

---

# **PSCouchDB Documentation**

**Matteo Guadrini**

**Aug 26, 2021**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Signing . . . . .	4
1.3	Start . . . . .	4
<b>2</b>	<b>Setup</b>	<b>5</b>
2.1	Single Node . . . . .	5
2.2	Cluster . . . . .	5
<b>3</b>	<b>Nodes</b>	<b>7</b>
3.1	Add a node . . . . .	7
3.2	Remove a node . . . . .	7
<b>4</b>	<b>Configuration</b>	<b>9</b>
4.1	Get configuration . . . . .	9
4.2	Modify configuration . . . . .	9
<b>5</b>	<b>Permission</b>	<b>11</b>
5.1	Admin Party . . . . .	11
5.2	Create Admin user . . . . .	11
5.3	Members access . . . . .	11
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
<b>6</b>	<b>Server operation</b>	<b>15</b>
6.1	Read the log . . . . .	17
6.2	Clear the log . . . . .	18
<b>7</b>	<b>Replication</b>	<b>19</b>
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
7.6	Enable/Disable Maintenance . . . . .	20
<b>8</b>	<b>Authentication</b>	<b>21</b>
<b>9</b>	<b>Databases</b>	<b>23</b>
9.1	Connect database . . . . .	23
9.2	Test a database . . . . .	23
9.3	Read a database . . . . .	23
9.4	Create a database . . . . .	24
9.5	Remove a database . . . . .	24
9.6	Copy a database . . . . .	24
9.7	Index . . . . .	24
9.8	Shards . . . . .	25
9.9	Changes . . . . .	25
9.10	Compact . . . . .	26
9.11	Write a commit . . . . .	26
9.12	Clear view . . . . .	26
9.13	Get purged info limit . . . . .	26
9.14	Set purged info limit . . . . .	26
9.15	Revisions . . . . .	26
9.16	Export and import databases . . . . .	27
9.17	Partition database . . . . .	28
<b>10</b>	<b>Documents</b>	<b>29</b>
10.1	Get a document . . . . .	29
10.2	Get partitioned documents . . . . .	29
10.3	Create a document . . . . .	31
10.4	Modify a document . . . . .	32
10.5	Delete a document . . . . .	32
10.6	Copy a document . . . . .	32
10.7	Local document . . . . .	33
10.8	Get a bulk documents . . . . .	33
10.9	Create documents in bulk . . . . .	33
10.10	Attachments . . . . .	34
10.11	Revisions . . . . .	34
10.12	Purge document . . . . .	35
<b>11</b>	<b>Query</b>	<b>37</b>
11.1	Find a document . . . . .	37
<b>12</b>	<b>Design documents</b>	<b>45</b>
12.1	Get a design document . . . . .	45
12.2	Creates a design document . . . . .	46
12.3	Compress design document . . . . .	48
12.4	Remove design document . . . . .	48
<b>13</b>	<b>Preferences</b>	<b>49</b>
<b>14</b>	<b>Cmdlets and aliases</b>	<b>51</b>
14.1	Cmdlets . . . . .	51
14.2	Aliases . . . . .	62
<b>15</b>	<b>Classes</b>	<b>65</b>

15.1	Create an PSCouchDB object	65
15.2	PSCouchDBRequest class	65
15.3	PSCouchDBQuery class	69
15.4	PSCouchDBDocument class	75
15.5	PSCouchDBAttachment class	77
15.6	PSCouchDBBulkDocument class	78
15.7	PSCouchDBSecurity class	79
15.8	PSCouchDBReplication class	81
15.9	PSCouchDBView class	83
15.10	PSCouchDBDesignDoc class	85
<b>16</b>	<b>Real uses</b>	<b>89</b>
16.1	Scripts	89
16.2	OOP	92
<b>17</b>	<b>Support</b>	<b>95</b>
17.1	Contribute	95
17.2	Issue	95
17.3	Donations	95



## Advanced CLI for CouchDB server

---

**Note:** If you are using CouchDB version 2, use the PSCouchDB 1.X version; if instead you are using CouchDB version 3 or 4, use the PSCouchDB version 2.X

---



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: powershell 5.1
copy /Y PSCouchDB %Windir%\System32\WindowsPowerShell\v1.0\Modules\PSCouchDB
# for Windows: powershell core 6 or high
copy /Y PSCouchDB %ProgramFiles%\WindowsPowerShell\Modules\PSCouchDB
# 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 3.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.

### 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 -Node 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

Starting from CouchDB 3.X, during installation and configuration, an administrator user must be specified. This effectively breaks the old Admin Party logic of the previous version.

## 5.2 Create Admin user

At this point the admin user can create/modify/delete documents from the database.

---

**Important:** The `-Authorization` parameter, accept two format: string and `PSCredential`. The string must be in this format: `user:password`. If you don't want the password to be displayed inside the terminal, this is the form of the parameter: `-Authorization (Get-Credential)`. See this section: [Permission](#)

---

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

```
using module PSCouchDB
$sec = New-Object PSCouchDBSecurity
$sec.AddMembers('member_user')
Grant-CouchDBDatabasePermission -Database test -Data $sec -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
$read_only = @"
function(newDoc, oldDoc, userCtx, secObj) {
    if (userCtx.roles.indexOf('admin') !== -1) {
        return;
    } else {
        throw({forbidden: "Only admin can edit documents!"})
    }
}
"@
$ddoc.SetValidateFunction($read_only)
New-CouchDBDesignDocument -Database test -Document "mydesigndoc" -Data $ddoc -
↪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:

```
using module PSCouchDB
$password = "password" | ConvertTo-SecureString -AsPlainText -Force
New-CouchDBUser -Userid other_admin -Password $password -Authorization "admin:password"
↪"
$sec = New-Object PSCouchDBSecurity -ArgumentList 'other_admin'
Grant-CouchDBDatabasePermission -Database test -Data $sec -Authorization
↪"admin:password"
Get-CouchDBDatabase -Database test -Authorization "other_admin:password"
```

## 5.6 Revoke database permissions

To remove all 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 or 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 or reset password of an admin.

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



## CHAPTER 6

---

### Server operation

---

Most of the cmdlets in this module are based on the `Send-CouchDBRequest` cmdlet.

```
Send-CouchDBRequest # alias creq  
Get-Help Send-CouchDBRequest -Full
```

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 -Authorization "admin:password"
```

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

```
Get-CouchDBDatabaseInfo -Keys test,test1,test2 -Authorization "admin:password"
```

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	<p><b>Option: normal:</b> Returns all historical DB changes, then closes the connection. Default.</p> <p><b>longpoll:</b> Closes the connection after the first event.</p> <p><b>continuous:</b> Send a line of JSON per event. Keeps the socket open until timeout.</p> <p><b>eventsourc</b>e: Like, continuous, but sends the events in EventSource format.</p>
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 eventsourc 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
```

To set proxy server for all calls:

```
Set-CouchDBProxy -Server 'http://myproxy.local:8080' -Credential (Get-Credential)
```

And remove it

```
Remove-CouchDBProxy
```

Tests the results of Lucene analyzer tokenization on sample text:

```
Search-CouchDBAnalyze -Field "english" -Text "running" -Authorization "admin:password"
```

Returns a count of completed, failed, running, stopped, and total jobs along with the state of resharding on the cluster:

```
Get-CouchDBReshards -Jobs -Authorization "admin:password"
```

This starts global resharding on all the nodes of the cluster:

```
Set-CouchDBReshards -State running -StateReason "Test start" -Authorization
↪ "admin:password"
```

Single resharding job for a particular range:

```
Set-CouchDBReshards -Database test -Err "Test message" -Type split -Range "80000000-
↪ ffffffff" -Authorization "admin:password"
```

Stop and remove specific job id:

```
Remove-CouchDBReshards -JobId "001-
↪ 638b90b9acf73cbb113afdfdba458bec80da6a6be23599019fb7b051cedfcc93" -Authorization
↪ "admin:password"
```

## 6.1 Read the log

To read entire log.

```
Read-CouchDBLog -Authorization "admin:password"
```

**Note:** The default path is specified in the server configuration. Run “ Get-CouchDBConfiguration -Session log -Key file -Authorization admin:password“. Otherwise, specify the path using the -Path parameter.

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

```
Read-CouchDBLog -Tail 15 -Authorization "admin:password"
```

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

```
Read-CouchDBLog -Level warning -Follow -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

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.

```
using module PSCouchDB
$rep = New-Object PSCouchDBReplication -ArgumentList 'test','test_dump'
$rep.SetContinuous()
New-CouchDBReplication -Data $rep -Authorization "admin:password"
```

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

```
Get-CouchDBDatabaseChanges -Database test_dump -Authorization "admin:password"
```

## 7.3 Modify replica

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

```
using module PSCouchDB
$rep = New-Object PSCouchDBReplication -ArgumentList 'test','test_dump'
$rep.SetRevision("4-c2cefa18494e47182a125b11eccecd13")
Set-CouchDBReplication -Data $rep -Authorization "admin:password"
```

## 7.4 Remove replica

To remove the replication agent.

```
Remove-CouchDBReplication -Document test_test_dump -Authorization "admin:password"
```

## 7.5 Replication request

Request, configure, or stop, a replication operation.

```
using module PSCouchDB
$rep = New-Object PSCouchDBReplication -ArgumentList 'test','test_dump'
$rep.AddDocIds(@("Hitchhikers","Hitchhikers_Guide"))
Request-CouchDBReplication -Data $rep -Authorization "admin:password"
```

## 7.6 Enable/Disable Maintenance

Enable maintenance mode.

```
Set-CouchDBMaintenanceMode -Authorization "admin:password"
```

Disable maintenance mode.

```
Set-CouchDBMaintenanceMode -Maintenance $false -Authorization "admin:password"
```

## CHAPTER 8

---

### Authentication

---

For security reasons, PSCouchDB does not use CouchDB token cookies. But authentication can be stored inside the current powershell session. By default, credential storage is allowed so that the `Authorization` parameter can only be specified once.

To modify the saved credential preference, set this:

```
$CouchDBSaveCredentialPreference = $false # default is $true
```

Set a cookie:

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

Get a session:

```
Get-CouchDBSession # PSCredential
Get-CouchDBSession -Clear # string format user:password
```

Now let verify a protected read database without `Authorization` param:

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

And remove a session:

```
Remove-CouchDBSession
```



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

## 9.1 Connect database

To permanently connect to a database, do the following:

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

and disconnect:

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

## 9.2 Test a database

To verify the existence of a database.

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

## 9.3 Read a database

Gets information about the specified database.

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

To get alist 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 0.
StartKey	Return databases starting with the specified key.

## 9.4 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

## 9.5 Remove a database

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

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

## 9.6 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"
```

## 9.7 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.

### 9.7.1 Get a index

To get list of index present on a database.

```
Get-CouchDBIndex -Database test -Authorization "admin:password"
```

### 9.7.2 Create a new index

Create a new index on a database.

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

### 9.7.3 Remove a index

Remove an existing index.

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

## 9.8 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.

### 9.8.1 Get shards

Get a list a database shards.

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

Get the shard document on database.

```
Get-CouchDBDatabaseShards -Database test -Document 00000000-1fffffff -Authorization  
↪ "admin:password"
```

### 9.8.2 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"
```

## 9.9 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 -Authorization "admin:password"
```

## 9.10 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"
```

## 9.11 Write a commit

Commits any recent changes to the specified database to disk.

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

## 9.12 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"
```

## 9.13 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 -Authorization "admin:password"
```

## 9.14 Set purged info limit

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

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

## 9.15 Revisions

Working with database revisions.

### 9.15.1 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 -Authorization
↪"admin:password"
```

### 9.15.2 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 -Authorization
↪"admin:password"
```

### 9.15.3 Get revision limit

Gets the current `revs_limit` (revision limit) setting.

```
Get-CouchDBRevisionLimit -Database test -Authorization "admin:password"
```

### 9.15.4 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"
```

## 9.16 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.

### 9.16.1 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 -Authorization "admin:password"
```

## 9.16.2 Import

For import or restore a database from JSON file.

```
Import-CouchDBDatabase -Database test -Path test_01-25-2019_00_01_00.json -  
↔Authorization "admin:password"
```

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 -Authorization "admin:password"
```

## 9.17 Partition database

To create a partitioned database, we simply run:

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

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" -Authorization  
↔ "admin:password"
```

### 10.2 Get partitioned documents

```
Get-CouchDBDocument -Database test -Partition testing -Authorization "admin:password"
```

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.
AttachmentsInfo	Includes encoding information in attachment stubs if the particular attachment is compressed.
AttachmentsSince	Includes attachments only since specified revisions. Doesn't includes attachments for specified revisions.
Conflicts	Includes information about conflicts in document.
DeletedConflicts	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.
Partition	CouchDB partition name

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&amp;update=false</code> . <code>update_after</code> is equivalent to <code>stable=true&amp;update=lazy</code> . <code>false</code> is equivalent to <code>stable=false&amp;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> .

### 10.3 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.

---

## 10.4 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"  
↪ "
```

## 10.5 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"
```

## 10.6 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"
```

## 10.7 Local document

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

```
Get-CouchDBDocument -Database test -Local -Authorization "admin:password"
```

## 10.8 Get a bulk documents

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

```
using module PSCouchDB
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList '{"_id":"test"}'
$bdocs.AddDocument('{"_id":"test1","_rev":"2-9a68ee74a8276c7f11146245ba43676f"}')
Get-CouchDBBulkDocument -Database test -Data $bdocs -Authorization "admin:password"
```

or run in background:

```
using module PSCouchDB
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList '{"_id":"test"}'
$bdocs.AddDocument('{"_id":"test1","_rev":"2-9a68ee74a8276c7f11146245ba43676f"}')
Get-CouchDBBulkDocument -Database test -Data $bdocs -Authorization "admin:password" -
↳AsJob
Get-Job -Id 1 | Receive-Job -Keep
```

## 10.9 Create documents in bulk

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

```
using module PSCouchDB
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList '{"_id":"test","name":"test"}'
$bdocs.AddDocument('{"_id":"test1","name":"test"}')
New-CouchDBBulkDocument -Database test -Data $bdocs -Authorization "admin:password"
```

or run in background:

```
using module PSCouchDB
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList '{"_id":"test","name":"test"}'
$bdocs.AddDocument('{"_id":"test1","name":"test"}')
New-CouchDBBulkDocument -Database test -Data $bdocs -Authorization "admin:password" -
↳AsJob
Get-Job -Id 1 | Receive-Job -Keep
```

## 10.10 Attachments

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

### 10.10.1 Get an attachment

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

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt -  
↔Authorization "admin:password"
```

Also is possible save a file.

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt -  
↔OutFile "C:\out.txt" -Authorization "admin:password"
```

Or get info of specific attachment.

```
Get-CouchDBAttachment -Database test -Document "Hitchhikers" -Attachment test.txt -  
↔Info -Authorization "admin:password"
```

### 10.10.2 Create an attachment

To replace or add an attachment.

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

### 10.10.3 Delete an attachment

To remove an attachment.

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

## 10.11 Revisions

### 10.11.1 Get a list of revisions

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

```
Get-CouchDBDocument -Database test -Document "Hitchhikers" -Revisions -Authorization  
↔"admin:password"
```

### 10.11.2 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 -Authorization  
↪ "admin:password"
```

### 10.11.3 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" -Authorization "admin:password"
```

### 10.11.4 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"
```

## 10.12 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 -Authorization "admin:password"
```

or with native Mango query

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

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() -Authorization
↪ "admin:password"
```

or search partitioned documents in a database, use the following cmdlet.

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

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" -Authorization  
↪ "admin:password"
```

**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": {
      "$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() -Authorization
↪ "admin:password"
```

```
Find-CouchDBDocuments -Database test -Selector "name" -Operator eq -Value "Arthur Dent"
↔ -Fields _id,name,planet -Authorization "admin:password"
```

**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 -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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" -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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" -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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' -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```

### 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 -Authorization "admin:password"
```



# CHAPTER 12

---

## 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.

**Warning:** Show and List functions are deprecated in CouchDB 3.0, and will be removed in CouchDB 4.0.

### 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" -Authorization  
↔ "admin:password"
```

To get all the Design Documents in a database.

```
Get-CouchDBDatabaseDesignDocument -Database test -Authorization "admin:password"
```

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 test -Document space -Attachment test.
↔txt -Authorization "admin:password" # Get content
Get-CouchDBDesignDocumentAttachment -Database test -Document space -Attachment test.
↔txt -OutFile "C:\test.txt" -Authorization "admin:password" # Save content
```

## 12.2 Creates a design document

Creates a new named design document.

```
New-CouchDBDesignDocument -Database test -Document "space" -ViewName "planet_view" -
↔ViewMapFunction "function(doc){if(doc.planet && doc.name) {emit(doc.planet, doc.
↔name);}}" -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_view" -
↪ViewMapFunction "function(doc){if(doc.planet && doc.name) {emit(doc.planet, doc.
↪name);}}" -Authorization "admin:password"
```

Now, navigate with your favorite browser to [http://localhost:5984/test/\\_design/space/\\_view/planet\\_view](http://localhost:5984/test/_design/space/_view/planet_view) or

```
Get-CouchDBDocument -Database test -Document "_design/space/_view/planet_view" -
↪Authorization "admin:password"
```

## 12.2.2 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.

```
Set-CouchDBDesignDocument -Database test -Document "space" -Revision "1-
↪88972423aac3fe5d474dd17d3ee18a8b" -ValidationFunction "function(newDoc, oldDoc,
↪userCtx, secObj){if (!(newDoc.name || newDoc.planet)) {throw({forbidden : 'no way'}
↪);}" -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":"no way"}`. Now retry with this:

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

## 12.2.3 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": {
```

(continues on next page)

(continued from previous page)

```
        "planet": "function(doc, req) { return '<h1>' + doc.title + '</h1>' }"
    }
}
'@
New-CouchDBDesignDocument -Database test -Document space -Data $ddoc -Authorization
↪ "admin:password"
```

## 12.2.4 Create design document attachment

To create an attachment in a design document.

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

## 12.2.5 Modify design document attachment

To modify or add an attachment in a design document.

```
Add-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"
```

# CHAPTER 13

---

## Preferences

---

Below a list of variable preferences of module

```
$CouchDBCachePreference      # default value: $false  
$CouchDBSaveCredentialPreference # default value: $true
```



---

## Cmdlets and aliases

---

Below is a list of cmdlets and aliases

### 14.1 Cmdlets

#### 14.1.1 Help

##### Search-CouchDBHelp

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

##### New-CouchDBObject

```
New-CouchDBObject [[-TypeName] <String>] [[-ArgumentList] <Array>] [<CommonParameters>]
↪]
```

#### 14.1.2 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] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [
↪<CommonParameters>]
```

##### Get-CouchDBNode

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

### Add-CouchDBNode

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

### Remove-CouchDBNode

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

### Get-CouchDBConfiguration

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

### Set-CouchDBConfiguration

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

## 14.1.3 Permission

### New-CouchDBAdmin

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

### New-CouchDBUser

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

### Grant-CouchDBDatabasePermission

```
Grant-CouchDBDatabasePermission [[-Server] <String>] [[-Port] <Int32>] [-Database]  
↪<String> [[-Data] <Object>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer]  
↪<String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Get-CouchDBDatabaseSecurity

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

### Revoke-CouchDBDatabasePermission

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

### Remove-CouchDBAdmin

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

### Remove-CouchDBUser

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

### Set-CouchDBUser

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

### Set-CouchDBAdmin

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

## 14.1.4 Server

### Send-CouchDBRequest

```
Send-CouchDBRequest [[-Method] <String>] [[-Server] <String>] [[-Port] <Int32>] [[-
↪Database] <String>] [[-Document] <String>] [[-Authorization] <Object>] [[-Revision]
↪<String>] [[-Attachment] <String>] [[-Data] <String>] [[-Params] <Array>] [-Ssl] [[-
↪JobName] <String>] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [
↪<CommonParameters>]
```

### Set-CouchDBProxy

```
Set-CouchDBProxy [-Server] <String> [[-Credential] <PSCredential>] [<CommonParameters>
↪]
```

### Remove-CouchDBProxy

```
Remove-CouchDBProxy [<CommonParameters>]
```

### Get-CouchDBServer

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

### Get-CouchDBActiveTask

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

### Get-CouchDBClusterSetup

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

### Get-CouchDBDatabaseUpdates

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

### Measure-CouchDBStatistics

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

### Restart-CouchDBServer

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

### New-CouchDBUuids

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

### Read-CouchDBLog

```
Read-CouchDBLog [[-Path] <String>] [[-Level] <String>] [-Follow] [[-Tail] <Int32>] [[-
↪Authorization] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>]
↪<Object>] [<CommonParameters>]
```

### Clear-CouchDBLog

```
Clear-CouchDBLog [[-Path] <String>] [-Rotate] [[-Authorization] <Object>] [-Ssl] [[-
↪ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Search-CouchDBAnalyze

```
Search-CouchDBAnalyze [[-Server] <String>] [[-Port] <Int32>] [-Field] <String> [-
↪Text] <String> [[-Authorization] <Object>] [-Ssl] [[-ProxyServer] <String>] [[-
↪ProxyCredential] <PSCredential>] [<CommonParameters>]
```

### Get-CouchDBReshards

```
Get-CouchDBReshards [-Server <String>] [-Port <Int32>] [-Jobs] [-Authorization
↪<Object>] [-Ssl] [-ProxyServer <String>] [-ProxyCredential <PSCredential>] [
↪<CommonParameters>]
```

(continues on next page)

(continued from previous page)

```
Get-CouchDBReshards [-Server <String>] [-Port <Int32>] [-State] [-JobId <String>] [-
  ↳Authorization <Object>] [-Ssl] [-ProxyServer <String>] [-ProxyCredential
  ↳<PSCredential>] [<CommonParameters>]
```

**Set-CouchDBReshards**

```
Set-CouchDBReshards [-Server <String>] [-Port <Int32>] [-Type <String>] [-Database
  ↳<String>] [-Node <String>] [-Range <String>] [-Shard <String>] [-Err <String>] [-
  ↳Authorization <Object>] [-Ssl] [-ProxyServer <String>] [-ProxyCredential
  ↳<PSCredential>] [<CommonParameters>]
```

```
Set-CouchDBReshards [-Server <String>] [-Port <Int32>] [-State <String>] [-
  ↳StateReason <String>] [-JobId <String>] [-Authorization <Object>] [-Ssl] [-
  ↳ProxyServer <String>] [-ProxyCredential <PSCredential>] [<CommonParameters>]
```

**Remove-CouchDBReshards**

```
Remove-CouchDBReshards [[-Server] <String>] [[-Port] <Int32>] [-JobId] <String> [[-
  ↳Authorization] <Object>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
  ↳<PSCredential>] [<CommonParameters>]
```

**Set-CouchDBMaintenanceMode**

```
Set-CouchDBMaintenanceMode [[-Server] <String>] [[-Port] <Int32>] [[-Node] <String>]
  ↳ [[-Maintenance] <Boolean>] [[-Authorization] <Object>] [-Ssl] [[-ProxyServer]
  ↳<String>] [[-ProxyCredential] <PSCredential>] [<CommonParameters>]
```

## 14.1.5 Replication

**Get-CouchDBReplication**

```
Get-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Database] <String>]
  ↳ [[-Document] <String>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>]
  ↳ [[-ProxyCredential] <Object>] [<CommonParameters>]
```

**Get-CouchDBReplicationScheduler**

```
Get-CouchDBReplicationScheduler [[-Server] <String>] [[-Port] <Int32>] [[-Limit]
  ↳<Int32>] [[-Skip] <Int32>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer]
  ↳<String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

**Get-CouchDBReplicationDocument**

```
Get-CouchDBReplicationDocument [[-Server] <String>] [[-Port] <Int32>] [[-Limit]
  ↳<Int32>] [[-Skip] <Int32>] [[-ReplicatorDatabase] <String>] [[-ReplicatorDocuments]
  ↳<String>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
  ↳ProxyCredential] <Object>] [<CommonParameters>]
```

**New-CouchDBReplication**

```
New-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Data] <Object>] [[-
  ↳Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
  ↳<Object>] [<CommonParameters>]
```

**Get-CouchDBDatabaseChanges**

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

### Set-CouchDBReplication

```
Set-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [-Data] <Object> [[-  
↪Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]  
↪<Object>] [<CommonParameters>]
```

### Remove-CouchDBReplication

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

### Request-CouchDBReplication

```
Request-CouchDBReplication [[-Server] <String>] [[-Port] <Int32>] [[-Data] <Object>] ↵  
↪ [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]  
↪<Object>] [<CommonParameters>]
```

## 14.1.6 Authentication

### Set-CouchDBSession

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

### Get-CouchDBSession

```
Get-CouchDBSession [[-Server] <String>] [[-Port] <Int32>] [[-Authorization] <String>] ↵  
↪ [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Remove-CouchDBSession

```
Remove-CouchDBSession [[-Server] <String>] [[-Port] <Int32>] [[-Authorization]  
↪<String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [  
↪<CommonParameters>]
```

## 14.1.7 Databases

### Connect-CouchDBDatabase

```
Connect-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> ↵  
↪ [[-Authorization] <Object>] [<CommonParameters>]
```

### Disconnect-CouchDBDatabase

```
Disconnect-CouchDBDatabase
```

### Test-CouchDBDatabase

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

### Copy-CouchDBDatabase

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

### Get-CouchDBDatabase

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

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

### New-CouchDBDatabase

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

### Remove-CouchDBDatabase

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

### Get-CouchDBIndex

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

### New-CouchDBIndex

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

### Remove-CouchDBIndex

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

### Get-CouchDBDatabaseInfo

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

### Get-CouchDBDatabaseShards

```
Get-CouchDBDatabaseShards [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Document] <String>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>]
↳ [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Sync-CouchDBDatabaseShards

```
Sync-CouchDBDatabaseShards [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↳ <Object>] [<CommonParameters>]
```

### Compress-CouchDBDatabase

```
Compress-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↳ <Object>] [<CommonParameters>]
```

### Write-CouchDBFullCommit

```
Write-CouchDBFullCommit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Authorization] <String>] [-Force] [-Ssl] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

### Clear-CouchDBView

```
Clear-CouchDBView [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
↳ Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↳ <Object>] [<CommonParameters>]
```

### Get-CouchDBDatabasePurgedLimit

```
Get-CouchDBDatabasePurgedLimit [[-Server] <String>] [[-Port] <Int32>] [-Database]
↳ <String> [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [<CommonParameters>]
```

### Set-CouchDBDatabasePurgedLimit

```
Set-CouchDBDatabasePurgedLimit [[-Server] <String>] [[-Port] <Int32>] [-Database]
↳ <String> [-Limit] <Int32> [[-Authorization] <String>] [-Ssl] [[-ProxyServer]
↳ <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Get-CouchDBMissingRevision

```
Get-CouchDBMissingRevision [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [-Document] <String> [-Revision] <Array> [[-Authorization] <String>] [-Ssl] [[-
↳ ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Get-CouchDBRevisionDifference

```
Get-CouchDBRevisionDifference [[-Server] <String>] [[-Port] <Int32>] [-Database]
↳ <String> [-Document] <String> [-Revision] <Array> [[-Authorization] <String>] [-
↳ Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Get-CouchDBRevisionLimit

```
Get-CouchDBRevisionLimit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↳ <Object>] [<CommonParameters>]
```

(continues on next page)

(continued from previous page)

**Set-CouchDBRevisionLimit**

```
Set-CouchDBRevisionLimit [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
↳ [[-Limit] <Int32>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [<CommonParameters>]
```

**Export-CouchDBDatabase**

```
Export-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [[-
↳ Path] <String>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

**Import-CouchDBDatabase**

```
Import-CouchDBDatabase [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↳ Path] <String> [-RemoveRevision] [[-Authorization] <String>] [-Ssl] [[-ProxyServer]
↳ <String>] [[-ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

**14.1.8 Documents****Get-CouchDBDocument**

```
Get-CouchDBDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↳ Document <String>] [-Partition <String>] [-Revision <String>] [-Local] [-Revisions]
↳ [-History] [-Attachments] [-AttachmentsInfo] [-AttachmentsSince <Array>] [-
↳ Conflicts] [-DeletedConflicts] [-Latest] [-LocalSequence] [-Metadata] [-
↳ OpenRevisions <Array>] [-Authorization <String>] [-Ssl] [[-ProxyServer] <String>]
↳ [[-ProxyCredential] <Object>] [-AsJob] [-Variable] [<CommonParameters>]
```

```
Get-CouchDBDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↳ Document <String>] [-Partition <String>] [-Revision <String>] [-Info] [-Local] [-
↳ Authorization <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↳ <Object>] [<CommonParameters>]
```

```
Get-CouchDBDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↳ AllDocuments] [-Partition <String>] [-Local] [-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] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

**New-CouchDBDocument**

```
New-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↳ Document] <String> [[-Partition] <String>] [-Data] <Object> [[-Attachment] <String>
↳ ] [-BatchMode] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
↳ ProxyCredential] <Object>] [<CommonParameters>]
```

**Set-CouchDBDocument**

```
Set-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
↳ Document] <String> [-Revision] <String> [[-Data] <Object>] [[-Partition] <String>]
↳ [[-Replace] [[-Attachment] <String>] [-BatchMode] [-NoConflict] [[-Authorization]
↳ <Object>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <PSCredential>] [
↳ <CommonParameters>]
```

**Remove-CouchDBDocument**

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

**Copy-CouchDBDocument**

```
Copy-CouchDBDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
  ↳Document] <String> [-Destination] <String> [[-Revision] <String>] [[-Authorization]
  ↳ <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [
  ↳ <CommonParameters>]
```

**Get-CouchDBBulkDocument**

```
Get-CouchDBBulkDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
  ↳ [[-Data] <Object>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
  ↳ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

**New-CouchDBBulkDocument**

```
New-CouchDBBulkDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
  ↳ [[-Data] <Object>] [[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-
  ↳ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

**Get-CouchDBAttachment**

```
Get-CouchDBAttachment [-Server <String>] [-Port <Int32>] [-Database <String>] [-
  ↳Document <String>] [-Revision <String>] [-Attachment <String>] [-OutFile <String>]
  ↳ [-Variable <String>] [-Authorization <String>] [-Ssl] [[-ProxyServer] <String>] [[-
  ↳ProxyCredential] <Object>] [<CommonParameters>]
```

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

**Add-CouchDBAttachment**

```
Add-CouchDBAttachment [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
  ↳Document] <String> [-Attachment] <Object> [-Revision] <String> [[-Authorization]
  ↳ <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [
  ↳ <CommonParameters>]
```

**Remove-CouchDBAttachment**

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

**Clear-CouchDBDocuments**

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

### Search-CouchDBFullText

```
Search-CouchDBFullText [[-Server] <String>] [[-Port] <Int32>] [-Database] <String> [-
  ↳Patterns] <Array> [-UseQueries] [[-Authorization] <String>] [-Ssl] [[-ProxyServer]
  ↳<String>] [[-ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

### Find-CouchDBDocuments

```
Find-CouchDBDocuments [-Server <String>] [-Port <Int32>] [-Database <String>] [-
  ↳Partition <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] [[-
  ↳ProxyServer] <String>] [[-ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

```
Find-CouchDBDocuments [-Server <String>] [-Port <Int32>] [-Database <String>] [-
  ↳Partition <String>] [-Explain] [-Find <String>] [-Authorization <String>] [-Ssl] [[-
  ↳ProxyServer] <String>] [[-ProxyCredential] <Object>] [-AsJob] [<CommonParameters>]
```

## 14.1.9 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] [[-
  ↳ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

### Get-CouchDBDesignDocument

```
Get-CouchDBDesignDocument [[-Server] <String>] [[-Port] <Int32>] [-Database] <String>
  ↳[-Document] <String> [-Info] [[-Authorization] <String>] [-Ssl] [[-ProxyServer]
  ↳<String>] [[-ProxyCredential] <Object>] [[-Variable] <String>] [<CommonParameters>]
```

### Get-CouchDBDesignDocumentAttachment

```
Get-CouchDBDesignDocumentAttachment [-Server <String>] [-Port <Int32>] [-Database]
  ↳<String> [-Document <String>] [-Revision <String>] [-Attachment <String>] [-
  ↳OutFile <String>] [-Authorization <String>] [-Ssl] [[-ProxyServer] <String>] [[-
  ↳ProxyCredential] <Object>] [-Variable <String>] [<CommonParameters>]
```

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

### Add-CouchDBDesignDocumentAttachment

```
Add-CouchDBDesignDocumentAttachment [[-Server] <String>] [[-Port] <Int32>] [-
  ↳Database] <String> [-Document] <String> [-Attachment] <Object> [-Revision] <String>
  ↳[[[-Authorization] <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
  ↳<Object>] [<CommonParameters>]
```

**New-CouchDBDesignDocument**

```
New-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-ViewName <String>] [-ViewMapFunction <String>] [-
↪ViewReduceFunction <String>] [-ValidationFunction <String>] [-Authorization <String>] [-
↪] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
↪]
```

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

**Set-CouchDBDesignDocument**

```
Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Revision <String>] [-ViewName <String>] [-ViewMapFunction
↪<String>] [-ViewReduceFunction <String>] [-ValidationFunction <String>] [-
↪Authorization <String>] [-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential]
↪<Object>] [<CommonParameters>]
```

```
Set-CouchDBDesignDocument [-Server <String>] [-Port <Int32>] [-Database <String>] [-
↪Document <String>] [-Revision <String>] [-Data <Object>] [-Authorization <String>]
↪[-Ssl] [[-ProxyServer] <String>] [[-ProxyCredential] <Object>] [<CommonParameters>]
```

**Compress-CouchDBDesignDocument**

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

**Remove-CouchDBDesignDocument**

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

**Remove-CouchDBDesignDocumentAttachment**

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

## 14.2 Aliases

```
creq -> Send-CouchDBRequest
acnode -> Add-CouchDBNode
ccdb -> Compress-CouchDBDatabase
ccdd -> Compress-CouchDBDesignDocument
ccdoc -> Clear-CouchDBDocuments
ccview -> Clear-CouchDBView
```

(continues on next page)

(continued from previous page)

```
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
gcconf -> Get-CouchDBConfiguration
gccs -> Get-CouchDBClusterSetup
gcdb -> Get-CouchDBDatabase
gcdbc -> Get-CouchDBDatabaseChanges
gcdbp -> Grant-CouchDBDatabasePermission
gcdbs -> Get-CouchDBDatabaseSecurity
gcdbsh -> Get-CouchDBDatabaseShards
gcdbrs -> Get-CouchDBReshards
scdbrs -> Set-CouchDBReshards
rcdbrs -> Remove-CouchDBReshards
gcdbu -> Get-CouchDBDatabaseUpdates
gcddd -> Get-CouchDBDatabaseDesignDocument
gcdoc -> 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
acatt -> Add-CouchDBAttachment
ncdb -> New-CouchDBDatabase
ncddoc -> New-CouchDBDesignDocument
adatt -> Add-CouchDBDesignDocumentAttachment
ncdoc -> New-CouchDBDocument
ncidx -> New-CouchDBIndex
ncrpl -> New-CouchDBReplication
ncusr -> New-CouchDBUser
ncuuid -> New-CouchDBUuids
ncbd -> New-CouchDBBulkDocument
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
```

(continues on next page)

(continued from previous page)

```
rcnode -> Remove-CouchDBNode
rcrpl -> Remove-CouchDBReplication
rcs -> Remove-CouchDBSession
rcsrv -> Restart-CouchDBServer
rcusr -> Remove-CouchDBUser
scadm -> Set-CouchDBAdmin
scconf -> Set-CouchDBConfiguration
scdbpl -> Set-CouchDBDatabasePurgedLimit
scddoc -> Set-CouchDBDesignDocument
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
rmdadmin -> Remove-CouchDBAdmin
scda -> Search-CouchDBAnalyze
```

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

## 15.1 Create an PSCouchDB object

To create every object defined in PSCouchDB module, use this.

```
$req = New-CouchDBObject -TypeName PSCouchDBRequest
$req.GetType()
```

## 15.2 PSCouchDBRequest class

This class is used to construct a http request object.

### 15.2.1 Properties

attachment	Property	PSCouchDBAttachment attachment {get;set;}
authorization	Property	pscredential authorization {get;set;}
data	Property	string data {get;set;}
database	Property	string database {get;set;}
document	Property	string document {get;set;}
method	Property	string method {get;set;}
<b>parameter</b>	Property	string <b>parameter</b> {get;set;}
port	Property	int port {get;set;}
protocol	Property	string protocol {get;set;}
server	Property	string server {get;set;}
uri	Property	System.UriBuilder uri {get;set;}

## 15.2.2 Methods

AddAttachment	Method	void AddAttachment(string file)
AddAuthorization	Method	void AddAuthorization(pscredential credential), void ↵ ↵AddAuthorization(string auth)
ClearCache	Method	void ClearCache()
DisableCache	Method	void DisableCache()
EnableCache	Method	void EnableCache()
Equals	Method	bool Equals(System.Object obj)
GetData	Method	string GetData()
GetHashCode	Method	int GetHashCode()
GetHeader	Method	string GetHeader()
GetStatus	Method	int GetStatus()
GetType	Method	type GetType()
GetUri	Method	uri GetUri()
RemoveProxy	Method	void RemoveProxy()
Request	Method	pobject Request()
RequestAsJob	Method	void RequestAsJob(string name)
SetData	Method	void SetData(string json)
SetDatabase	Method	void SetDatabase(string database)
SetDocument	Method	void SetDocument(string document)
SetMethod	Method	void SetMethod(string method)
SetParameter	Method	void SetParameter(array parameter)
SetPort	Method	void SetPort(int port)
SetProxy	Method	void SetProxy(string uri), void SetProxy(string uri, ↵ ↵string user, string pass), void SetProxy(string uri, pscredential credential)
SetServer	Method	void SetServer(string server)
SetSsl	Method	void SetSsl(), void SetSsl(int port)
ToString	Method	string ToString()

## 15.2.3 Build a request

To create a PSCouchDBRequest object, just do the following.

```
using module PSCouchDB
$req = New-Object -TypeName PSCouchDBRequest           # GET http://
↵localhost:5984/
$req = New-Object -TypeName PSCouchDBRequest -ArgumentList 'db'           # GET http://
↵localhost:5984/db
$req = New-Object -TypeName PSCouchDBRequest -ArgumentList 'db','doc1' # GET http://
↵localhost:5984/db/doc1
$req.GetType()
```

## 15.2.4 Set server

To set a different server (default is localhost).

```
$req.SetServer('https://cdb1.local:443') # http URI
$req.SetServer('cdb1.local')           # FQDN
$req.SetServer('127.0.0.15')           # ip address
```

## 15.2.5 Set port

To set a different port (default is 5984).

```
$req.SetPort(8080)
```

## 15.2.6 Set proxy

To set a proxy for request.

```
$req.SetProxy('https://myproxy.mydomain.com:8080') # proxy_
↳without credential
$req.SetProxy('https://myproxy.mydomain.com:8080', 'user', 'password') # proxy with_
↳user and password
# Create credential object
$secStringPassword = ConvertTo-SecureString 'password' -AsPlainText -Force
$credObject = New-Object System.Management.Automation.PSCredential ('user',
↳$secStringPassword)
$req.SetProxy('https://myproxy.mydomain.com:8080', $credObject) # proxy with_
↳PSCredential object
$req.RemoveProxy() # remove_
↳proxy server and credential
```

## 15.2.7 Set SSL

To set a SSL https.

```
$req.SetSsl() # https on 6984
$req.SetSsl(443) # https on 443
```

## 15.2.8 Set a method

Default method is GET. To set other method, run this.

```
$req.SetMethod('PUT')
```

## 15.2.9 Add authorization

To add authorization with two methods.

```
$req.AddAuthorization('admin:password') # string version
[string]$userName = 'admin'
[string]$userPassword = 'password'
[securestring]$secStringPassword = ConvertTo-SecureString $userPassword -AsPlainText -
↳Force
[pscredential]$credObject = New-Object System.Management.Automation.PSCredential (
↳$userName, $secStringPassword)
$req.AddAuthorization($credObject) # PSCredential object version
```

### 15.2.10 Set database

To set a database on the URI object.

```
$req.SetDatabase('db')
```

### 15.2.11 Set document

To set a document on the URI object.

```
$req.SetDocument('doc')
```

### 15.2.12 Add attachment

To add an attachment to document.

```
$req.AddAttachment('/path/of/file.txt')
```

### 15.2.13 Set parameter

To set a parameter on the URI object.

```
$req.SetParameter('param')           # single param string
$req.SetParameter('param1=true')     # key=value param string
$req.SetParameter(@('param','param1=true')) # array param
```

### 15.2.14 Set data

To set a json data.

```
$data = @"
{
  "doc_ids": [
    "test"
  ]
}
"@
$req.SetData($Data)
$req.GetData() # verify json data
```

### 15.2.15 Get Uri

```
$req.GetUri()
```

### 15.2.16 Get request header

To get a request header.

```
$req.getHeader ()
```

### 15.2.17 Get status code

To get a status code of last request.

```
$req.getStatus ()
```

### 15.2.18 Get request

To get a result of request.

```
$req.Request ()
```

### 15.2.19 Get request in background

To send a request in background like daemon or job.

```
$req.RequestAsJob ('name of job')
```

## 15.3 PSCouchDBQuery class

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

### 15.3.1 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;}

### 15.3.2 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)
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)
RemoveIndexies	Method	void RemoveIndexies(), void RemoveIndexies(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()

### 15.3.3 Build a query

To create a PSCouchDBQuery object, just do the following.

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

### 15.3.4 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)
```

## 15.3.5 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
```

## 15.3.6 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')
```

## 15.3.7 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')
```

### 15.3.8 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')
```

### 15.3.9 ReadQuorum

To set *ReadQuorum*.

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

To remove it.

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

### 15.3.10 Bookmark

To configure bookmarks created previously.

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

To remove it.

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

### 15.3.11 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
```

### 15.3.12 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)
```

### 15.3.13 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')
```

(continues on next page)

(continued from previous page)

```
#or
$query.AddSelector('name', 'Arthur')
$query.AddSelector('planet', @('Heart', 'Magrathea'))
$query.AddSelectorOperator('$in', 'planet', 'Magrathea')
#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$')
```

## 15.3.14 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()
```

## 15.3.15 Native query format (Mango)

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

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

## 15.4 PSCouchDBDocument class

This class is used to construct a documents.

### 15.4.1 Properties

<code>_id</code>	Property	<code>string _id {get;set;}</code>
<code>_rev</code>	Property	<code>string _rev {get;set;}</code>
<code>_attachments</code>	Property	<code>hashtable _attachments {get;set;}</code>

### 15.4.2 Methods

<code>AddAttachment</code>	Method	<code>void AddAttachment(PSCouchDBAttachment attachment),</code> <code>↪void AddAttachment(string attachment)</code>
<code>Equals</code>	Method	<code>bool Equals(System.Object obj)</code>
<code>FromJson</code>	Method	<code>hashtable FromJson(string json)</code>
<code>GetDocument</code>	Method	<code>hashtable GetDocument()</code>
<code>GetHashCode</code>	Method	<code>int GetHashCode()</code>
<code>GetType</code>	Method	<code>type GetType()</code>
<code>RemoveAllAttachments</code>	Method	<code>void RemoveAllAttachments()</code>
<code>RemoveAttachment</code>	Method	<code>void RemoveAttachment(string attachment)</code>
<code>RemoveElement</code>	Method	<code>void RemoveElement(string key)</code>
<code>ReplaceAttachment</code>	Method	<code>void ReplaceAttachment(PSCouchDBAttachment</code> <code>↪attachment), void ReplaceAttachment(string attachment)</code>
<code>SetDocumentId</code>	Method	<code>void SetDocumentId(string id)</code>
<code>SetElement</code>	Method	<code>void SetElement(string key), void SetElement(string</code> <code>↪key, System.Object value)</code>
<code>ToJson</code>	Method	<code>string ToJson(), string ToJson(int depth), string</code> <code>↪ToJson(int depth, bool compress)</code>
<code>Tostring</code>	Method	<code>string ToString()</code>

### 15.4.3 Build a document

To create a PSCouchDBDocument object, just do the following.

```
using module PSCouchDB
$doc = New-Object -TypeName PSCouchDBDocument
$doc.GetType()
```

#### Add element to document

Add one element to our document object.

```
$doc.SetElement("test") # New key "test" with empty value
$doc.SetElement("test1", "value1") # New key "test1" with value "value1"
```

### Modify element to document

Modify or add an exists element on document object.

```
$doc.SetElement("test", "newvalue")
```

### Remove element to document

Delete an exists element on document object.

```
$doc.RemoveElement("test")
```

### View document

To view entire element of document object.

```
$doc.GetDocument()
```

### Get json document

To get json representation of document object.

```
$doc.ToJson()
```

### Add one attachment

Add an attachment to doc object.

```
$doc.AddAttachment('C:\test.txt') # string option  
$attachment = New-Object PSCouchDBAttachment -ArgumentList 'C:\test.txt'  
$doc.AddAttachment($attachment) # PSCouchDBAttachment option
```

### Replace one attachment

Replace an attachment to doc object.

```
$doc.ReplaceAttachment('C:\test.txt') # string option  
$attachment = New-Object PSCouchDBAttachment -ArgumentList 'C:\test.txt'  
$doc.ReplaceAttachment($attachment) # PSCouchDBAttachment option
```

### Remove one attachment

Remove an attachment to doc object.

```
$doc.RemoveAttachment('test.txt')
```

## Remove all attachments

Remove all attachments to doc object.

```
$doc.RemoveAllAttachments()
```

## 15.5 PSCouchDBAttachment class

This class is used to construct an attachment documents.

### 15.5.1 Properties

```
content_type Property  string content_type {get;set;}
filename      Property  string filename {get;set;}
```

### 15.5.2 Methods

```
Equals      Method  bool Equals(System.Object obj)
GetData     Method  string GetData()
SaveData    Method  void GetData()
GetHashCode Method  int GetHashCode()
GetRawData  Method  byte[] GetRawData()
GetType     Method  type GetType()
ToString    Method  string ToString()
```

### 15.5.3 Build an attachment

To create a PSCouchDBAttachment object, just do the following.

```
using module PSCouchDB
$attachment = New-Object PSCouchDBAttachment -ArgumentList "C:\test\test.log"
$attachment.GetType()
```

### 15.5.4 Get content of an attachment

Get content of an attachment of a documents

```
$attachment.GetData()
```

### 15.5.5 Attach a file to document

Create document object PSCouchDBDocument with attachment

```
$attach = New-Object PSCouchDBAttachment -ArgumentList "C:\test\test.log"
$doc1 = New-Object PSCouchDBDocument -ArgumentList '122', '1-
↳2c903913030efb4d711db085b1f44107', "C:\test\test.log"
$doc2 = New-Object PSCouchDBDocument -ArgumentList '122', '1-
↳2c903913030efb4d711db085b1f44107', $attach
$doc1.GetDocument()
$doc2.GetDocument()
```

## 15.6 PSCouchDBBulkDocument class

This class is used to construct an bulk documents.

### 15.6.1 Properties

docs	Property	PSCouchDBDocument[] docs {get;set;}
------	----------	-------------------------------------

### 15.6.2 Methods

AddDocument	Method	void AddDocument(string doc), void_
↳AddDocument		(PSCouchDBDocument doc)
Equals	Method	bool Equals(System.Object obj)
GetDocuments	Method	PSCouchDBDocument[] GetDocuments()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
RemoveDocument	Method	void RemoveDocument(string _id)
SetDeleted	Method	void SetDeleted()
ToString	Method	string ToString()

### 15.6.3 Create bulk document

Create a bulk document.

```
using module PSCouchDB
$bdocs = New-Object PSCouchDBBulkDocument
$bdocs.GetType()
```

You can create also a bulk document with one or more documents.

```
using module PSCouchDB
$doc120 = New-Object PSCouchDBDocument -ArgumentList '120'
$doc121 = New-Object PSCouchDBDocument -ArgumentList '121'
$doc122 = New-Object PSCouchDBDocument -ArgumentList '122'
# One document
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList $doc120
↳ # PSCouchDBDocument
$bdocs = New-Object PSCouchDBBulkDocument -ArgumentList '{"_id":"test","name":"test"}
↳' # JSON
# Two or more documents
$bdocs = [PSCouchDBBulkDocument]@{docs=@($doc120,$doc121,$doc122)}
$bdocs.GetType()
```

## 15.6.4 Add document

To add document to bulk documents.

```
$bdocs.AddDocument($doc120) # PSCouchDBDocument
$bdocs.AddDocument('{"_id":"test","name":"test"}') # JSON
```

## 15.6.5 Remove document

To remove document to bulk documents.

```
$bdocs.RemoveDocument('120') # _id of document
```

## 15.7 PSCouchDBSecurity class

This class is used to construct a security database documents.

### 15.7.1 Properties

admins	Property	pobject admins {get;set;}
members	Property	pobject members {get;set;}

### 15.7.2 Methods

AddAdmins	Method	void AddAdmins(string name), void AddAdmins(array name), ↵ ↵void AddAdmins(string name, str...
AddMembers	Method	void AddMembers(string name), void AddMembers(array name), ↵ void AddMembers(string name, ...
Equals	Method	bool Equals(System.Object obj)
GetAdmins	Method	hashtable GetAdmins()
GetHashCode	Method	int GetHashCode()
GetMembers	Method	hashtable GetMembers()
GetType	Method	type GetType()
RemoveAdminName	Method	void RemoveAdminName(string name)
RemoveAdminRole	Method	void RemoveAdminRole(string role)
RemoveMemberName	Method	void RemoveMemberName(string name)
RemoveMemberRole	Method	void RemoveMemberRole(string role)
ToJson	Method	string ToJson()
ToString	Method	string ToString()

### 15.7.3 Create security document

Create a security document for assign permission to a database.

```
using module PSCouchDB
$sec = New-Object PSCouchDBSecurity
$sec.GetType()
```

### 15.7.4 Get admins

To get all admin names and roles.

```
$sec.GetAdmins()
```

### 15.7.5 Get members

To get all member names and roles.

```
$sec.GetMembers()
```

### 15.7.6 Add admins

Add one or more admin names and roles to security object.

```
$sec.AddAdmins('root') # add admin name
$sec.AddAdmins('root', 'roots') # add admin name and role
$sec.AddAdmins(@( 'root', 'admin')) # add admin names
$sec.AddAdmins(@( 'root', 'admin'), @( 'roots', 'admins')) # add admin names and
↳roles
```

### 15.7.7 Add members

Add one or more member names and roles to security object.

```
$sec.AddMembers('member1') # add member name
$sec.AddMembers('member1', 'access') # add member name and
↳role
$sec.AddMembers(@( 'member1', 'member2')) # add member names
$sec.AddMembers(@( 'member1', 'member2'), @( 'access', 'read')) # add member names
↳and roles
```

### 15.7.8 Remove admin

Remove one admin to security object.

```
$sec.RemoveAdminName('root') # remove member name
$sec.RemoveAdminRole('roots') # remove member role
```

### 15.7.9 Remove member

Remove one member to security object.

```
$sec.RemoveMemberName('member1') # remove member name
$sec.RemoveMemberRole('access') # remove member role
```

### 15.7.10 Get json

To get json string to security object.

```
$sec.ToJson()
```

## 15.8 PSCouchDBReplication class

This class is used to construct a replica database documents.

### 15.8.1 Properties

continuous	Property	bool continuous {get;set;}
source	Property	System.UriBuilder source {get;set;}
target	Property	System.UriBuilder target {get;set;}
_id	Property	string _id {get;set;}
_rev	Property	string _rev {get;set;}

### 15.8.2 Methods

AddDocIds	Method	void AddDocIds(array ids)
AddSourceAuthentication	Method	void AddSourceAuthentication(string user, string ↵ ↵passwd)
AddTargetAuthentication	Method	void AddTargetAuthentication(string user, string ↵ ↵passwd)
CreateTarget	Method	void CreateTarget()
Equals	Method	bool Equals(System.Object obj)
GetDocument	Method	hashtable GetDocument()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
SetCancel	Method	void SetCancel()
SetCheckpointInterval	Method	void SetCheckpointInterval(int ms)
SetContinuous	Method	void SetContinuous()
SetFilter	Method	void SetFilter(string <b>filter</b> )
SetQueryParams	Method	void SetQueryParams(hashtable paramaters)
SetRevision	Method	void SetRevision(string revision)
SetSelector	Method	void SetSelector(string selector)
SetSelector	Method	void SetSelector(PSCouchDBQuery selector)
SetSinceSequence	Method	void SetSinceSequence(string sequence)
SetSourceProxy	Method	void SetSourceProxy(string proxyUri)
SetSourceServer	Method	void SetSourceServer(string server)
SetSourceSsl	Method	void SetSourceSsl(), void SetSourceSsl(int port)
SetTargetProxy	Method	void SetTargetProxy(string proxyUri)
SetTargetServer	Method	void SetTargetServer(string server)
SetTargetSsl	Method	void SetTargetSsl(), void SetTargetSsl(int port)
ToJson	Method	string ToJson()
ToString	Method	string ToString()
UseCheckpoints	Method	void UseCheckpoints()

### 15.8.3 Create replication document

Create a replication document.

```
using module PSCouchDB
$rep = New-Object PSCouchDBReplication -ArgumentList 'db','repdb'
$rep.GetType()
```

### 15.8.4 Set revision

Set revision to replication document.

```
$rep.SetRevision("1-f6d66c4d70da66cded6bea889468eb14")
```

### 15.8.5 Set authentication

Add authentication for source and target database.

```
$rep.AddSourceAuthentication("admin","password") # Source
$rep.AddTargetAuthentication("admin","password") # Target
```

### 15.8.6 Set SSL

Set SSL (https protocol) for source and target database.

```
$rep.SetSourceSsl() # Source
$rep.SetTargetSsl(443) # Target, specifying port
```

### 15.8.7 Set server

Set server (default localhost) for source and target database.

```
$rep.SetSourceServer('db1.local') # Source
$rep.SetTargetServer('db2.local') # Target, specifying port
```

### 15.8.8 Set cancel operation

Set cancel flag for request replica operation.

```
$rep.SetCancel()
```

### 15.8.9 Set continuous replica

Set continuous flag for replica operation.

```
$rep.SetContinuous()
```

### 15.8.10 Other flag for replica

```
$rep.SetCheckpointInterval(300) # checkpoint interval in milliseconds
$rep.CreateTarget()             # create target database
$rep.AddDocIds('test','test2') # replicate only ids specified
$rep.SetFilter()                # set filter function (ddoc/filter format)
$rep.SetSourceProxy()           # set source proxy server
$rep.SetTargetProxy()           # set target proxy server
$rep.SetQueryParams()           # set query for filter function
$rep.SetSelector()              # set selector (see PSCouchDBQuery)
$rep.SetSinceSequence()         # set since sequence
$rep.UseCheckpoints()           # use checkpoint for replication
```

### 15.8.11 Get replication document

Get the replication document.

```
$rep.GetDocument() # hashtable format
$rep.ToJson()      # json format
```

## 15.9 PSCouchDBView class

This class is used to construct a view.

### 15.9.1 Properties

map	Property	string map {get;set;}
name	Property	string name {get;set;}
reduce	Property	string reduce {get;set;}
view	Property	pobject view {get;set;}

### 15.9.2 Methods

AddMapFunction	Method	void AddMapFunction(string <b>function</b> )
AddReduceFunction	Method	void AddReduceFunction(string <b>function</b> )
BuilMapFunction	Method	string [PSCouchDBView]::BuilMapFunction(hashtable_ ↪condition)
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetJsonView	Method	string GetJsonView()
GetType	Method	<b>type</b> GetType()
GetView	Method	hashtable GetView()
RemoveMapFunction	Method	void RemoveMapFunction()
RemoveReduceFunction	Method	void RemoveReduceFunction()
ReplaceMapFunction	Method	void ReplaceMapFunction(string <b>function</b> )
ReplaceReduceFunction	Method	void ReplaceReduceFunction(string <b>function</b> )
ToString	Method	string ToString()

### 15.9.3 Build a view

To create a PSCouchDBView object, just do the following.

```
using module PSCouchDB
$view = New-Object PSCouchDBView -ArgumentList "test_view"
$doc.GetType()
```

### 15.9.4 Get view

Get content of view, in two methods.

```
$view.GetView()      # hashtable content
$view.GetJsonView() # string json content
```

### 15.9.5 Add function

Add one map function to view object.

```
$view.AddMapFunction("function(doc) { emit(doc.name, doc.age); }") # add_
↪first map function
$view.ReplaceMapFunction("function(doc) { emit(doc.name, doc.surname); }") # replace_
↪exists map function
```

Add one reduce function to view object. The sets valid for the reduce functions are: `_approx_count_distinct`, `_count`, `_stats`, `_sum`

```
$view.AddReduceFunction("_sum") # add first reduce function
$view.ReplaceReduceFunction("_count") # replace exists reduce function
```

### 15.9.6 Remove function

Remove exists map function to view object.

```
$view.RemoveMapFunction()
```

Remove exists reduce function to view object.

```
$view.RemoveReduceFunction()
```

### 15.9.7 Build a map function

This object have a method than permit to create a simple map function. Before, create a condition hashtable.

```
$condition = @{
    EQUAL = 'doc.field1 == 0'; # Add if condition to function: if (doc.field1 ==
↪0) {}
    EQUEMIT = 'doc.field1'; # Add emit function to if equal condition: if
↪(doc.field1 == 0) {emit(doc.field1)}
    MINIMUM = 'doc.field1 < 0'; # Add if condition to function: if (doc.field1 <
↪0) {}
```

(continues on next page)

(continued from previous page)

```

    MINEMIT = 'doc.field2';      # Add emit function to if equal condition: if_
↪ (doc.field1 < 0) {emit(doc.field1)}
    MAXIMUM = 'doc.field1 > 0'; # Add if condition to function: if (doc.field1 >_
↪ 0) {}
    MAXEMIT = 'doc.field3';      # Add emit function to if equal condition: if_
↪ (doc.field1 > 0) {emit(doc.field1)}
    EMITDOC = "doc"              # If other emit is specified, this is null
}

```

Now pass this hashtable like argument to method.

```
$map = [PSCouchDBView]::BuildMapFunction($condition)
```

## 15.10 PSCouchDBDesignDoc class

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

### 15.10.1 Properties

validate_doc_update	Property	string validate_doc_update {get;set;}
views	Property	System.Object views {get;set;}
_attachments	Property	hashtable _attachments {get;set;}
_id	Property	string _id {get;set;}
_rev	Property	string _rev {get;set;}

### 15.10.2 Methods

AddAttachment	Method	void AddAttachment(PSCouchDBAttachment attachment), ↪ ↪void AddAttachment(string attach...
AddView	Method	void AddView(PSCouchDBView view), void AddView(string ↪ ↪name, string map), void AddVie...
Equals	Method	bool Equals(System.Object obj)
FromJson	Method	hashtable FromJson(string json)
GetDocument	Method	hashtable GetDocument()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
RemoveAllAttachments	Method	void RemoveAllAttachments()
RemoveAttachment	Method	void RemoveAttachment(string attachment)
RemoveElement	Method	void RemoveElement(string key)
RemoveView	Method	void RemoveView(string name)
ReplaceAttachment	Method	void ReplaceAttachment(PSCouchDBAttachment ↪ ↪attachment), void ReplaceAttachment(strin...
ReplaceView	Method	void ReplaceView(PSCouchDBView view), void ↪ ↪ReplaceView(string name, string map), voi...
SetElement	Method	void SetElement(string key), void SetElement(string ↪ ↪key, string value)
SetValidateFunction	Method	void SetValidateFunction(string <b>function</b> )
ToJson	Method	string ToJson(), string ToJson(int depth), string ↪ ↪ToJson(int depth, bool compress)
ToString	Method	string ToString()

### 15.10.3 Build a design document

To create a `PSCouchDBDesignDoc` object, just do the following.

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

### 15.10.4 Work with views

Views are the primary tool used for querying and reporting on CouchDB documents. With `AddView` it is possible to add map and reduce function.

#### Add map function

```
$ddoc.AddView('myview', 'function(doc){emit(doc);}')
```

#### Add reduce function

```
$ddoc.AddView('myview', 'function(doc){emit(doc);}', '_count')
```

#### Add view object

```
$view = New-Object PSCouchDBView -ArgumentList "myview"
$view.AddMapFunction("function(doc) { emit(doc.name, doc.age); }")
$view.AddReduceFunction("_sum")
$ddoc.AddView($view)
```

### 15.10.5 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.SetValidateFunction('function(newDoc, oldDoc, userCtx, secObj) {if (newDoc.type_
↳ == "post") { // validation logic for posts}}')
```

### 15.10.6 Native design document

To receive the design document in native format use the `ToJson` method.

```
$ddoc.ToJson()
```

### 15.10.7 Create design document

See [Create design document](#).

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



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

## 16.1 Scripts

Scripts, automations and tools are the main use of this module.

### 16.1.1 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 -Authorization "admin:password"
```

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,
```

(continues on next page)

(continued from previous page)

```

        NumberOfProcessors,
        @{Expression={$_.TotalPhysicalMemory / 1GB};Label=
↪"TotalPhysicalMemoryGB"}})
        $info.Add('ComputerCPU', (Get-CimInstance win32_processor -ComputerName
↪$ComputerName |
        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 -Authorization "admin:password") {
            Set-CouchDBDocument -Database hw_inventory -Document $ComputerName -Data
↪$info -Revision $(Get-CouchDBDocument -Database hw_inventory -Document
↪$ComputerName -Authorization "admin:password")._rev -Replace -Authorization
↪"admin:password"
        } else {
            New-CouchDBDocument -Database hw_inventory -Document $ComputerName -Data
↪$info -Authorization "admin:password"
        }
    }
}
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 -Authorization "admin:password"
    $docs.docs
}

```

Schedule the script at any hour or minute you want.

## 16.1.2 Log storage

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

Edit your *profile.ps1* and put this function:

```

using module PSCouchDB

# Write log

```

(continues on next page)

(continued from previous page)

```

function Write-WindowsLog () {
    param(
        [Parameter(mandatory=$true)]
        [string] $ComputerName,
        $Authorization
    )
    # Define logs
    $Logs = @("Application","Security","System")
    # Loop foreach log
    foreach ($Log in $logs) {
        $count = 0
        $DBName = "${ComputerName}_$Log".ToLower()
        # Test if database log exists
        if ($null -eq (Test-CouchDBDatabase -Database $DBName -ErrorAction_
↪SilentlyContinue -Authorization $Authorization)) {
            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 -Authorization $Authorization)) {
                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)
        }
    }
}

# 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.*")

```

(continues on next page)

(continued from previous page)

```

    $q.AddLogicalOperator('$or')
    $q.AddFields("_id")
    $q.AddFields("MachineName")
    $q.AddFields("Data")
    $q.AddFields("Index")
    $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.ToLower())" -Find
    ↪$q.GetNativeQuery() -Authorization "admin:password"
    $docs.docs
}

```

## 16.2 OOP

This module has classes representing the various types of documents that exist in CouchDB. You can use them to create objects, extend them to create new custom classes and much more. Everything you know about OOP, you can take advantage of it and use it.

### 16.2.1 Custom document

Creating a custom document to reflect a specific template is very easy.

```

using module PSCouchDB

class PersonDocument : PSCouchDBDocument {
    [string] $Name
    [string] $Surname
    [int] $Age
    [string] $Title

    PersonDocument ([string]$Name, [string]$Surname) {
        $this.Name = $Name
        $this.Surname = $Surname
        $this.SetElement('Name', $Name)
        $this.SetElement('Surname', $Surname)
    }
}

# Create instance
$person = New-Object PersonDocument -ArgumentList 'Matteo','Guadrini'

# Add my age
$person.SetElement('Age', 34)
$person.Age = 34

# View CouchDB document
$person.ToJson()

```

(continues on next page)

(continued from previous page)

```
# Write on database
New-CouchDBDocument -Database 'persons' -Document $person.__id -Data $person -
↔Authorization "admin:password"
```

---

**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.

### 17.1 Contribute

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

### 17.2 Issue

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

### 17.3 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](#)