Definition

db.createCollection**(name, options)**

*Changed in version 3.4:*Added support for:

* Creation of views (see also [db.createView()](https://docs.mongodb.com/manual/reference/method/db.createView/#db.createView)).
* Collation.

Creates a new collection or [view](https://docs.mongodb.com/manual/core/views/).

Because MongoDB creates a collection implicitly when the collection is first referenced in a command, this method is used primarily for creating new collections that use specific options. For example, you usedb.createCollection() to create a [capped collection](https://docs.mongodb.com/manual/reference/glossary/#term-capped-collection), or to create a new collection that uses[document validation](https://docs.mongodb.com/manual/core/schema-validation/). db.createCollection() is also used to pre-allocate space for an ordinary collection.

db.createCollection() is a wrapper around the database command [create](https://docs.mongodb.com/manual/reference/command/create/#dbcmd.create).

The db.createCollection() method has the following prototype form:

db.createCollection(<name>, { capped: <**boolean**>,

autoIndexId: <**boolean**>,

size: <number>,

max: <number>,

storageEngine: <document>,

validator: <document>,

validationLevel: <string>,

validationAction: <string>,

indexOptionDefaults: <document>,

viewOn: <string>,

pipeline: <pipeline>,

collation: <document> } )

The db.createCollection() method has the following parameters:

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| name | string | The name of the collection to create. |
| options | document | Optional. Configuration options for creating a capped collection, for preallocating space in a new collection, or for creating a view. |

The options document contains the following fields:

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| capped | boolean | Optional. To create a [capped collection](https://docs.mongodb.com/manual/reference/glossary/#term-capped-collection), specify true. If you specify true, you must also set a maximum size in the size field. |
| autoIndexId | boolean | Optional. Specify false to disable the automatic creation of an index on the \_id field.  **IMPORTANT**  For replica sets, do not set autoIndexId to false.  *Deprecated since version 3.2.* |
| size | number | Optional. Specify a maximum size in bytes for a capped collection. Once a capped collection reaches its maximum size, MongoDB removes the older documents to make space for the new documents. The size field is required for capped collections and ignored for other collections. |
| max | number | Optional. The maximum number of documents allowed in the capped collection. The size limit takes precedence over this limit. If a capped collection reaches the size limit before it reaches the maximum number of documents, MongoDB removes old documents. If you prefer to use the max limit, ensure that the size limit, which is required for a capped collection, is sufficient to contain the maximum number of documents. |
| usePowerOf2Sizes | boolean | Optional. Available for the MMAPv1 storage engine only.  *Deprecated since version 3.0:*For the MMAPv1 storage engine, all collections use the [power of 2 sizes allocation](https://docs.mongodb.com/manual/core/mmapv1/#power-of-2-allocation) unless the noPaddingoption is true. The usePowerOf2Sizes option does not affect the allocation strategy. |
| noPadding | boolean | Optional. Available for the MMAPv1 storage engine only.  *New in version 3.0:*noPadding flag disables the [power of 2 sizes allocation](https://docs.mongodb.com/manual/core/mmapv1/#power-of-2-allocation) for the collection. With noPadding flag set to true, the allocation strategy does not include additional space to accommodate document growth, as such, document growth will result in new allocation. Use for collections with workloads that are insert-only or in-place updates (such as incrementing counters).  Defaults to false.  **WARNING**  Do not set noPadding if the workload includes removes or any updates that may cause documents to grow. For more information, see [No Padding Allocation Strategy](https://docs.mongodb.com/manual/core/mmapv1/#exact-fit-allocation). |
| storageEngine | document | Optional. Available for the WiredTiger storage engine only.  *New in version 3.0.*  Allows users to specify configuration to the storage engine on a per-collection basis when creating a collection. The value of thestorageEngine option should take the following form:  { <storage-engine-name>: <options> }  Storage engine configuration specified when creating collections are validated and logged to the [oplog](https://docs.mongodb.com/manual/reference/glossary/#term-oplog) during replication to support replica sets with members that use different storage engines. |
| validator | document | Optional. Allows users to specify [validation rules or expressions](https://docs.mongodb.com/manual/core/schema-validation/) for the collection. For more information, see [Schema Validation](https://docs.mongodb.com/manual/core/schema-validation/).  *New in version 3.2.*  The validator option takes a document that specifies the validation rules or expressions. You can specify the expressions using the same operators as the [query operators](https://docs.mongodb.com/manual/reference/operator/query/#query-selectors) with the exception of $geoNear, [$near](https://docs.mongodb.com/manual/reference/operator/query/near/#op._S_near), [$nearSphere](https://docs.mongodb.com/manual/reference/operator/query/nearSphere/#op._S_nearSphere), [$text](https://docs.mongodb.com/manual/reference/operator/query/text/#op._S_text), and [$where](https://docs.mongodb.com/manual/reference/operator/query/where/#op._S_where).  **NOTE**   * Validation occurs during updates and inserts. Existing documents do not undergo validation checks until modification. * You cannot specify a validator for collections in the admin,local, and config databases. * You cannot specify a validator for system.\* collections. |
| validationLevel | string | Optional. Determines how strictly MongoDB applies the validation rules to existing documents during an update.  *New in version 3.2.*   | **validationLevel** | **Description** | | --- | --- | | "off" | No validation for inserts or updates. | | "strict" | **Default** Apply validation rules to all inserts and all updates. | | "moderate" | Apply validation rules to inserts and to updates on existing *valid*documents. Do not apply rules to updates on existing *invalid*documents. | |
| validationAction | string | Optional. Determines whether to error on invalid documents or just warn about the violations but allow invalid documents to be inserted.  *New in version 3.2.*  **IMPORTANT**  Validation of documents only applies to those documents as determined by the validationLevel.   | **validationAction** | **Description** | | --- | --- | | "error" | **Default** Documents must pass validation before the write occurs. Otherwise, the write operation fails. | | "warn" | Documents do not have to pass validation. If the document fails validation, the write operation logs the validation failure. | |
| indexOptionDefaults | document | Optional. Allows users to specify a default configuration for indexes when creating a collection.  The indexOptionDefaults option accepts a storageEnginedocument, which should take the following form:  { <storage-engine-name>: <options> }  Storage engine configuration specified when creating indexes are validated and logged to the [oplog](https://docs.mongodb.com/manual/reference/glossary/#term-oplog) during replication to support replica sets with members that use different storage engines.  *New in version 3.2.* |
| viewOn | string | The name of the source collection or view from which to create the view. The name is not the full namespace of the collection or view; i.e. does not include the database name and implies the same database as the view to create.  See also [db.createView()](https://docs.mongodb.com/manual/reference/method/db.createView/#db.createView).  *New in version 3.4.* |
| pipeline | array | An array that consists of the [aggregation pipeline stage](https://docs.mongodb.com/manual/core/aggregation-pipeline/#id1).[db.createView](https://docs.mongodb.com/manual/reference/method/db.createView/#db.createView) creates the view by applying the specified pipeline to the viewOn collection or view.  The view definition is public; i.e. [db.getCollectionInfos()](https://docs.mongodb.com/manual/reference/method/db.getCollectionInfos/#db.getCollectionInfos) and explain operations on the view will include the pipeline that defines the view. As such, avoid referring directly to sensitive fields and values in view definitions.  See also [db.createView()](https://docs.mongodb.com/manual/reference/method/db.createView/#db.createView).  *New in version 3.4.* |
| collation | document | Specifies the default [collation](https://docs.mongodb.com/manual/reference/bson-type-comparison-order/#collation) for the collection.  [Collation](https://docs.mongodb.com/manual/reference/collation/) allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks.  The collation option has the following syntax:  collation: {  locale: <string>,  caseLevel: <**boolean**>,  caseFirst: <string>,  strength: <**int**>,  numericOrdering: <**boolean**>,  alternate: <string>,  maxVariable: <string>,  backwards: <**boolean**>  }  When specifying collation, the locale field is mandatory; all other collation fields are optional. For descriptions of the fields, see [Collation Document](https://docs.mongodb.com/manual/reference/collation/#collation-document-fields).  If you specify a collation at the collection level:   * Indexes on that collection will be created with that collation unless the index creation operation explicitly specify a different collation. * Operations on that collection use the collection’s default collation unless they explicitly specify a different collation.   You cannot specify multiple collations for an operation. For example, you cannot specify different collations per field, or if performing a find with a sort, you cannot use one collation for the find and another for the sort.  If no collation is specified for the collection or for the operations, MongoDB uses the simple binary comparison used in prior versions for string comparisons.  For a collection, you can only specify the collation during the collection creation. Once set, you cannot modify the collection’s default collation. |