data while preserving the ability to quickly iterate over application data models and unstructured data feeds. However, many schema-free databases do not allow for complex queries and transactional processing, making advanced data management difficult. This is where DocumentDB comes in. Microsoft developed DocumentDB to fulfill these requirements when managing data for today’s applications.

DocumentDB is a true schema-free NoSQL database service designed for modern mobile, web, gaming, and IoT applications. DocumentDB delivers consistently fast reads and writes, schema flexibility, and the ability to easily scale a database up and down on demand. It does not assume or require any schema for the JSON documents it indexes. By default, it automatically indexes all the documents in the database and does not expect or require any schema or creation of secondary indices. DocumentDB enables complex ad hoc queries using a SQL language, supports well defined consistency levels, and offers JavaScript language integrated, multi-document transaction processing using the familiar programming model of stored procedures, triggers, and UDFs.

As a JSON database, DocumentDB natively supports JSON documents enabling easy iteration of application schema, and support applications that need key-value, document or tabular data models. DocumentDB embraces the ubiquity of JSON and JavaScript, eliminating mismatch between application defined objects and database schema. Deep integration of JavaScript also allows developers to execute application logic efficiently and directly - within the database engine in a database transaction.

Azure DocumentDB offers the following key capabilities and benefits:

* **Elastically scalable throughput and storage:** Easily scale up or scale down your DocumentDB JSON database to meet your application needs. Your data is stored on solid state disks (SSD) for low predictable latencies. DocumentDB supports containers for storing JSON data called collections which can scale to virtually unlimited storage sizes and provioned throughput. You can elastically scale DocumentDB with predictable performance seamlessly as your application grows.
* **Ad hoc queries with familiar SQL syntax:** Store heterogeneous JSON documents within DocumentDB and query these documents through a familiar SQL syntax. DocumentDB utilizes a highly concurrent, lock free, log structured indexing technology to automatically index all document content. This enables rich real-time queries without the need to specify schema hints, secondary indexes, or views. Learn more in [Query DocumentDB](documentdb-sql-query.md).
* **JavaScript execution within the database:** Express application logic as stored procedures, triggers, and user defined functions (UDFs) using standard JavaScript. This allows your application logic to operate over data without worrying about the mismatch between the application and the database schema. DocumentDB provides full transactional execution of JavaScript application logic directly inside the database engine. The deep integration of JavaScript enables the execution of INSERT, REPLACE, DELETE, and SELECT operations from within a JavaScript program as an isolated transaction. Learn more in [DocumentDB server-side programming](documentdb-programming.md).
* **Tunable consistency levels:** Select from four well defined consistency levels to achieve optimal trade-off between consistency and performance. For queries and read operations, DocumentDB offers four distinct consistency levels: strong, bounded-staleness, session, and eventual. These granular, well-defined consistency levels allow you to make sound trade-offs between consistency, availability, and latency. Learn more in [Using consistency levels to maximize availability and performance in DocumentDB](documentdb-consistency-levels.md).
* **Fully managed:** Eliminate the need to manage database and machine resources. As a fully-managed Microsoft Azure service, you do not need to manage virtual machines, deploy and configure software, manage scaling, or deal with complex data-tier upgrades. Every database is automatically backed up and protected against regional failures. You can easily add a DocumentDB account and provision capacity as you need it, allowing you to focus on your application instead of operating and managing your database.
* **Open by design:** Get started quickly by using existing skills and tools. Programming against DocumentDB is simple, approachable, and does not require you to adopt new tools or adhere to custom extensions to JSON or JavaScript. You can access all of the database functionality including CRUD, query, and JavaScript processing over a simple RESTful HTTP interface. DocumentDB embraces existing formats, languages, and standards while offering high value database capabilities on top of them.

You can use DocumentDB to store flexible datasets that require query retrieval and transactional processing. Application scenarios may include user data for interactive web, mobile, and gaming applications as well as storage, retrieval, and processing of IoT device generated JSON data. A database can store any volume of JSON documents, as DocumentDB is well suited for applications that run at scale on the internet.

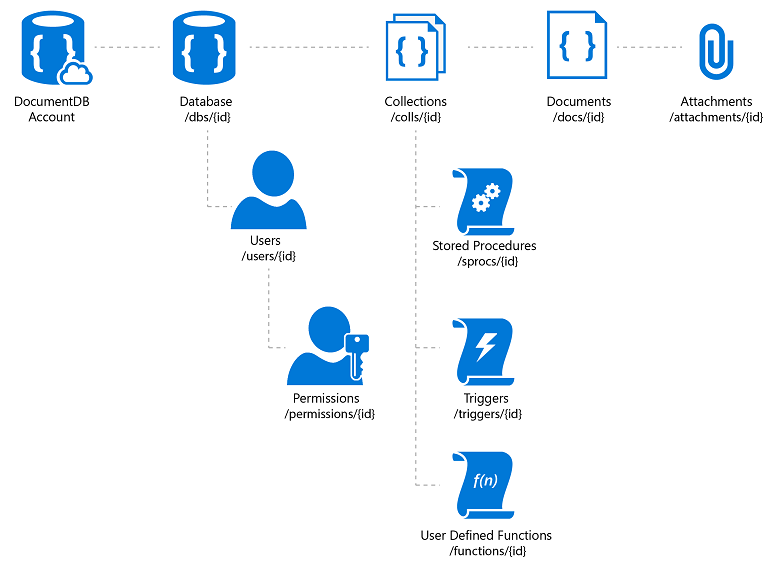
## Azure DocumentDB resources

Azure DocumentDB manages data through well-defined database resources. These resources are replicated for high availability and are uniquely addressable by their logical URI. DocumentDB offers a simple HTTP based RESTful programming model for all resources.

The DocumentDB database account is a unique namespace that gives you access to Azure DocumentDB. Before you can create a database account, you must have an Azure subscription, which gives you access to a variety of Azure services.

All resources within DocumentDB are modeled and stored as JSON documents. Resources are managed as items, which are JSON documents containing metadata, and as feeds which are collections of items. Sets of items are contained within their respective feeds.

The image below shows the relationships between the DocumentDB resources:



The hierarchical relationship between resources in DocumentDB, a NoSQL JSON database

A database account consists of a set of databases, each containing multiple collections, each of which can contain stored procedures, triggers, UDFs, documents, and related attachments. A database also has associated users, each with a set of permissions to access various other collections, stored procedures, triggers, UDFs, documents, or attachments. While databases, users, permissions, and collections are system-defined resources with well-known schemas - documents, stored procedures, triggers, UDFs, and attachments contain arbitrary, user defined JSON content.

## Developing with Azure DocumentDB

Azure DocumentDB exposes resources through a REST API that can be called by any language capable of making HTTP/HTTPS requests. Additionally, DocumentDB offers programming libraries for several popular languages. These libraries simplify many aspects of working with Azure DocumentDB by handling details such as address caching, exception management, automatic retries and so forth. Libraries are currently available for the following languages and platforms:

|  |  |
| --- | --- |
| Download | Documentation |
| [.NET SDK](http://go.microsoft.com/fwlink/?LinkID=402989) | [.NET library](https://msdn.microsoft.com/library/azure/dn948556.aspx) |
| [Node.js SDK](http://go.microsoft.com/fwlink/?LinkID=402990) | [Node.js library](http://azure.github.io/azure-documentdb-node/) |
| [Java SDK](http://go.microsoft.com/fwlink/?LinkID=402380) | [Java library](http://azure.github.io/azure-documentdb-java/) |
| [JavaScript SDK](http://go.microsoft.com/fwlink/?LinkID=402991) | [JavaScript library](http://azure.github.io/azure-documentdb-js/) |
| n/a | [Server-side JavaScript SDK](http://azure.github.io/azure-documentdb-js-server/) |
| [Python SDK](https://pypi.python.org/pypi/pydocumentdb) | [Python library](http://azure.github.io/azure-documentdb-python/) |

Beyond basic create, read, update, and delete operations, DocumentDB provides a rich SQL query interface for retrieving JSON documents and server side support for transactional execution of JavaScript application logic. The query and script execution interfaces are available through all platform libraries as well as the REST APIs.

### SQL query

Azure DocumentDB supports querying documents using a SQL language, which is rooted in the JavaScript type system, and expressions with support for relational, hierarchical, and spatial queries. The DocumentDB query language is a simple yet powerful interface to query JSON documents. The language supports a subset of ANSI SQL grammar and adds deep integration of JavaScript object, arrays, object construction, and function invocation. DocumentDB provides its query model without any explicit schema or indexing hints from the developer.

User Defined Functions (UDFs) can be registered with DocumentDB and referenced as part of a SQL query, thereby extending the grammar to support custom application logic. These UDFs are written as JavaScript programs and executed within the database.

For .NET developers, DocumentDB also offers a LINQ query provider as part of the [.NET SDK](https://msdn.microsoft.com/library/azure/microsoft.azure.documents.linq.aspx).

### Transactions and JavaScript execution

DocumentDB allows you to write application logic as named programs written entirely in JavaScript. These programs are registered for a collection and can issue database operations on the documents within a given collection. JavaScript can be registered for execution as a trigger, stored procedure or user defined function. Triggers and stored procedures can create, read, update, and delete documents whereas user defined functions execute as part of the query execution logic without write access to the collection.

JavaScript execution within DocumentDB is modeled after the concepts supported by relational database systems, with JavaScript as a modern replacement for Transact-SQL. All JavaScript logic is executed within an ambient ACID transaction with snapshot isolation. During the course of its execution, if the JavaScript throws an exception, then the entire transaction is aborted.

## Next steps

If you already have an Azure account, you can get started with DocumentDB in the [Azure Portal](https://portal.azure.com/#gallery/Microsoft.DocumentDB) by [creating a DocumentDB database account](documentdb-create-account.md).

If you don’t have an Azure account, you can:

* Sign up for an [Azure free trial](https://azure.microsoft.com/pricing/free-trial/), which gives you 30 days and $200 to try all the Azure services.
* If you have an MSDN subscription, you are eligible for [$150 in free Azure credits per month](https://azure.microsoft.com/pricing/member-offers/msdn-benefits-details/) to use on any Azure service.

Then, when you’re ready to learn more, visit our [learning path](https://azure.microsoft.com/documentation/learning-paths/documentdb/) to navigate all the learning resources available to you.