

Azure Storage Table Functionality

Janne Kemppe
2015



opiframe

Azure Table Storage Functionality

- * **Azure Storage Tables are arranged according to following.**
- * Azure Storage tables use dynamic storage system.
 - * 1) **Storage Account** can have multiple Tables.
 - * 2) **Table** can have multiple Entities.
 - * 3) **Entity** can have maximum of 255 **Properties**, each of which is a key-value pair. Entities are fully dynamic and they do not need to have same properties.
- * Storage table is effectively a spreadsheet where Entity equals one row and property equals one cell.

Azure Table Storage Functionality

- * **Azure Table Storage is not relational database table.**
- * See information:
 - * <https://msdn.microsoft.com/en-us/library/azure/jj553018.aspx>
- * Major difference (to Azure SQL) is in capacity and quotas section:
 - * Maximum entity size is 1 megabyte
 - * Maximum data size is 200 Terabytes
 - * Azure SQL has 150 gigabytes

Azure Table Storage Functionality

- * **Azure Storage Table's major advantage is in storing huge number of events.**
- * Storage Tables are typically used for logging (write and forget persistently) and audit tables.
- * Azure Tables have limitation that maximum number of top-level dynamic objects is 1000 in a single request.
 - * This limit is due performance reasons.
 - * There is a workaround by using a continuation token which allows fetching data as 1000 entity blocks and then continuing fetching next batch using token as a startup point.

Azure Table Storage Functionality

- * **Each entity requires three properties.**
- * Each entity reserves three properties:
 - * PartitionKey
 - * RowKey
 - * Timestamp (automatically added)
- * (There is also fourth called ETag, which is reserved for optimistic concurrency. This means second user is informed that first user has changed something in same row.)

Azure Table Storage Functionality

- * **PartitioningKey** is intended to be used for Partitioning / "Sharding".
- * The logic is that **Partition Key** can divide large table to specific major shards, for instance Users (as user identifier).
- * The **Row Key** is then used to identify particular events belonging that that partition.
- * **Time Stamp** is used to make sure changes can be traced to order of execution.
- * See also:
 - * [https://en.wikipedia.org/wiki/Shard_\(database_architecture\)](https://en.wikipedia.org/wiki/Shard_(database_architecture))
 - * [https://en.wikipedia.org/wiki/Partition_\(database\)](https://en.wikipedia.org/wiki/Partition_(database))

- * **Azure Storage Tables support two main design patterns:**
- * Study following patterns when determining data retrieval architecture:
 - * Command and Query Responsibility Segregation Pattern:
 - * <https://msdn.microsoft.com/en-us/library/dn568103.aspx>
 - * Event Sourcing Pattern:
 - * <https://msdn.microsoft.com/en-us/library/dn589792.aspx>

Azure Table Storage Functionality

- * **Tables are typically accessed through Azure Web apps but this is not a requirement.**
- * Most common method of accessing Azure Storage Tables is through Azure Storage Client Library which is effectively a wrapper around RESTful Azure Management API.

Azure Table Storage Functionality

- * Typical scenario is following:
 - * 1) Create a connection to the storage account. This typically means setting a connection string.
 - * 2) Create a connection to a table/create a new table.
 - * 3) Insert entities as individually or as a batch insertion (cheaper).
 - * Typically one entity is one object of a class.
 - * 4) All commands use `TableQuery<T>`
- * Additional information can be found in:
 - * <https://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-tables/>

Azure Table Storage Functionality

- * Look also additional information how `TableQuery<T>` based query is actually executed using Odata
 - * <https://msdn.microsoft.com/en-us/library/azure/dd894031.aspx>
 - * <http://www.odata.org/documentation/odata-version-2-0/uri-conventions/>

Azure Table Storage Functionality

- * Ohjattu harjoitus: Start using Storage Table using a quick example.

Questions?