
PSCouchDB Documentation

Matteo Guadrini

Dec 15, 2020

Contents:

1	Introduction	3
1.1	Installation	3
1.2	Signing	3
1.3	Start	4
2	Setup	5
2.1	Single Node	5
2.2	Cluster	5
3	Nodes	7
3.1	Add a node	7
3.2	Remove a node	7
4	Configuration	9
4.1	Get configuration	9
4.2	Modify configuration	9
5	Permission	11
5.1	Admin Party	11
5.2	Create Admin user	11
5.3	Members access	12
5.4	Read only access	12
5.5	Limit write access	12
5.6	Revoke database permissions	12
5.7	Remove an admin	13
5.8	Remove a user	13
5.9	Reset user password	13
5.10	Reset admin password	13
6	Server operation	15
6.1	Read the log	16
6.2	Clear the log	17
7	Replication	19
7.1	Get replica	19
7.2	Create replica	19
7.3	Modify replica	20

7.4	Remove replica	20
7.5	Replication request	20
8	Authentication	21
9	Databases	23
9.1	Test a database	23
9.2	Read a database	23
9.3	Create a database	24
9.4	Remove a database	24
9.5	Copy a database	24
9.6	Index	24
9.7	Shards	25
9.8	Changes	25
9.9	Compact	25
9.10	Write a commit	26
9.11	Clear view	26
9.12	Get purged info limit	26
9.13	Set purged info limit	26
9.14	Revisions	26
9.15	Export and import databases	27
10	Documents	29
10.1	Get a document	29
10.2	Create a document	30
10.3	Modify a document	31
10.4	Delete a document	31
10.5	Copy a document	31
10.6	Local document	32
10.7	Get a bulk documents	32
10.8	Create documents in bulk	32
10.9	Attachments	32
10.10	Revisions	33
10.11	Purge document	34
11	Query	35
11.1	Find a document	35
12	Design documents	43
12.1	Get a design document	43
12.2	Creates a design document	44
12.3	Compress design document	47
12.4	Remove design document	47
13	Cmdlets and aliases	49
13.1	Cmdlets	49
13.2	Aliases	59
14	Classes	61
14.1	PSCouchDBQuery class	61
14.2	PSCouchDBDesignDoc class	67
15	Real uses	71
15.1	Simple machine inventory	71
15.2	Log storage	72

16 Support **75**

16.1 Contribute 75

16.2 Issue 75

16.3 Donations 75



The complete and powerful powershell module for CouchDB v2.X and v3.X

CHAPTER 1

Introduction

PSCouchDB is a powershell module to simplify your work on couchdb database.

Warning: Before continuing, install the latest version of CouchDB, following the [documentation](#).

1.1 Installation

Download and install latest [PSCouchDB module](#) by copying it under %Windir%\System32\WindowsPowerShell\v1.0\Modules for all users or under %UserProfile%\Documents\WindowsPowerShell\Modules for the current user or install through [PowershellGallery](#).

Note: For unix users the powershell module path is /usr/local/share/powershell/Modules for all users and ~/.local/share/powershell/Modules for current user.

1.1.1 Install with git

Installation by git from Github.

```
git clone https://github.com/MatteoGuadrini/PSCouchDB.git
cd PSCouchDB
# for Windows
copy /Y PSCouchDB %Windir%\System32\WindowsPowerShell\v1.0\Modules
# for Unix
cp -var PSCouchDB /usr/local/share/powershell/Modules
```

1.2 Signing

Important: This module is not signed. Before import or execute cmdlet on this module, see [about_signing](#) session. Verify execution of scripts is allowed with `Get-ExecutionPolicy` (should be `RemoteSigned` or `Unrestricted`). If scripts are not enabled, run PowerShell as Administrator and call `Set-ExecutionPolicy RemoteSigned -Scope CurrentUser -Confirm`.

1.3 Start

To get started, let's look at all the cmdlets in the module through this command:

```
Search-CouchDBHelp -Pattern . | Format-Table name -AutoSize
```

If we wanted to get only the cmdlets for a given task, use:

```
Search-CouchDBHelp -Pattern replication
```

and display the help of one the cmdlet found:

```
Search-CouchDBHelp -Pattern New-CouchDBReplication | Get-Help -Full
```

or search CouchDB API name:

```
Search-CouchDBHelp -Pattern _cluster_setup
```

or help about a module.

```
Get-Help about_pscouchdb
```

CouchDB 2.x can be deployed in either a single-node or a clustered setup. This section covers the first-time setup steps required for each of these configuration.

Warning: Before configuring one of the two modes, it is advisable to create an administrative user. Without it, the two cluster modes will not be complete. For more details, see [Permission](#) section.

2.1 Single Node

A single node cluster is nothing more than a cluster extended to a single node, ie the local one. To configure it, run:

```
Enable-CouchDBCluster -SingleNode -Authorization "admin:password"
```

2.2 Cluster

Same thing of the single node, but with two or more nodes. By default the nodes are 3.

```
Enable-CouchDBCluster -Authorization "admin:password"
```


Nodes are the single elements of a cluster. by element we mean a server, local or remote. To verify the active cluster nodes, run:

```
Get-CouchDBNode -Authorization "admin:password"
```

3.1 Add a node

To manually add a node to the cluster, simply run:

```
Add-CouchDBNode -BindAddress 127.0.1.1 -Authorization "admin:password"
```

3.2 Remove a node

To manually add a node to the cluster, simply run:

```
Get-CouchDBNode -Authorization "admin:password"  
Remove-CouchDBNode -BindAddress couchdb@127.0.1.1 -Authorization "admin:password"
```


The CouchDB Server Configuration API provide an interface to query and update the various configuration values within a running CouchDB instance.

4.1 Get configuration

To get the entire CouchDB server configuration. The structure is organized by different configuration sections, with individual values.

```
Get-CouchDBConfiguration -Authorization "admin:password"
```

4.2 Modify configuration

To modify configuration, see help of this cmdlet:

```
Set-CouchDBConfiguration -?  
help Set-CouchDBConfiguration  
Get-Help Set-CouchDBConfiguration
```

4.2.1 Modify an element

For example, to change SSL port 6984 with 443:

```
Set-CouchDBConfiguration -Element ssl -Key port -Value 443 -Authorization  
↪ "admin:password"
```

Note: This cmdlet return the old value. To verify the changed value, run: `Get-CouchDBConfiguration -Authorization "admin:password" | Select-Object ssl | Format-List`

5.1 Admin Party

When you start out fresh, CouchDB allows any request to be made by anyone. Create a database and delete some documents? Same deal. CouchDB calls this the Admin Party. Everybody has privileges to do anything.

This is very nice for testing environment, but anyone could delete documents or the whole database. By default, CouchDB, will listen only on your loopback network interface (127.0.0.1 or localhost) and thus only you will be able to make requests to CouchDB, nobody else. But when it is necessary to expose the service on public ip, you will want to think about restricting access.

CouchDB has the idea of an admin user (for example, an administrator, a super user, or root) that is allowed to do anything to a CouchDB installation. By default, everybody is an admin.

To restrict permissions, one or more administrators must be created.

Note: In CouchDB 3.X you have to set an admin in the installation process. Doing so will not make Admin Party work anymore.

5.2 Create Admin user

Admin Party allows any user to perform any database operation. This could be perfect for application development or on a test machine, but for production it would create many problems. To solve this problem, just create an admin user. At this point the admin user can create/modify/delete documents from the database.

Important: If the password is not specified, it will be prompted. For example, `-Authorization admin` will ask you to write the password at the prompt. The password has the format `*`.

```
$password = "password" | ConvertTo-SecureString -AsPlainText -Force
New-CouchDBAdmin -Userid admin -Password $password
```

Naturally, all reading requests can be made without user and password.

5.3 Members access

To protect a database from unauthorized requests, you must first create a user used for this purpose.

```
$password = "password" | ConvertTo-SecureString -AsPlainText -Force
New-CouchDBUser -Userid member_user -Password $password -Authorization "admin:password"
↪ "
```

And then enable it to the server.

```
Grant-CouchDBDatabasePermission -Database test -ReaderUser member_user -Authorization
↪ "admin:password"
```

Let's check the permissions now.

```
Get-CouchDBDatabaseSecurity -Database test -Authorization "member_user:password"
Get-CouchDBDatabase -Database test -Authorization "member_user:password"
```

5.4 Read only access

To protect a database from write requests, you need to create a design document that will contain a validation function. See this section: [Classes](#)

```
using module PSCouchDB
$ddoc = New-Object -TypeName PSCouchDBDesignDoc
$ddoc.AddValidation($true)
New-CouchDBDesignDocument -Database test -Document "mydesignndoc" -Data $ddoc.
↪ GetDesignDocuments() -Authorization "admin:password"
```

5.5 Limit write access

If you want to limit a single database with different admin user for reading and writing, use this cmdlet:

```
$password = "password" | ConvertTo-SecureString -AsPlainText -Force
New-CouchDBUser -Userid other_admin -Password $password -Authorization "admin:password"
↪ "
Grant-CouchDBDatabasePermission -Database test -AdminUser other_admin -Authorization
↪ "admin:password"
Get-CouchDBDatabase -Database test -Authorization "other_admin:password"
```

5.6 Revoke database permissions

To remove permissions from one database, run this cmdlet:

```
Revoke-CouchDBDatabasePermission -Database test -Authorization "admin:password"
```

5.7 Remove an admin

To remove an administrative user, run:

```
Remove-CouchDBAdmin -Userid admin -Authorization "admin:password"
```

5.8 Remove a user

To remove a simple user, run:

```
$user = Get-CouchDBUser -Userid member_user | Select-Object _id,_rev
Remove-CouchDBUser -Userid $user._id -Revision $user._rev -Authorization
↪ "admin:password"
```

5.9 Reset user password

To modify o reset password of a user.

```
$password = "new_password" | ConvertTo-SecureString -AsPlainText -Force
Set-CouchDBUser -Userid member_user -Password $password -Revision "2-
↪ 4705a219cdcca7c72aac4f623f5c46a8" -Authorization "admin:password"
```

5.10 Reset admin password

To modify o reset password of an admin.

```
$password = "new_password" | ConvertTo-SecureString -AsPlainText -Force
Set-CouchDBAdmin -Userid test_user -Password $password -Authorization "admin:password"
```


CHAPTER 6

Server operation

Below are all the cmdlets needed to get all the information on the CouchDB server.

To get the version and other info on the server, run:

```
Get-CouchDBServer
```

To get list of running tasks, including the task type, name, status and process ID:

```
Get-CouchDBActiveTask -Authorization "admin:password"
```

To get list of all the databases in the CouchDB instance:

```
Get-CouchDBDatabase
```

To get information of a list of the specified databases in the CouchDB instance:

```
Get-CouchDBDatabaseInfo -Keys test,test1,test2
```

To get the status of the node or cluster, run this:

```
Get-CouchDBClusterSetup -Authorization "admin:password"
```

To get a list of all database events in the CouchDB instance:

```
Get-CouchDBDatabaseUpdates -Authorization "admin:password"
```

For other parameter see the table below:

PARAMETER	DESCRIPTION
Feed	Option: normal: Returns all historical DB changes, then closes the connection. Default. longpoll: Closes the connection after the first event. continuous: Send a line of JSON per event. Keeps the socket open until timeout. eventsouce: Like, continuous, but sends the events in EventSource format.
Timeout	Number of seconds until CouchDB closes the connection. Default is 60.
Heartbeat	Period in milliseconds after which an empty line is sent in the results. Only applicable for longpoll, continuous, and eventsouce feeds. Overrides any timeout to keep the feed alive indefinitely. Default is 60000.
Since	Return only updates since the specified sequence ID. May be the string “now” to begin showing only new updates.

To displays the nodes that are part of the cluster:

```
Get-CouchDBNode -Authorization "admin:password"
```

To get the statistics for the running server:

```
Measure-CouchDBStatistics -Authorization "admin:password"
Measure-CouchDBStatistics -System -Authorization "admin:password"
```

To restart server:

Note: This task required privileged access on operating system

```
Restart-CouchDBServer
```

And check health:

```
Get-CouchDBServer -Status
```

To get one or more Universally Unique Identifiers (UUIDs) from the CouchDB instance:

```
New-CouchDBUuids
```

6.1 Read the log

To read entire log.

```
Read-CouchDBLog
```

Note: The default path on Windows is `C:\CouchDB\couch.log`, while on Unix it is `/var/log/couchdb/couch.log`. Otherwise, if the log is in a custom path, specify the path using the `-Path` parameter.

This example is used to read only the last 15 lines of log.

```
Read-CouchDBLog -Tail 15
```

Instead this to stay in append on the log for the level of warning.

```
Read-CouchDBLog -Level warning -Follow
```

6.1.1 Level

Each entry in the log has its own color, so as to identify the line of interest “at a glance”.

debug : Detailed debug logging.

info : Informative logging. Includes HTTP requests headlines, startup of an external processes etc.

notice

warning : Warning messages are alerts about edge situations that may lead to errors.

error : Error level includes only things that go wrong, like crash reports and HTTP error responses (5xx codes).

critical

alert

emergency

6.2 Clear the log

Clear entire and rotate (save a copy in the same folder) log.

```
Clear-CouchDBLog -Rotate
```


The replication is an incremental one way process involving two databases (a source and a destination).

7.1 Get replica

To get a replication document.

```
Get-CouchDBReplication -Authorization "admin:password"
```

To get a list of replication jobs.

```
Get-CouchDBReplicationScheduler -Authorization "admin:password"
```

And to get a list of replication document states.

```
Get-CouchDBReplicationDocument -Authorization "admin:password"
```

7.2 Create replica

Creation of the replicator database and replication agent is automatically in the same time.

```
New-CouchDBReplication -SourceDatabase test -TargetDatabase test_dump -Continuous -  
↪Authorization "admin:password"
```

Note: The `Authorization` parameter refers to source database. It is preferable that the destination database is not password protected, otherwise an inaccessibility error will return.

Now that we have a replicated document, let's take a look at the change.

```
Get-CouchDBDatabaseChanges -Database test_dump
```

7.3 Modify replica

To change the replication agent settings (continuous: true | false).

```
$replica = Get-CouchDBReplication -Authorization "admin:password"
Write-Output $replica.rows.id[1] $replica.rows.value[1].rev
# continuous: false
Set-CouchDBReplication -Document $replica.rows.id[1] -Revision $replica.rows.value[1].
↪rev -Authorization "admin:password"
# continuous: true
Set-CouchDBReplication -Document $replica.rows.id[1] -Revision $replica.rows.value[1].
↪rev -Continuous -Authorization "admin:password"
```

7.4 Remove replica

To remove the replication agent.

```
Remove-CouchDBReplication -Document localhost-test_localhost-test_dump -Authorization_
↪-Authorization "admin:password"
```

7.5 Replication request

Request, configure, or stop, a replication operation.

```
Request-CouchDBReplication -SourceDatabase test -TargetDatabase test_dump -Documents
↪"Hitchhikers", "Hitchhikers_Guide" -Authorization "admin:password"
```

CHAPTER 8

Authentication

CouchDB generates a token that the client can use for the next few requests to CouchDB. Tokens are valid until a timeout. When CouchDB sees a valid token in a subsequent request, it will authenticate the user by this token without requesting the password again. By default, cookies are valid for 10 minutes.

Set a cookie:

```
$password = "password" | ConvertTo-SecureString -AsPlainText -Force  
Set-CouchDBSession -UserId admin -Password $password
```

Get a session:

```
Get-CouchDBSession
```

Now let verify a protected read database without Authorization param:

```
Get-CouchDBDatabase -Database test
```

And remove a session:

```
Remove-CouchDBSession -Authorization "admin:password"
```


The Database endpoint provides an interface to an entire database with in CouchDB. These are database-level, rather than document-level requests.

9.1 Test a database

To verify the existence of a database.

```
Test-CouchDBDatabase -Database test
```

9.2 Read a database

Gets information about the specified database.

```
Get-CouchDBDatabase -Database test
```

To get a list of all databases, run this:

```
Get-CouchDBDatabase
```

And if use `_all_dbs` view, in this table you can find all the possible parameters.

PARAMETER	DESCRIPTION
Descending	Return the databases in descending order by key. Default is <code>false</code> .
EndKey	Stop returning databases when the specified key is reached.
Limit	Limit the number of the returned databases to the specified number.
Skip	Skip this number of databases before starting to return the results. Default is <code>0</code> .
StartKey	Return databases starting with the specified key.

9.3 Create a database

Creates a new database.

```
New-CouchDBDatabase -Database test -Authorization "admin:password"
```

Note: If we repeat the same request to CouchDB, it will response with 412 since the database already exists. If an invalid database name is supplied, CouchDB returns response with 400

9.4 Remove a database

Deletes the specified database, and all the documents and attachments contained within it.

```
Remove-CouchDBDatabase -Database test -Authorization "admin:password"
```

9.5 Copy a database

Create a new database *test_copy* by copying it from *test* database.

```
Copy-CouchDBDatabase -Database test -Destination test_copy -Authorization  
↪ "admin:password"
```

Or copying local database *test* to a remote server.

```
Copy-CouchDBDatabase -RemoteServer remote_srv -Database test -RemoteAuthorization  
↪ "admin:password"
```

9.6 Index

Mango is a declarative JSON querying language for CouchDB databases. Mango wraps several index types, starting with the Primary Index out-of-the-box.

9.6.1 Get a index

To get list of index present on a database.

```
Get-CouchDBIndex -Database test
```

9.6.2 Create a new index

Create a new index on a database.

```
New-CouchDBIndex -Database test -Name test-index -Fields name,surname -Authorization  
↪ "admin:password"
```

9.6.3 Remove a index

Remove an existing index.

```
$idx = Get-CouchDBIndex -Database test
Remove-CouchDBIndex -Database test -DesignDoc $idx.indexes.ddoc[1] -Name $idx.indexes.
↳name[1] -Authorization "admin:password"
```

9.7 Shards

Get a list of database shards. Each shard will have its internal database range, and the nodes on which replicas of those shards are stored.

9.7.1 Get shards

Get a list a database shards.

```
Get-CouchDBDatabaseShards -Database test
```

Get the shard document on database.

```
Get-CouchDBDatabaseShards -Database test -Document 00000000-1fffffffff
```

9.7.2 Sync shards

For the given database, force-starts internal shard synchronization for all replicas of all database shards.

```
Sync-CouchDBDatabaseShards -Database test -Authorization "admin:password"
```

9.8 Changes

To get a sorted list of changes made to documents in the database, in time order of application, can be obtained from the database's `_changes` resource.

```
Get-CouchDBDatabaseChanges -Database test
```

9.9 Compact

Request compaction of the specified database. Compaction can only be requested on an individual database; you cannot compact all the databases for a CouchDB instance. The compaction process runs as a background process.

```
Compress-CouchDBDatabase -Database test -Authorization "admin:password"
```

9.10 Write a commit

Commits any recent changes to the specified database to disk.

```
Write-CouchDBFullCommit -Database test -Authorization "admin:password"
```

9.11 Clear view

Removes view index files that are no longer required by CouchDB as a result of changed views within design documents.

```
Clear-CouchDBView -Database test -Authorization "admin:password"
```

9.12 Get purged info limit

Gets the current `purged_infos_limit` (purged documents limit) setting, the maximum number of historical purges (purged document IDs with their revisions) that can be stored in the database.

```
Get-CouchDBDatabasePurgedLimit -Database test
```

9.13 Set purged info limit

Set the current `purged_infos_limit` (purged documents limit) setting.

```
Set-CouchDBDatabasePurgedLimit -Database test -Limit 1500 -Authorization  
↪ "admin:password"
```

9.14 Revisions

Working with database revisions.

9.14.1 Get missing revisions

Get a list of document revisions, returns the document revisions that do not exist in the database.

```
Get-CouchDBMissingRevision -Database test -Document "Hitchhikers" -Revision 2-  
↪ 7051cbe5c8faecd085a3fa619e6e6337, 3-825cb35de44c433bfb2df415563a19de
```

9.14.2 Get revision difference

Given a set of document/revision IDs, returns the subset of those that do not correspond to revisions stored in the database.

```
Get-CouchDBRevisionDifference -Database test -Document "Hitchhikers" -Revision 2-
↪ 7051cbe5c8faecd085a3fa619e6e6337, 3-825cb35de44c433bfb2df415563a19de
```

9.14.3 Get revision limit

Gets the current revs_limit (revision limit) setting.

```
Get-CouchDBRevisionLimit -Database test
```

9.14.4 Set revision limit

Sets the maximum number of document revisions that will be tracked by CouchDB.

```
Set-CouchDBRevisionLimit -Database test -Limit 1500 -Authorization "admin:password"
```

9.15 Export and import databases

One of the most common practices for perform backup a database is to export it. On the other hand, to restore a database, just import it.

9.15.1 Export

For export a database in a json file format.

Note: If you do not specify the path and file name, a JSON file will be saved in UTF8 format in the current path (\$pwd) with this name: name-of-database_t-i-m-e_s_t_a_m_p.json.

```
Export-CouchDBDatabase -Database test
```

9.15.2 Import

For import or restore a database from JSON file.

```
Import-CouchDBDatabase -Database test -Path test_01-25-2019_00_01_00.json
```

And this, for create a new database from JSON file.

```
Import-CouchDBDatabase -Database test_restored -Path test_01-25-2019_00_01_00.json -
↪ RemoveRevision
```


CHAPTER 10

Documents

All documents are contained in a database. Each document is in json format.

10.1 Get a document

To get document by the specified `Document` from the specified `Database`. Unless you request a specific revision, the latest revision of the document will always be returned.

```
Get-CouchDBDocument -Database test -Document "Hitchhikers"
```

In this table you can find all the possible parameters to get the document.

PARAMETER	DESCRIPTION
Revision	The CouchDB revision document.
Local	Return CouchDB local document.
Revisions	Return all CouchDB db revisions.
History	Return all info CouchDB db revisions.
Attachments	Includes attachments bodies in response.
Attach-mentsInfo	Includes encoding information in attachment stubs if the particular attachment is compressed.
AttachmentsS-ince	Includes attachments only since specified revisions. Doesn't includes attachments for specified revisions.
Conflicts	Includes information about conflicts in document.
DeletedCon-flicts	Includes information about deleted conflicted revisions.
Latest	Forces retrieving latest "leaf" revision, no matter what rev was requested.
LocalSequence	Includes last update sequence for the document.
Metadata	Acts same as specifying all conflicts, deleted_conflicts and revs_info query parameters.
OpenRevisions	Retrieves documents of specified leaf revisions. Additionally, value all id default and to return all leaf revisions.

And if use `_all_docs` view, in this table you can find all the possible parameters.

PARAMETER	DESCRIPTION
Descending	Return the documents in descending by key order. Default is <code>false</code> .
End-Key	Stop returning records when the specified key is reached.
End-Key-Documents	Stop returning records when the specified document ID is reached.
Group	Group the results using the reduce function to a group or single row. Implies <code>reduce</code> is <code>true</code> and the maximum <code>group_level</code> . Default is <code>false</code> .
GroupLevel	Specify the group level to be used. Implies <code>group</code> is <code>true</code> .
Include-Documents	Include the associated document with each row. Default is <code>false</code> .
InclusiveEnd	Specifies whether the specified end key should be included in the result. Default is <code>true</code> .
Key	Return only documents that match the specified key. The document must be <code>_all_docs</code> .
Keys	Return only documents where the key matches one of the keys specified in the array.
Limit	Limit the number of the returned design documents to the specified number.
Reduce	Use the reduction function. Default is <code>true</code> when a reduce function is defined.
Skip	Skip this number of records before starting to return the results.
Sorted	Sort returned rows. Setting this to <code>false</code> offers a performance boost. The <code>total_rows</code> and <code>offset</code> fields are not available when this is set to <code>false</code> . Default is <code>true</code> .
Stable	Whether or not the view results should be returned from a stable set of shards. Default is <code>false</code> .
Stale	Allow the results from a stale view to be used. Supported values: <code>ok</code> , <code>update_after</code> and <code>false</code> . <code>ok</code> is equivalent to <code>stable=true&update=false</code> . <code>update_after</code> is equivalent to <code>stable=true&update=lazy</code> . <code>false</code> is equivalent to <code>stable=false&update=true</code> .
StartKey	Return records starting with the specified key.
StartKey-Documents	Return records starting with the specified document ID. Ignored if <code>startkey</code> is not set.
Update	Whether or not the view in question should be updated prior to responding to the user. Supported values: <code>true</code> , <code>false</code> , <code>lazy</code> . Default is <code>true</code> .
Update-Sequence	Whether to include in the response an <code>update_seq</code> value indicating the sequence id of the database the view reflects. Default is <code>false</code> .

10.2 Create a document

To creates a new named document, or creates a new revision of the existing document. The `Data` parameter it can be a json or a hashtable object.

```
$data = '{"planet":"Magrathea", "name":"Slartibartfast"}'
New-CouchDBDocument -Database test -Document "Hitchhikers" -Data $data -Authorization
↪ "admin:password"
```

There is also the possibility of enabling a batch mode.

```
$data = '{"planet":"Magrathea", "name":"Slartibartfast"}'
New-CouchDBDocument -Database test -Document "Hitchhikers" -Data $data -BatchMode -
↪ Authorization "admin:password"
Write-CouchDBFullCommit -Database test -Authorization "admin:password"
```

Note: Until you run the `Write-CouchDBFullCommit` cmdlet, the document will not be written to disk but kept only in memory. This can be useful in case of bulk writing.

10.3 Modify a document

With `Revision` parameter it is possible to overwrite the document. The document retain the previously written elements. If an item is specified again, it will be overwritten.

```
$data = @{"answer"=42; "ask"="Ultimate Question of Life, the Universe and Everything"}
Set-CouchDBDocument -Database test -Document "Hitchhikers" -Revision 1-
↪ 2c903913030efb4d711db085b1f44107 -Data $data -Authorization "admin:password"
```

With `Replace` parameter, the document is re-write again.

```
$data = '{"planet":"Heart", "name":"Arthur Dent"}'
Set-CouchDBDocument -Database test -Document "Hitchhikers" -Revision 2-
↪ 9a68ee74a8276c7f11146245ba43676f -Data $data -Replace -Authorization "admin:password"
↪ "
```

10.4 Delete a document

To delete a document, specify `Revision` parameter.

Note: CouchDB doesn't completely delete the specified document. Instead, it leaves a tombstone with very basic information about the document. The tombstone is required so that the delete action can be replicated across databases.

```
Remove-CouchDBDocument -Database test -Document "Hitchhikers" -Revision "3-
↪ 399796e5ce019e04311637e8a8a0f402" -Authorization "admin:password"
```

10.5 Copy a document

Copies an existing document to a new or existing document. Copying a document is only possible within the same database.

```
Copy-CouchDBDocument -Database test -Document "Hitchhikers" -Destination "Hitchhikers_
↳Guide" -Authorization "admin:password"
Copy-CouchDBDocument -Database test -Document "Hitchhikers" -Destination "Hitchhikers_
↳Guide _deleted" -Revision 3-399796e5ce019e04311637e8a8a0f402 -Authorization
↳"admin:password"
```

10.6 Local document

To get of all of the local documents in a given database.

```
Get-CouchDBDocument -Database test -Local
```

10.7 Get a bulk documents

This method can be called to query several documents in bulk.

```
Get-CouchDBBulkDocument -Database test -Document "Hitchhikers","Hitchhikers Guide _
↳deleted","Hitchhikers Guide"
```

10.8 Create documents in bulk

The bulk document API allows you to create and update multiple documents at the same time within a single request.

```
Set-CouchDBBulkDocument -Database test -Document "Hitchhikers","Hitchhikers_new",
↳"Hitchhikers Guide" -Revision 4-7051cbe5c8faecd085a3fa619e6e6337,$null,3-
↳399796e5ce019e04311637e8a8a0f402 -Authorization "admin:password"
```

10.9 Attachments

Document can includes attachments, then the returned structure will contain a summary of the attachments associated with the document.

10.9.1 Get an attachment

It's possible to retrieve document with all attached files content.

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt
```

Also is possible save a file.

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt -
↳OutFile "C:\out.txt"
```

Or get info of specific attachment.

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt -
↳ Info
```

10.9.2 Create an attachment

It's possible to retrieve document with all attached files content.

```
New-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment "C:\test.txt
↳ " -Revision "4-f6d66c4d70da66cded6bea889468eb14" -Authorization "admin:password"
```

10.9.3 Modify an attachment

To replace or add an attachment.

```
Set-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment "C:\out.txt"
↳ -Authorization "admin:password"
```

10.9.4 Delete an attachment

To remove an attachment.

```
Remove-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment out.txt -
↳ Revision "5-7bf1766d9a5f3e4a60b400e98d62f523" -Authorization "admin:password"
```

10.10 Revisions

10.10.1 Get a list of revisions

You can obtain a list of the revisions for a given document.

```
Get-CouchDBDocument -Database test -Document "Hitchhikers" -Revisions
```

10.10.2 Get a history of revisions

You can get additional information (history) about the revisions for a given document.

```
Get-CouchDBDocument -Database test -Document "Hitchhikers" -History
```

10.10.3 Get a specific revision

To get a specific revision, use the `Revision` parameter, and specify the full revision number.

```
Get-CouchDBDocument -Database test -Document "Hitchhikers" -Revision "5-
↳ 7bf1766d9a5f3e4a60b400e98d62f523"
```

10.10.4 Missing revision

With given a list of document revisions, returns the document revisions that do not exist in the database.

```
Get-CouchDBMissingRevision -Database test -Document "Hitchhikers" -Revision 2-  
↪ 7051cbe5c8faecd085a3fa619e6e6337,5-7bf1766d9a5f3e4a60b400e98d62f523 -Authorization  
↪ "admin:password"
```

10.11 Purge document

A database purge permanently removes the references to documents in the database. Normal deletion of a document within CouchDB does not remove the document from the database, instead, the document is marked as `_deleted=true` (and a new revision is created). This is to ensure that deleted documents can be replicated to other databases as having been deleted.

```
Clear-CouchDBDocuments -Database test -Document "Hitchhikers" -Authorization  
↪ "admin:password"
```

11.1 Find a document

To search for documents in a database, use the following cmdlet.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet
```

or with native Mango query

```
Find-CouchDBDocuments -Database test -Find '{"selector": {"name":{"$eq":"Arthur Dent"}
↪},"fields":["_id","name","planet"]}'
```

or with class (for complex query)

```
using module PSCouchDB
$q = New-Object -TypeName PSCouchDBQuery
$q.AddSelector("name","Arthur Dent")
$q.AddSelectorOperator('$eq')
$q.AddFields("_id")
$q.AddFields("name")
$q.AddFields("planet")
Find-CouchDBDocuments -Database test -Find $q.GetNativeQuery()
```

If you want to use Mango queries, follow the next sections. Otherwise you can see more examples in the [Classes](#) section.

11.1.1 Search a document

To perform a more generic search in a database, without knowing the various selectors, use:

```
Search-CouchDBFullText -Database test -Patterns "space","planet"
```

Warning: This search is much slower than the `Find-CouchdbDocuments` cmdlet.

11.1.2 Selector

Selectors are expressed as a JSON object describing documents of interest. Within this structure, you can apply conditional logic using specially named fields.

```
{
  "selector": {
    "name": "Arthur Dent"
  }
}
```

```
{
  "selector": {
    "name": {
      "FirstName": "Arthur Dent"
    }
  }
}

{
  "selector": {
    "name.FirstName": "Arthur Dent"
  }
}
```

11.1.3 Operators

Operators are identified by the use of a dollar sign (\$) prefix in the name field. There are two core types of operators in the selector syntax:

- Combination operators
- Condition operators

```
{
  "selector": {
    "name": "Arthur Dent"
  }
}
```

There are two implicit operators:

- Equality
- And

In a selector, any field containing a JSON value, but that has no operators in it, is considered to be an equality condition. The implicit equality test applies also for fields and subfields.

```
{
  "selector": {
    "name": {
```

(continues on next page)

(continued from previous page)

```

        "$eq": "Arthur Dent"
    }
}

```

is same to

```

{
    "selector": {
        "name": "Arthur Dent"
    }
}

```

List of available operators:

Operator type	Operator	Purpose
(In)equality	lt	The field is less than the argument
	lte	The field is less than or equal to the argument
	eq	The field is equal to the argument
	ne	The field is not equal to the argument
	gte	The field is greater than or equal to the argument
	gt	The field is greater than the to the argument
Object	exists	Check whether the field exists or not, regardless
	type	Check the document field's type. Valid values are "null", "boolean", "number", "string", "array", and "object"
Array	in	The document field must exist in the list provided
	nin	The document field not must exist in the list provided
	size	Special condition to match the length of an array field in a document. Non-array fields cannot match this condition
Miscellaneous	mod	Divisor and Remainder are both positive or negative integers. Non-integer values result in a 404.
	regex	A regular expression pattern to match against the document field. The matching algorithms are based on the Perl Compatible Regular Expression (PCRE) library.

Examples

```

using module PSCouchDB
$q = New-Object -TypeName PSCouchDBQuery
$q.AddSelector("name", "Arthur Dent")
$q.AddSelectorOperator('$eq')
$q.AddFields("_id")
$q.AddFields("name")
$q.AddFields("planet")
Find-CouchDBDocuments -Database test -Find $q.GetNativeQuery()

```

```

Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet

```

Warning: Pay attention to the \$ (dollar) sign. If you use the PSCouchDBQuery class or a native query, the sign is required.

11.1.4 Logical operators

Logical operators are used to combine selectors.

Important: Logical operators are only available when creating an object of type `PSCouchDBQuery` or use a native query string. For more details, see [Classes section](#) section.

AND

```
{
  "$and": [
    {
      "_id": { "$gt": null }
    },
    {
      "name": {
        "$eq": "Arthur Dent"
      }
    }
  ]
}
```

OR

```
{
  "name": "Arthur Dent",
  "$or": [
    { "planet": "Heart" },
    { "planet": "Magrathea" }
  ]
}
```

NOT

```
{
  "name": {
    "$eq": "Arthur Dent"
  },
  "name": {
    "$eq": "Slartibartfast"
  },
  "$not": {
    "name": "Ford Prefect"
  }
}
```

Operator	Purpose
and	Matches if all the selectors in the array match
or	Matches if any of the selectors in the array match. All selectors must use the same index
not	Matches if the given selector does not match
nor	Matches if none of the selectors in the array match
all	Matches an array value if it contains all the elements of the argument array
elem-Match	Matches and returns all documents that contain an array field with at least one element that matches all the specified query criteria
allMatch	Matches and returns all documents that contain an array field with all its elements matching all the specified query criteria

11.1.5 Sort

The sort field contains a list of field name and direction pairs, expressed as a basic array. The first field name and direction pair is the topmost level of sort. The second pair, if provided, is the next level of sort. The direction value is “asc” for ascending, and “desc” for descending. If you omit the direction value, the default “asc” is used.

```
{
  "selector": {"name": "Arthur Dent"},
  "sort": [{"name": "asc"}, {"planet": "asc"}]
}
```

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Sort name,planet
```

11.1.6 Limit

Maximum number of results returned. Default is 25.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Limit 100
```

11.1.7 Skip

Skip the first ‘n’ results, where ‘n’ is the value specified.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Skip 10
```

11.1.8 Use index

Instruct a query to use a specific index.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -UseIndex "index_planet"
```

11.1.9 Read quorum

Read quorum needed for the result. This defaults to 1, in which case the document found in the index is returned.

If set to a higher value, each document is read from at least that many replicas before it is returned in the results. This is likely to take more time than using only the document stored locally with the index.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -ReadQuorum 3
```

11.1.10 Bookmark

A string that enables you to specify which page of results you require. Used for paging through result sets. Every query returns an opaque string under the bookmark key that can then be passed back in a query to get the next page of results. If any part of the selector query changes between requests, the results are undefined.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Bookmark "my_bookmark"
```

11.1.11 No Update

Whether to update the index prior to returning the result. Default is true.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -NoUpdate
```

11.1.12 Stable

Whether or not the view results should be returned from a “stable” set of shards.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Stable
```

11.1.13 Stale

Combination of update=false and stable=true options. Possible options: "ok"

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Stale 'ok'
```

11.1.14 Execution statistics

Include execution statistics in the query response.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -ExecutionStats
```

11.1.15 Explain

Shows which index is being used by the query.

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↪ -Fields _id,name,planet -Sort name,planet -Explain
```

Design documents

In CouchDB, design documents provide the main interface for building a CouchDB application. The design document defines the views used to extract information from CouchDB through one or more views. Design documents are created within your CouchDB instance in the same way as you create database documents, but the content and definition of the documents is different.

12.1 Get a design document

Returns the contents of the design document specified with the name of the design document and from the specified database. Unless you request a specific revision, the latest revision of the document will always be returned.

```
Get-CouchDBDesignDocument -Database test -Document "space"
```

To get all the Design Documents in a database.

```
Get-CouchDBDatabaseDesignDocument -Database test
```

In this table you can find all the possible parameters to get the design documents with this cmdlet.

PARAMETER	DESCRIPTION
Descending	Return the design documents in descending by key order. Default is false.
EndKey	Stop returning records when the specified key is reached.
EndKeyDocument	Stop returning records when the specified design document ID is reached.
IncludeDocument	Include the full content of the design documents in the return. Default is false.
InclusiveEnd	Specifies whether the specified end key should be included in the result. Default is true.
Key	Return only design documents that match the specified key.
Keys	Return only design documents that match the specified keys.
Conflict	Includes conflicts information in response. Ignored if include_docs isn't true. Default is false.
Limit	Limit the number of the returned design documents to the specified number.
Skip	Skip this number of records before starting to return the results. Default is 0.
StartKey	Return records starting with the specified key.
StartKeyDocument	Return records starting with the specified design document ID.
UpdateSequence	Response includes an update_seq value indicating which sequence id of the underlying database the view reflects. Default is false.

12.1.1 Get design document attachment

To retrieve or save an attachment in a design document.

```
Get-CouchDBDesignDocumentAttachment -Database test2 -Document space -Attachment test.  
↪txt -OutFile
```

12.2 Creates a design document

Creates a new named design document.

```
New-CouchDBDesignDocument -Database test -Document "space" -ViewName "planet_view" -  
↪Authorization "admin:password"
```

12.2.1 Views

The definition of a view within a design document also creates an index based on the key information defined within each view. The production and use of the index significantly increases the speed of access and searching or selecting documents from the view. However, the index is not updated when new documents are added or modified in the database. View indexes are updated incrementally in the following situations:

- A new document has been added to the database.
- A document has been deleted from the database.
- A document in the database has been updated.

```
New-CouchDBDesignDocument -Database test -Document space -ViewName planet -ViewKey_
↪planet -Authorization "admin:password"
Set-CouchDBDesignDocument -Database test -Document space -ViewName planet_heart -
↪ViewKey planet -ViewValue "Heart" -Authorization "admin:password"
Set-CouchDBDesignDocument -Database test -Document space -ViewName planet_heart -
↪ViewKey planet -ViewValue "Heart2" -Authorization "admin:password"
```

Now, navigate with your favorite browser to http://localhost:5984/test/_design/space/_view/planet or

```
Get-CouchDBDocument -Database test -Document "_design/space/_view/planet"
```

12.2.2 List

Applies list function for the view function from the same design document.

```
New-CouchDBDesignDocument -Database test -Document space -ViewName planet -ListName_
↪planet_list -Authorization "admin:password"
```

Now, navigate with your favorite browser to http://localhost:5984/test/_design/space/_list/planet_list/planet or

```
$html = Get-CouchDBDocument -Database test -Document "_design/space/_list/planet_list/
↪planet"
$html.html.body.table.tr | fl
```

12.2.3 Show

Show functions are used to represent documents as HTML pages with nice formatting. They can also be used to run server-side functions without requiring a pre-existing document.

```
New-CouchDBDesignDocument -Database test -Document space -ShowName planet -
↪Authorization "admin:password"
Set-CouchDBDesignDocument -Database test -Document space -ShowName planet -ShowKey_
↪planet -Authorization "admin:password"
Set-CouchDBDesignDocument -Database test -Document space -ShowName planet -ShowKey_
↪planet -ShowValue "Heart" -Authorization "admin:password"
```

Now, navigate with your favorite browser to http://localhost:5984/test/_design/space/_show/planet/Hitchhikers or

```
Get-CouchDBDocument -Database test -Document "_design/space/_show/planet/Hitchhikers"
```

12.2.4 Validation

A design document may contain a function named `validate_doc_update` which can be used to prevent invalid or unauthorized document update requests from being stored. Only one function is allowed at a time.

```
New-CouchDBDesignDocument -Database test -Document space -ValidationRequirements_
↪planet -Authorization "admin:password"
Set-CouchDBDesignDocument -Database test -Document space -ValidationRequirements name,
↪planet -ValidationAuthor -Replace -Authorization "admin:password"
```

Now try to create a new document without validation element

```
$data = '{"planet":"Magrathea"}'
New-CouchDBDocument -Database test -Document "Test_Validation" -Data $data -
↪Authorization "admin:password"
```

Received an error: Invoke-RestMethod : {"error":"forbidden","reason":"Document must have a name"}. Now retry with this:

```
$data = '{"planet":"Magrathea", "name":"Slartibartfast"}'
New-CouchDBDocument -Database test -Document "Test_Validation" -Data $data -
↪Authorization "admin:password"
```

Note: Note that for this type of function, when you want to modify a design document, you need to specify the `-Replace` parameter, otherwise the function will not be changed.

12.2.5 Custom functions

It is also possible to define a custom Design Document, creating a here string that defines the document itself.

```
$ddoc = @"
{
  "language": "javascript",
  "views": {
    "all": {
      "map": "function(doc) { emit(doc.title, doc) }"
    },
    "by_title": {
      "map": "function(doc) { if (doc.title != null) emit(doc.title, doc) }"
    },
    "by_planet": {
      "map": "function(doc) { for(i=0;i<doc.keywords.length();i++) { emit(doc.
↪keywords[i], doc); } }"
    }
  },
  "shows": {
    "planet": "function(doc, req) { return '<h1>' + doc.title + '</h1>' }"
  }
}
"@
New-CouchDBDesignDocument -Database test -Document space -Data $ddoc -Authorization
↪"admin:password"
```

12.2.6 Create design document attachment

To create an attachment in a design document.

```
New-CouchDBDesignDocumentAttachment -Database test -Document space -Attachment
↪"C:\test.txt" -Revision 3-cfae968df80635ad15a9709e0264a988 -Authorization
↪"admin:password"
```

12.2.7 Modify design document attachment

To modify or add an attachment in a design document.

```
Set-CouchDBDesignDocumentAttachment -Database test -Document space -Attachment  
↪ "C:\test2.txt" -Revision 4-cfae968df80635ad15d5709e0264a988 -Authorization  
↪ "admin:password"
```

12.3 Compress design document

The compaction operation is the way to reduce disk space usage by removing unused and old data from database or view index files. This operation is very similar to the vacuum (SQLite ex.) operation available for other database management systems.

```
Compress-CouchDBDesignDocument -Database test -DesignDoc space -Authorization  
↪ "admin:password"
```

12.4 Remove design document

To remove a design document.

```
Remove-CouchDBDesignDocument -Database test -Document "mydesigndoc" -Revision "1-  
↪ 85a961d0d9b235b7b4f07baed1a38fda" -Authorization "admin:password"
```

12.4.1 Remove design document attachment

To remove an attachment in a design document.

```
Remove-CouchDBDesignDocumentAttachment -Database test -Document space -Attachment  
↪ "C:\test2.txt" -Revision 5-cfae778df80635ad15daa09e0264a988 -Authorization  
↪ "admin:password"
```

Cmdlets and aliases

Below is a list of cmdlets and aliases

13.1 Cmdlets

13.1.1 Help

Search-CouchDBHelp

```
Search-CouchDBHelp [-Pattern] <Object> [<CommonParameters>]
```

13.1.2 Configuration

Enable-CouchDBCluster

```
Enable-CouchDBCluster [[-Server] <String>] [[-Port] <Int32>] [[-NodeCount] <Int32>] [-  
↪SingleNode] [[-BindAddress] <String>] [[-BindPort] <Int32>] [[-RemoteNode] <String>  
↪] [[-RemoteUser] <String>] [[-RemotePassword] <SecureString>] [[-Authorization]  
↪<String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBNode

```
Get-CouchDBNode [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [[-  
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Add-CouchDBNode

```
Add-CouchDBNode [[-Server] <String>] [[-Port] <Int32>] [[-BindPort] <Int32>] [-  
↪BindAddress] <String> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Remove-CouchDBNode

```
Remove-CouchDBNode [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [-  
↪Node] <String> [[-Authorization] <String>] [-Force] [-Ssl] [-WhatIf] [-Confirm] [  
↪<CommonParameters>]
```

Get-CouchDBConfiguration

```
Get-CouchDBConfiguration [[-Server] <String>] [[-Port] <Int32>] [[-Node] <String>] [[-  
↪Session] <String>] [[-Key] <String>] [[-Authorization] <String>] [-Ssl] [  
↪<CommonParameters>]
```

Set-CouchDBConfiguration

```
Set-CouchDBConfiguration [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>  
↪] [[-Node] <String>] [-Element] <String> [-Key] <String> [-Value] <String> [[-  
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

13.1.3 Permission

New-CouchDBAdmin

```
New-CouchDBAdmin [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [[-  
↪Node] <String>] [-Userid] <String> [-Password] <SecureString> [[-Authorization]  
↪<String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBUser

```
New-CouchDBUser [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [-  
↪Userid] <String> [-Password] <SecureString> [[-Roles] <Array>] [[-Authorization]  
↪<String>] [-Ssl] [<CommonParameters>]
```

Grant-CouchDBDatabasePermission

```
Grant-CouchDBDatabasePermission [[-Server] <String>] [[-Port] <Int32>] [-Database]  
↪<String> [[-AdminUser] <Array>] [[-AdminRoles] <Array>] [[-MemberUser] <Array>] [[-  
↪MemberRoles] <Array>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBDatabaseSecurity

```
Get-CouchDBDatabaseSecurity [[-Server] <String>] [[-Port] <Int32>] [-Database]  
↪<String> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Revoke-CouchDBDatabasePermission

```
Revoke-CouchDBDatabasePermission [[-Server] <String>] [[-Port] <Int32>] [-Database]  
↪<String> [[-Authorization] <String>] [-Force] [-Ssl] [-WhatIf] [-Confirm] [  
↪<CommonParameters>]
```

Remove-CouchDBAdmin

```
Remove-CouchDBAdmin [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [[-  
↪Node] <String>] [-Userid] <String> [[-Authorization] <String>] [-Force] [-Ssl] [-  
↪WhatIf] [-Confirm] [<CommonParameters>]
```

Remove-CouchDBUser

```
Remove-CouchDBUser [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [-
↪Userid] <String> [-Revision] <String> [[-Authorization] <String>] [-Force] [-Ssl] [-
↪WhatIf] [-Confirm] [<CommonParameters>]
```

Set-CouchDBUser

```
Set-CouchDBUser [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [-
↪Userid] <String> [-Password] <SecureString> [[-Roles] <Array>] [-Revision] <String>
↪ [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Set-CouchDBAdmin

```
Set-CouchDBAdmin [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [[-
↪Node] <String>] [-Userid] <String> [-Password] <SecureString> [[-Authorization]
↪ <String>] [-Ssl] [<CommonParameters>]
```

13.1.4 Server

Get-CouchDBServer

```
Get-CouchDBServer [[-Server] <String>] [[-Port] <Int32>] [[-Authorization] <String>]
↪ [-Status] [-Ssl] [<CommonParameters>]
```

Get-CouchDBActiveTask

```
Get-CouchDBActiveTask [[-Server] <String>] [[-Port] <Int32>] [[-Authorization]
↪ <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBClusterSetup

```
Get-CouchDBClusterSetup [[-Server] <String>] [[-Port] <Int32>] [[-
↪EnsureDatabaseExist] <Array>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
↪]
```

Get-CouchDBDatabaseUpdates

```
Get-CouchDBDatabaseUpdates [[-Server] <String>] [[-Port] <Int32>] [[-Feed] <String>]
↪ [[-Timeout] <Int32>] [[-Heartbeat] <Int32>] [[-Since] <String>] [[-Authorization]
↪ <String>] [-Ssl] [<CommonParameters>]
```

Measure-CouchDBStatistics

```
Measure-CouchDBStatistics [[-Server] <String>] [[-Port] <Int32>] [-System] [[-
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Restart-CouchDBServer

```
Restart-CouchDBServer [-Force] [-WhatIf] [-Confirm] [<CommonParameters>]
```

New-CouchDBUuids

```
New-CouchDBUuids [[-Server] <String>] [[-Port] <Int32>] [[-Count] <Int32>] [[-
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Read-CouchDBLog

```
Read-CouchDBLog [[-Path] <String>] [[-Level] <String>] [-Follow] [[-Tail] <Int32>] [  
↪ <CommonParameters>]
```

Clear-CouchDBLog

```
Clear-CouchDBLog [[-Path] <String>] [-Rotate] [<CommonParameters>]
```

13.1.5 Replication

Get-CouchDBReplication

```
Get-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>]   
↪ [[-Document] <String>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBReplicationScheduler

```
Get-CouchDBReplicationScheduler [[-Server] <String>] [[-Port] <Int32>] [[-Limit]   
↪ <Int32>] [[-Skip] <Int32>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBReplicationDocument

```
Get-CouchDBReplicationDocument [[-Server] <String>] [[-Port] <Int32>] [[-Limit]   
↪ <Int32>] [[-Skip] <Int32>] [[-ReplicatorDatabase] <String>] [[-ReplicatorDocuments]   
↪ <String>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBReplication

```
New-CouchDBReplication [[-SourceServer] <String>] [[-TargetServer] <String>] [[-   
↪ SourcePort] <Int32>] [[-TargetPort] <Int32>] [[-Database] <String>] [[-   
↪ SourceDatabase] <String>] [[-TargetDatabase] <String>] [-Continuous] [[-   
↪ Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBDatabaseChanges

```
Get-CouchDBDatabaseChanges [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>   
↪ [[-Filter] <Array>] [-Continuous] [[-Authorization] <String>] [-Ssl] [  
↪ <CommonParameters>]
```

Set-CouchDBReplication

```
Set-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>]   
↪ [-Document] <String> [-Revision] <String> [-Continuous] [[-Authorization] <String>]   
↪ [-Ssl] [<CommonParameters>]
```

Remove-CouchDBReplication

```
Remove-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>   
↪] [-Document] <String> [-Revision] <String> [[-Authorization] <String>] [-Force] [-   
↪ Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Request-CouchDBReplication

```
Request-CouchDBReplication [[-SourceServer] <String>] [[-TargetServer] <String>] [[-   
↪ SourcePort] <Int32>] [[-TargetPort] <Int32>] [-SourceDatabase] <String> [-   
↪ TargetDatabase] <String> [[-Proxy] <String>] [[-Documents] <Array>] [[-Filter]   
↪ <String>] [-Continuous] [-Cancel] [-CreateTargetDatabase] [[-Authorization] <String>   
↪] [-Ssl] [<CommonParameters>]
```

(continues on next page)

(continued from previous page)

13.1.6 Authentication

Set-CouchDBSession

```
Set-CouchDBSession [[-Server] <String>] [[-Port] <Int32>] [-UserId] <String> [-
  Password] <SecureString> [-Ssl] [<CommonParameters>]
```

Get-CouchDBSession

```
Get-CouchDBSession [[-Server] <String>] [[-Port] <Int32>] [[-Authorization] <String>] [-
  Ssl] [<CommonParameters>]
```

Remove-CouchDBSession

```
Remove-CouchDBSession [[-Server] <String>] [[-Port] <Int32>] [[-Authorization]
  <String>] [-Ssl] [<CommonParameters>]
```

13.1.7 Databases

Test-CouchDBDatabase

```
Test-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
  Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Copy-CouchDBDatabase

```
Copy-CouchDBDatabase [[-Server] <String>] [[-RemoteServer] <String>] [[-Port] <Int32>
  [-RemotePort] <Int32>] [-Database] <String> [[-Destination] <String>] [[-
  ExcludeIds] <Array>] [[-Authorization] <String>] [[-RemoteAuthorization] <String>] [-
  Ssl] [<CommonParameters>]
```

Get-CouchDBDatabase

```
Get-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>] [-
  Descending] [[-EndKey] <String>] [[-Limit] <Int32>] [[-Skip] <Int32>] [[-StartKey]
  <String>] [[-Authorization] <String>] [-Ssl]
  [<CommonParameters>]
```

New-CouchDBDatabase

```
New-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
  Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Remove-CouchDBDatabase

```
Remove-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
  Authorization] <String>] [-Force] [-Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Get-CouchDBIndex

```
Get-CouchDBIndex [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
  Authorization] <String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBIndex

```
New-CouchDBIndex [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-Name]
↪<String> [-Fields] <Array> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Remove-CouchDBIndex

```
Remove-CouchDBIndex [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↪DesignDoc] <String> [-Name] <String> [[-Authorization] <String>] [-Force] [-Ssl] [-
↪WhatIf] [-Confirm] [<CommonParameters>]
```

Get-CouchDBDatabaseInfo

```
Get-CouchDBDatabaseInfo [[-Server] <String>] [[-Port] <Int32>] [[-Keys] <Array>] [[-
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBDatabaseShards

```
Get-CouchDBDatabaseShards [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> ↪
↪ [[-Document] <String>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Sync-CouchDBDatabaseShards

```
Sync-CouchDBDatabaseShards [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↪ [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Compress-CouchDBDatabase

```
Compress-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> ↪
↪ [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Write-CouchDBFullCommit

```
Write-CouchDBFullCommit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> ↪
↪ [[-Authorization] <String>] [-Force] [-Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]
↪ ]
```

Clear-CouchDBView

```
Clear-CouchDBView [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
↪Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBDatabasePurgedLimit

```
Get-CouchDBDatabasePurgedLimit [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Set-CouchDBDatabasePurgedLimit

```
Set-CouchDBDatabasePurgedLimit [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [-Limit] <Int32> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBMissingRevision

```
Get-CouchDBMissingRevision [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↪ [-Document] <String> [-Revision] <Array> [[-Authorization] <String>] [-Ssl] [
↪<CommonParameters>]
```

Get-CouchDBRevisionDifference

```
Get-CouchDBRevisionDifference [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [-Document] <String> [-Revision] <Array> [[-Authorization] <String>] [-
↪Ssl] [<CommonParameters>]
```

Get-CouchDBRevisionLimit

```
Get-CouchDBRevisionLimit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↪[[[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Set-CouchDBRevisionLimit

```
Set-CouchDBRevisionLimit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↪[[[-Limit] <Int32>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Export-CouchDBDatabase

```
Export-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
↪Path] <String>] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Import-CouchDBDatabase

```
Import-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↪Path] <String> [-RemoveRevision] [[-Authorization] <String>] [-Ssl] [
↪<CommonParameters>]
```

13.1.8 Documents

Get-CouchDBDocument

```
Get-CouchDBDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Revision <String>] [-Local] [-Revisions] [-History] [-
↪Attachments] [-AttachmentsInfo] [-AttachmentsSince <Array>] [-Conflicts] [-
↪DeletedConflicts] [-Latest] [-LocalSequence] [-Metadata] [-OpenRevisions <Array>] [-
↪Descending] [-EndKey <String>] [-EndKeyDocument <String>] [-Group] [-GroupLevel
↪<Int32>] [-IncludeDocuments] [-InclusiveEnd <Boolean>] [-Key <Object>] [-Keys
↪<Array>] [-Limit <Int32>] [-Reduce <Boolean>] [-Skip <Int32>] [-Sorted <Boolean>] [-
↪Stable] [-Stale <String>] [-StartKey <String>] [-StartKeyDocument <String>] [-
↪Update <String>] [-UpdateSequence] [-Authorization <String>] [-Ssl] [
↪<CommonParameters>]

Get-CouchDBDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Revision <String>] [-Info] [-Local] [-Authorization <String>]
↪[-Ssl] [<CommonParameters>]
```

New-CouchDBDocument

```
New-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↪Document] <String> [-Data] <Object> [-BatchMode] [[-Authorization] <String>] [-Ssl]
↪[<CommonParameters>]
```

Set-CouchDBDocument

```
Set-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↪Document] <String> [-Revision] <String> [[-Data] <Object>] [-Replace] [-BatchMode]
↪[-NoConflict] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Remove-CouchDBDocument

```
Remove-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <String> [-Revision] <String> [[-Authorization] <String>] [-Force] [-Ssl]   
→[-WhatIf] [-Confirm] [<CommonParameters>]
```

Copy-CouchDBDocument

```
Copy-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <String> [-Destination] <String> [[-Revision] <String>] [[-Authorization]  
→<String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBBulkDocument

```
Get-CouchDBBulkDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <Array> [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Set-CouchDBBulkDocument

```
Set-CouchDBBulkDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <Array> [[-Revision] <Array>] [-IsDeleted] [[-Authorization] <String>] [-  
→Ssl] [<CommonParameters>]
```

Get-CouchDBAttachment

```
Get-CouchDBAttachment [-Server <String>] [-Port <Int32>] [-Database <String>] [-  
→Document <String>] [-Revision <String>] [-Attachment <String>] [-OutFile <String>]   
→[-Authorization <String>] [-Ssl] [<CommonParameters>]
```

```
Get-CouchDBAttachment [-Server <String>] [-Port <Int32>] [-Database <String>] [-  
→Document <String>] [-Revision <String>] [-Info] [-Attachment <String>] [-  
→Authorization <String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBAttachment

```
New-CouchDBAttachment [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <String> [-Attachment] <String> [-Revision] <String> [[-Authorization]  
→<String>] [-Ssl] [<CommonParameters>]
```

Set-CouchDBAttachment

```
Set-CouchDBAttachment [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <String> [-Revision] <String> [[-Attachment] <String>] [[-Authorization]  
→<String>] [-Ssl] [<CommonParameters>]
```

Remove-CouchDBAttachment

```
Remove-CouchDBAttachment [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>   
→[-Document] <String> [-Attachment] <String> [-Revision] <String> [[-Authorization]  
→<String>] [-Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Clear-CouchDBDocuments

```
Clear-CouchDBDocuments [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-  
→Document] <Array> [[-Authorization] <String>] [-Force] [-Ssl] [-WhatIf] [-Confirm] [  
→<CommonParameters>]
```

Search-CouchDBFullText

```
Search-CouchDBFullText [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↪Patterns] <Array> [-UseQueries] [[-Authorization] <String>] [-Ssl] [
↪<CommonParameters>]
```

Find-CouchDBDocuments

```
Find-CouchDBDocuments [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Explain] [-Selector <String>] [-Value <Object>] [-Limit <Int32>] [-Skip <Int32>] [-
↪Fields <Array>] [-Sort <Array>] [-UseIndex <Array>] [-ReadQuorum <Int32>] [-
↪Bookmark <String>] [-NoUpdate] [-Stable] [-Stale <String>] [-ExecutionStats] [-
↪Operator<String>] [-Authorization <String>] [-Ssl] [<CommonParameters>]
```

```
Find-CouchDBDocuments [-Server <String>] [-Port <Int32>] [-Database <String>] [-Find
↪<String>] [-Authorization <String>] [-Ssl] [<CommonParameters>]
```

13.1.9 Design documents

Get-CouchDBDatabaseDesignDocument

```
Get-CouchDBDatabaseDesignDocument [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [-Descending] [[-EndKey] <String>] [[-EndKeyDocument] <String>] [-
↪IncludeDocument] [[-InclusiveEnd] <Boolean>] [[-Key] <String>] [[-Keys] <Array>] [-
↪Conflict] [[-Limit] <Int32>] [[-Skip] <Int32>] [[-StartKey] <String>] [[-
↪StartKeyDocument] <String>] [-UpdateSequence] [[-Authorization] <String>] [-Ssl] [
↪<CommonParameters>]
```

Get-CouchDBDesignDocument

```
Get-CouchDBDesignDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> ↪
↪[-Document] <String> [-Info] [[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

Get-CouchDBDesignDocumentAttachment

```
Get-CouchDBDesignDocumentAttachment [-Server <String>] [-Port <Int32>] [-Database
↪<String>] [-Document <String>] [-Revision <String>] [-Attachment <String>] [-
↪OutFile <String>] [-Authorization <String>] [-Ssl] [<CommonParameters>]
```

```
Get-CouchDBDesignDocumentAttachment [-Server <String>] [-Port <Int32>] [-Database
↪<String>] [-Document <String>] [-Revision <String>] [-Info] [-Attachment <String>] ↪
↪[-Authorization <String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBDesignDocumentAttachment

```
New-CouchDBDesignDocumentAttachment [[-Server] <String>] [[-Port] <Int32>] [-
↪Database] <String> [-Document] <String> [-Attachment] <String> [-Revision] <String> ↪
↪[[[-Authorization] <String>] [-Ssl] [<CommonParameters>]
```

New-CouchDBDesignDocument

```
New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ViewName <String>] [-ViewKey <String>] [-ViewValue <String>] [-
↪GetDoc] [-Authorization <String>] [-Ssl] [<CommonParameters>]
```

```
New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Data <Object>] [-Authorization <String>] [-Ssl] [
↪<CommonParameters>]
```

(continues on next page)

(continued from previous page)

```

New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ValidationRequirements <Array>] [-ValidationAuthor] [-
↪Authorization <String>] [-Ssl] [<CommonParameters>]

New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ShowName <String>] [-ShowKey <String>] [-ShowValue <String>] [-
↪Authorization <String>] [-Ssl] [<CommonParameters>]

New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ViewName <String>] [-ViewKey <String>] [-ViewValue <String>] [-
↪GetDoc] [-ListName <String>] [-Authorization <String>] [-Ssl] [<CommonParameters>]

```

Set-CouchDBDesignDocument

```

Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ViewName <String>] [-ViewKey <String>] [-ViewValue <String>] [-
↪GetDoc] [-Replace] [-Authorization <String>] [-Ssl] [<CommonParameters>]

Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Data <Object>] [-Replace] [-Authorization <String>] [-Ssl] [
↪<CommonParameters>]

Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ValidationRequirements <Array>] [-ValidationAuthor] [-Replace]
↪[-Authorization <String>] [-Ssl] [<CommonParameters>]

Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ShowName <String>] [-ShowKey <String>] [-ShowValue <String>] [-
↪Replace] [-Authorization <String>] [-Ssl] [<CommonParameters>]

Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ViewName <String>] [-ViewKey <String>] [-ViewValue <String>] [-
↪GetDoc] [-ListName <String>] [-Replace] [-Authorization <String>] [-Ssl] [
↪<CommonParameters>]

```

Set-CouchDBDesignDocumentAttachment

```

Set-CouchDBDesignDocumentAttachment [[-Server] <String>] [[-Port] <Int32>] [-
↪Database] <String> [-Document] <String> [-Revision] <String> [-Attachment] <String>
↪[[[-Authorization] <String>] [-Ssl] [<CommonParameters>]

```

Compress-CouchDBDesignDocument

```

Compress-CouchDBDesignDocument [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [-DesignDoc] <String> [[-Authorization] <String>] [-Ssl] [
↪<CommonParameters>]

```

Remove-CouchDBDesignDocument

```

Remove-CouchDBDesignDocument [[-Server] <String>] [[-Port] <Int32>] [-Database]
↪<String> [-Document] <String> [-Revision] <String> [[-Authorization] <String>] [-
↪Force] [-Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]

```

Remove-CouchDBDesignDocumentAttachment

```
Remove-CouchDBDesignDocumentAttachment [[-Server] <String>] [[-Port] <Int32>] [-
Database] <String> [-Document] <String> [-Attachment] <String> [-Revision] <String>
[-Authorization] <String> [-Ssl] [-WhatIf] [-Confirm] [<CommonParameters>]
```

13.2 Aliases

```
acnode -> Add-CouchDBNode
ccdb -> Compress-CouchDBDatabase
ccdd -> Compress-CouchDBDesignDocument
ccdoc -> Clear-CouchDBDocuments
ccview -> Clear-CouchDBView
cpdoc -> Copy-CouchDBDocument
gcdb -> Copy-CouchDBDatabase
eccl -> Enable-CouchDBCluster
fcdoc -> Find-CouchDBDocuments
finddoc -> Find-CouchDBDocuments
gcadm -> Get-CouchDBAdmin
gcatt -> Get-CouchDBAttachment
gcbdoc -> Get-CouchDBBulkDocument
gcbpl -> Get-CouchDBDatabasePurgedLimit
gccnf -> Get-CouchDBConfiguration
gccs -> Get-CouchDBClusterSetup
gcdb -> Get-CouchDBDatabase
gcdbc -> Get-CouchDBDatabaseChanges
gcdbp -> Grant-CouchDBDatabasePermission
gcdbsh -> Get-CouchDBDatabaseSecurity
gcdbu -> Get-CouchDBDatabaseUpdates
gcddd -> Get-CouchDBDatabaseDesignDocument
gcddoc -> Get-CouchDBDesignDocument
gcdatt -> Get-CouchDBDesignDocumentAttachment
gcdoc -> Get-CouchDBDocument
gcidx -> Get-CouchDBIndex
gcmr -> Get-CouchDBMissingRevision
gcnode -> Get-CouchDBNode
gcrd -> Get-CouchDBRevisionDifference
gcr1 -> Get-CouchDBRevisionLimit
gcrpdoc -> Get-CouchDBReplicationDocument
gcrpl -> Get-CouchDBReplication
gcrpls -> Get-CouchDBReplicationScheduler
gcsi -> Get-CouchDBServer
gctsk -> Get-CouchDBActiveTask
gcusr -> Get-CouchDBUser
helpc -> Search-CouchDBHelp
mcsts -> Measure-CouchDBStatistics
ncadm -> New-CouchDBAdmin
ncatt -> New-CouchDBAttachment
ncdb -> New-CouchDBDatabase
ncddoc -> New-CouchDBDesignDocument
ndatt -> New-CouchDBDesignDocumentAttachment
ncdoc -> New-CouchDBDocument
ncidx -> New-CouchDBIndex
ncrpl -> New-CouchDBReplication
ncusr -> New-CouchDBUser
ncuuid -> New-CouchDBUuids
```

(continues on next page)

(continued from previous page)

```
rcadm -> Remove-CouchDBAdmin
rcatt -> Remove-CouchDBAttachment
rcdb -> Remove-CouchDBDatabase
rcdbp -> Revoke-CouchDBDatabasePermission
rcdbr -> Request-CouchDBReplication
rcddoc -> Remove-CouchDBDesignDocument
rdatt -> Remove-CouchDBDesignDocumentAttachment
rcdoc -> Remove-CouchDBDocument
rcidx -> Remove-CouchDBIndex
rcnode -> Remove-CouchDBNode
rcrpl -> Remove-CouchDBReplication
rcs -> Remove-CouchDBSession
rcsrv -> Restart-CouchDBServer
rcusr -> Remove-CouchDBUser
scadm -> Set-CouchDBAdmin
scatt -> Set-CouchDBAttachment
scbd -> Set-CouchDBBulkDocument
scconf -> Set-CouchDBConfiguration
scdbpl -> Set-CouchDBDatabasePurgedLimit
scddoc -> Set-CouchDBDesignDocument
sdatt -> Set-CouchDBDesignDocumentAttachment
scdoc -> Set-CouchDBDocument
scds -> Sync-CouchDBDatabaseShards
scft -> Search-CouchDBFullText
scrl -> Set-CouchDBRevisionLimit
scrpl -> Set-CouchDBReplication
scs -> Set-CouchDBSession
scusr -> Set-CouchDBUser
src -> Search-CouchDBHelp
tcdb -> Test-CouchDBDatabase
wcfc -> Write-CouchDBFullCommit
ecdb -> Export-CouchDBDatabase
exportdb -> Export-CouchDBDatabase
icdb -> Export-CouchDBDatabase
importdb -> Export-CouchDBDatabase
rdblog -> Read-CouchDBLog
cdblog -> Clear-CouchDBLog
mkdb -> New-CouchDBDatabase
mkdoc -> New-CouchDBDocument
mkuser -> New-CouchDBUser
mkadmin -> New-CouchDBAdmin
rmdb -> Remove-CouchDBDatabase
rmdoc -> Remove-CouchDBDocument
rmuser -> Remove-CouchDBUser
rmadmin -> Remove-CouchDBAdmin
```

CHAPTER 14

Classes

PSCouchDB module has powershell classes that can be used to construct certain objects.

14.1 PSCouchDBQuery class

This class is used to construct a query object that is compatible with Mango core.

14.1.1 Properties

bookmark	Property	string bookmark {get;set;}
execution_stats	Property	bool execution_stats {get;set;}
fields	Property	array fields {get;set;}
limit	Property	int limit {get;set;}
r	Property	int r {get;set;}
selector	Property	hashtable selector {get;set;}
skip	Property	int skip {get;set;}
sort	Property	array sort {get;set;}
stable	Property	bool stable {get;set;}
stale	Property	string stale {get;set;}
update	Property	bool update {get;set;}
use_index	Property	array use_index {get;set;}

14.1.2 Methods

AddFields	Method	void AddFields(System.Object fields)
AddIndexies	Method	void AddIndexies(System.Object indexies)
AddLogicalOperator	Method	void AddLogicalOperator(System.Object operator)
AddSelector	Method	void AddSelector(System.Object key, System.Object ↪value)

(continues on next page)

(continued from previous page)

AddSelectorOperator	Method	void AddSelectorOperator(System.Object operator), void ↵ ↵AddSelectorOperator(System.Object operator, System.Object key, System.Object value)
AddSortAsc	Method	void AddSortAsc(System.Object selector)
AddSortDesc	Method	void AddSortDesc(System.Object selector)
DisableUpdate	Method	void DisableUpdate()
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetNativeQuery	Method	string GetNativeQuery()
GetType	Method	type GetType()
RemoveFields	Method	void RemoveFields(), void RemoveFields(System.Object ↵ ↵field)
RemoveIndexes	Method	void RemoveIndexes(), void RemoveIndexes(System. ↵Object index)
RemoveSelector	Method	void RemoveSelector(System.Object key)
RemoveSort	Method	void RemoveSort(), void RemoveSort(System.Object sort)
ReplaceSelector	Method	void ReplaceSelector(System.Object key, System.Object ↵ ↵value)
SetBookmark	Method	void SetBookmark(System.Object bookmark)
SetExecutionStat	Method	void SetExecutionStat(System.Object bool)
SetLimit	Method	void SetLimit(System.Object limit)
SetReadQuorum	Method	void SetReadQuorum(System.Object r)
SetSkip	Method	void SetSkip(System.Object skip)
SetStable	Method	void SetStable(System.Object bool)
SetStale	Method	void SetStale()
ToString	Method	string ToString()

14.1.3 Build a query

To create a PSCouchDBQuery object, just do the following.

```
using module PSCouchDB
$query = New-Object -TypeName PSCouchDBQuery
$query.GetType()
```

14.1.4 Work with selector

A CouchDB query is interpreted by Mango engine, so must have some elements. *Selector* is first our element which allows to have a search criterion.

Add selector

To add one selector to object using *AddSelector* method.

```
$query.AddSelector('key', 'value')
```

The search criterion and its exact key and your value. Now, to verify our query, just get the json, with this method *GetNativeQuery*.

```
$query.GetNativeQuery()
```

Remove selector

If we were wrong to enter the values, it will be enough to remove them with *RemoveSelector* and then insert them again.

```
$query.RemoveSelector('key')
$query.AddSelector('answer', 42)
```

Replace selector

Instead if we were wrong to enter only the value of our search key, just do a replace, using the *ReplaceSelector* method

```
$query.ReplaceSelector('answer', 43)
```

14.1.5 Limit and Skip

To limit or skip line of query result, set the values with the appropriate methods, *SetLimit* and *SetSkip*.

```
$query.SetLimit(5)
$query.SetSkip(1)
```

To remove the set values, just set them to null.

```
$query.limit = $null
$query.skip = $null
```

14.1.6 Sort

To add a sort criterion, use the *AddSortAsc* method for ascending and *AddSortDesc* for the descendant.

```
$query.AddSortAsc('answer')
$query.AddSortDesc('answer')
```

To reset the sort, just remove sorting with *RemoveSort*.

```
$query.RemoveSort()
$query.RemoveSort('answer')
```

14.1.7 Fields

Fields are the values that return from the query. To add them with *AddFields*.

```
$query.AddFields('_id')
$query.AddFields('_rev')
$query.AddFields('answer')
```

To remove all fields use *RemoveFields*.

```
$query.RemoveFields()
```

To remove manually one or more fields.

```
$query.RemoveFields('_rev')
```

14.1.8 Indexies

To configure indexes created previously with `New-CouchDBIndex`.

```
$query.AddIndexies('test-index')
```

To remove all indexes or one.

```
$query.RemoveIndexies()  
$query.RemoveIndexies('test-index')
```

14.1.9 ReadQuorum

To set *ReadQuorum*.

```
$query.SetReadQuorum(2)
```

To remove it.

```
$query.r = $null
```

14.1.10 Bookmark

To configure bookmarks created previously.

```
$query.SetBookmark('mybookmark')
```

To remove it.

```
$query.bookmark = $null
```

14.1.11 Update, Stable and Stale

Update is enabled by default. To disable it.

```
$query.DisableUpdate()
```

To re-enabled it.

```
$query.update = $true
```

To enable *stable*.

```
$query.SetStable($true) #or $query.SetStable(1)
```

To disable it.

```
$query.SetStable($false) #or $query.SetStable(0)
```

stale properties, basically sets *update* to false and *stable* to true.

```
$query.SetStale()
```

To restore all changes.

```
$query.update = $true
$query.stable = $false
$query.stale = $null
```

14.1.12 ExecutionStat

To return execution statistic, just enable it.

```
$query.SetExecutionStat($true) #or $query.SetExecutionStat(1)
```

To disable it.

```
$query.SetExecutionStat($false) #or $query.SetExecutionStat(0)
```

14.1.13 Selector Operators

The selector operators that can be used are the following: *\$lt*, *\$lte*, *\$eq*, *\$ne*, *\$gte*, *\$gt*, *\$exists*, *\$type*, *\$in*, *\$nin*, *\$size*, *\$mod*, *\$regex* (see [operator table](#)). The method *AddSelectorOperator* works in two ways: by specifying only the operator, so it will be applied to all the selector; by specifying the selector and the value that you want to associate.

\$lt*, *\$lte*, *\$eq*, *\$ne*, *\$gte*, *\$gt

The implicit operator used is *\$eq*. The *AddSelectorOperator* method append operators at the designated selector.

```
$query.AddSelectorOperator('$eq')
Find-CouchDBDocuments -Database test -Find $query.GetNativeQuery()
```

To change operator or restore changes.

```
$query.ReplaceSelector('answer', 42) #to restore only this
$query.AddSelectorOperator('$lt')
Find-CouchDBDocuments -Database test -Find $query.GetNativeQuery()
```

\$exists*, *\$type*, *\$in*, *\$nin*, *\$size*, *\$mod

With these operators we must also specify the selector we want and its value.

```
$query.AddSelectorOperator('$exists', 'answer', 'true')
#or
$query.AddSelectorOperator('$type', 'answer', 'string')
#or
$query.AddSelector('name', 'Arthur')
$query.AddSelector('planet', @('Heart', 'Magrathea'))
$query.AddSelectorOperator('$in', 'planet', 'Magrathea')
```

(continues on next page)

(continued from previous page)

```
#or
$query.AddSelectorOperator('$nin','planet','Vogsphere')
#or apply operator for all selector
$query.ReplaceSelector('answer',43)
$query.ReplaceSelector('name','Arthur')
$query.ReplaceSelector('planet',@('heart','magrathea'))
$query.AddSelectorOperator('$in')
Find-CouchDBDocuments -Database test -Find $query.GetNativeQuery()
```

\$regex

CouchDB support regular expression (BRE and ERE).

```
$query.AddSelector('name','Arthur')
$query.AddSelector('planet',@('Heart','Magrathea'))
$query.AddSelectorOperator('$regex','name','^[Aa]r{1}[th]{2}.r$')
```

14.1.14 Logical operators

PSCouchDBQuery object support logical operators; these are the allowed operators: `$and`, `$or`, `$not`, `$nor`, `$all`, `$elemMatch`, `$allMatch` (see [logical operator table](#)).

\$and,\$or,\$not,\$nor

With method *AddLogicalOperator* logical conditions can be added.

```
$query.AddSelector('answer',43)
$query.AddSelector('name','Arthur')
$query.AddSelector('planet',@('Heart','Magrathea'))
$query.AddLogicalOperator('$or')
Find-CouchDBDocuments -Database test -Find $query.GetNativeQuery()
```

\$all,\$elemMatch,\$allMatch

With these logical operators, return a single or all matches.

```
$query.AddSelector('name','Arthur')
$query.AddLogicalOperator('$elemMatch')
Find-CouchDBDocuments -Database test -Find $query.GetNativeQuery()
```

14.1.15 Native query format (Mango)

To receive the object in native format (Mango query) use the *GetNativeQuery* method.

. code-block:: powershell

```
$query.GetNativeQuery()
```

14.2 PSCouchDBDesignDoc class

This class is used to construct a design documents, simple or complex.

14.2.1 Properties

lists	Property	hashtable lists {get;set;}
shows	Property	hashtable shows {get;set;}
validate_doc_update	Property	string validate_doc_update {get;set;}
views	Property	hashtable views {get;set;}

14.2.2 Methods

AddList	Method	void AddList(System.Object name)
AddShow	Method	void AddShow(System.Object name), void AddShow(System.Object name, System.Object key)...
AddValidation	Method	void AddValidation(array requirements), void AddValidation(array requirements, System.Object key)...
AddView	Method	void AddView(System.Object name), void AddView(System.Object name, System.Object key)...
Equals	Method	bool Equals(System.Object obj)
GetDesignDocuments	Method	string GetDesignDocuments()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
SetName	Method	void SetName(System.Object name)
ToString	Method	string ToString()

14.2.3 Build a design document

To create a PSCouchDBDesignDoc object, just do the following.

```
using module PSCouchDB
$ddoc = New-Object -TypeName PSCouchDBDesignDoc
$ddoc.GetType()
```

14.2.4 Work with views

Views are the primary tool used for querying and reporting on CouchDB documents. With *AddView* it is possible to add four types of predefined views.

Simple view

```
$ddoc.AddView('myview')
```

View if document key exists

```
$ddoc.AddView('myviewexists','key')
```

View if document key exists and your value is value specified

```
$ddoc.AddView('myviewvalue','key','value')
```

View if document key exists and your value is value specified, and return entire doc

```
$ddoc.AddView('myviewdoc','key','value',$true)
```

14.2.5 Work with Lists

List functions are used to represent documents in various formats, commonly as HTML pages with nice formatting. Use *AddList* for add one list or more. List functions they need at least one view.

```
$ddoc.AddList("mylist")
```

14.2.6 Work with Shows

With *AddShow* it is possible to add three types of predefined show function. Only one function is allowed at a time.

Simple show

```
$ddoc.AddShow("myshow")
```

Show if document key exists

```
# reset object
$ddoc = New-Object -TypeName PSCouchDBDesignDoc
$ddoc.AddShow("myshow","key")
```

Show if document key exists and your value is value specified

```
# reset object
$ddoc = New-Object -TypeName PSCouchDBDesignDoc
$ddoc.AddShow("myshow","key","value")
```

14.2.7 Work with validation

A design document may contain a function named `validate_doc_update` which can be used to prevent invalid or unauthorized document update requests from being stored. Use *AddValidation* for add one. Only one function is allowed at a time.

```
$ddoc.AddValidation("myvalidation")  
#or add also user authorization  
$ddoc.AddValidation("myvalidation",$true)  
#or create ReadOnly CouchDB database  
$ddoc.AddValidation($true)
```

14.2.8 Native design document

To receive the design document in native format use the *GetDesignDocuments* method.

```
$ddoc.GetDesignDocuments()
```

14.2.9 Create design document

See [Create design document](#).

```
New-CouchDBDesignDocument -Database test -Document "mydesigndoc" -Data $ddoc.  
↪GetDesignDocuments() -Authorization "admin:password"
```


Below are some examples of real-world applications that can give the idea of the module's potential together with CouchDB.

15.1 Simple machine inventory

This is a simple hardware inventory in business scenario, build in three steps.

First step, create database.

```
New-CouchDBDatabase -Database hw_inventory
```

Second, edit a new file *hw_inventory.ps1* e paste this:

```
# Create an Active Directory session
$session = New-PSSession -ComputerName "your_domain_controller.local"
Import-PSSession -Session $session -module ActiveDirectory
# Get all computer
$AllComputers = Get-ADComputer -Filter * -Properties DNSHostName
foreach ($ComputerName in $AllComputers.DNSHostName) {
    $info = @{}
    # Test connection of computer
    if (Test-Connection $ComputerName -Count 1 -Quiet) {
        # Select various info
        $info.Add('ComputerHW', (Get-CimInstance -Class Win32_ComputerSystem -
↪ComputerName $ComputerName |
        select Manufacturer,
            Model,
            NumberOfProcessors,
            @{Expression={$_.TotalPhysicalMemory / 1GB};Label=
↪"TotalPhysicalMemoryGB"}))
        $info.Add('ComputerCPU', (Get-CimInstance win32_processor -ComputerName
↪$ComputerName |
```

(continues on next page)

(continued from previous page)

```

        select DeviceID,
               Name,
               Manufacturer,
               NumberOfCores,
               NumberOfLogicalProcessors))
        $info.Add('ComputerDisks', (Get-CimInstance -Class Win32_LogicalDisk -Filter
↪ "DriveType=3" -ComputerName $ComputerName |
        select DeviceID,
               VolumeName,
               @{Expression={$_.Size / 1GB};Label="SizeGB"})
        $info.Add("timestamp", (Get-Date -f MM-dd-yyyy_HH-mm-ss))
        # Write on database
        if (Get-CouchDBDocument -Database hw_inventory -Document $ComputerName -
↪ ErrorAction SilentlyContinue) {
            Set-CouchDBDocument -Database hw_inventory -Document $ComputerName -Data
↪ $info -Revision $(Get-CouchDBDocument -Database hw_inventory -Document
↪ $ComputerName)._rev -Replace
        } else {
            New-CouchDBDocument -Database hw_inventory -Document $ComputerName -Data
↪ $info
        }
    }
}
Get-PSSession | Remove-PSSession

```

Third, edit your *profile.ps1* and put this function:

```

# Find computer into inventory
function Find-ComputerInventory () {
    param(
        [Parameter(mandatory=$true)]
        [string] $Computername
    )
    $docs = Find-CouchDBDocuments -Database hw_inventory -Selector "_id" -Operator_
↪ regex -Value ".$Computername.*" -Fields _id,ComputerHW,ComputerCPU,ComputerDisks,
↪ timestamp
    $docs.docs
}

```

Schedule the script every six hours.

15.2 Log storage

To create an application that stores logs from various machines in two steps.

First, edit your *profile.ps1* and put this function:

```

function Write-WindowsLog () {
    param(
        [Parameter(mandatory=$true)]
        [string] $ComputerName,
        [string] $Authorization
    )
    # Define logs
    $Logs = @("Application", "Security", "System")

```

(continues on next page)

(continued from previous page)

```

# Loop foreach log
foreach ($Log in $logs) {
    $count = 0
    $DBName = "${ComputerName}_$Log".ToLower()
    # Test if database log exists
    if ($null -eq (Test-CouchDBDatabase -Database $DBName -ErrorAction_
↪SilentlyContinue)) {
        New-CouchDBDatabase -Database $DBName -Authorization $Authorization
    }
    # Get log
    $LogList = Get-EventLog -LogName $Log -ComputerName $ComputerName
    $LogList | foreach {
        $count++
        # Write on database
        if (-not (Get-CouchDBDocument -Database $DBName -Document $_.Index -
↪ErrorAction SilentlyContinue)) {
            New-CouchDBDocument -Database $DBName -Document $_.Index -Data ($_ |_
↪Convertto-Json -Depth 10) -Authorization $Authorization | Out-Null
        }
        Write-Progress -Activity "Write log $Log in progress" -Status "Progress
↪$count/$(($LogList.count))" -PercentComplete ($count/$LogList.count*100)
    }
}
}

```

Second, edit again your *profile.ps1* and put this function:

```

using module PSCouchDB
# Find log with criteria
function Find-WindowsLog () {
    param(
        [Parameter(mandatory=$true)]
        [string] $ComputerName,
        [Parameter(mandatory=$true)]
        [ValidateSet("Application", "Security", "System")]
        [string] $Log,
        $SearchCriteria
    )
    # Check if criteria is a string or a int
    if ($SearchCriteria.GetType() -eq [int]) {
        $SearchCriteria = [int]$SearchCriteria
    }
    # Create Mango query
    $q = New-Object -TypeName PSCouchDBQuery
    $q.AddSelector("CategoryNumber", $SearchCriteria)
    $q.AddSelector("EventID", $SearchCriteria)
    $q.AddSelector("Message", "")
    $q.AddSelector("UserName", "")
    $q.AddSelector("Source", "")
    $q.AddSelectorOperator('$regex', "Message", ".$SearchCriteria.*")
    $q.AddSelectorOperator('$regex', "UserName", ".$SearchCriteria.*")
    $q.AddSelectorOperator('$regex', "Source", ".$SearchCriteria.*")
    $q.AddLogicalOperator('$or')
    $q.AddFields("_id")
    $q.AddFields("MachineName")
    $q.AddFields("Data")
    $q.AddFields("Index")
}

```

(continues on next page)

(continued from previous page)

```
$q.AddFields("CategoryNumber")
$q.AddFields("EventID")
$q.AddFields("EntryType")
$q.AddFields("Message")
$q.AddFields("Source")
$q.AddFields("ReplacementStrings")
$q.AddFields("InstanceId")
$q.AddFields("UserName")
$docs = Find-CouchDBDocuments -Database "${ComputerName}_" -Log ($Log.ToLower()) -Find
↪ $q.GetNativeQuery()
    $docs.docs
}
```

Note: With PSCouchDB, with just a few lines of code, you can create simple applications for complex tasks.

This module is open source, under [GPLv3](#) license. Anyone who wants, passion, respect and love can contribute to this project, for himself and for all of humanity.

16.1 Contribute

If you wanted to contribute to this powershell module, [fork](#).

16.2 Issue

If you have found a bug or have an improvement in mind, open an issue.

16.3 Donations

Donating is important. If you do not want to do it to me, do it to some companies that do not speculate. My main activity and the people of non-profit associations is to work for others, be they male or female, religious or non-religious, white or black or yellow or red, rich and poor. The only real purpose is to serve the humanity of one's experience. Below you will find some links to do it. **Thanks a lot.**

For me

For [Telethon](#)

The Telethon Foundation is a non-profit organization recognized by the Ministry of University and Scientific and Technological Research. They were born in 1990 to respond to the appeal of patients suffering from rare diseases. Come today, we are organized to dare to listen to them and answers, every day of the year.

[Adopt the future](#)