

22.10.0 Komponente Memcached (gb.memcached)

Viele werden sich fragen: Was ist Memcached? Auf der Webseite <https://memcached.org> finden Sie eine gute Antwort:

- ☑ Memcached ist ein "...kostenloses und quelloffenes, hochleistungsfähiges Objekt-Caching-System mit verteiltem Speicher.
- ☑ Es ist generisch, aber für den Einsatz bei der Beschleunigung dynamischer Webanwendungen gedacht, indem es die Datenbanklast mindert.
- ☑ Memcached ist ein In-Memory-Schlüsselwertspeicher für kleine Teile beliebiger Daten (Strings, Objekte) aus Ergebnissen von Datenbankaufrufen, API-Aufrufen oder dem Rendering von Seiten. Memcached ist einfach und dennoch leistungsfähig".

Die Komponente gb.memcached mit ihrer einzigen Klasse Memcached zielt darauf ab, Ihnen eine voll nutzbare Speicher-Schnittstelle zur Verfügung zu stellen.

22.10.0.1 Eigenschaften der Klasse Memcached

Die Klasse *Memcached* verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Debug	Boolean	Gibt an, ob der Debugging-Modus eingeschaltet ist oder schaltet den Debugging-Modus mit dem Wert True ein.
Error	String	Gibt den Fehler-String zurück, der vom <i>letzten</i> Befehl zurückgegeben wird, der fehlgeschlagen ist.
Host	String	Gibt den Host vom Memcached-Server (MC-Server) zurück oder setzt ihn. Der Standard-Host ist "localhost" oder "127.0.0.1".
Port	Integer	Gibt den Port zurück, der für die Verbindung vom MC-Client zum MC-Server verwendet wird oder setzt ihn.
SlabsAutomove	Integer	Aktiviert oder deaktiviert die Funktion "SlabsAutomove", die im aktivierten Zustand (1) die Defragmentierung im Cache-Speicher verhindert. Der Standard-Wert ist 1.
Status	Integer	Gibt den Status der Verbindung vom MC-Client zum MC-Server zurück.
Version	String	Gibt die aktuelle Version des Memcached-Servers zurück.

Tabelle 22.10.0.1.1 : Eigenschaften der Klasse Memcached

So können Sie ein neues Memcached-Client-Objekt erzeugen, dann ausgewählte Eigenschaften festlegen und abschließend den konfigurierten den Client starten:

```
Dim hMCClient As Memcached

hMCClient = New Memcached ' Es wird ein neuer MC-Client(!) erzeugt
hMCClient.Host = "localhost" ' Default-Host: localhost; Alternative: "127.0.0.1"
hMCClient.Port = 11211 ' Default-Port-Number
hMCClient.Debug = True ' True → Only for tests in the IDE

If Not hMCClient.Open() Then
    Message.Error(("Memcached.Open-Error") & gb.NewLine & Error.Text & gb.NewLine & Error.Where)
    Return
Endif
```

Hinweise:

- Wenn die Memcached-Eigenschaft *Debug* mit True gesetzt ist, dann gibt ein Memcached-Client alle Befehle, die an den MC-Server gesendet werden, in einer Konsole aus.
- Der Memcached-Status kann alternativ die folgenden Werte annehmen:
Net.Inactive (0) → Die Verbindung vom MC-Client zum MC-Server ist geschlossen.
Net.Connected (7) → Der MC-Client ist mit dem MC-Server verbunden.
- Die Standard-Portnummer für den Memcached-Server ist 11211.

22.10.0.2 Methoden der Klasse Memcached

Die Klasse *Memcached* verfügt über diese fünf Methoden:

Methoden	Rückgabotyp	Beschreibung
Open()	Boolean	Startet eine Verbindung vom MC-Client zum MC-Server und gibt <i>True</i> zurück, wenn das Öffnen erfolgreich war.
Close()	Boolean	Schließt eine Verbindung vom MC-Client zum MC-Server und gibt <i>True</i> zurück, wenn das Schließen erfolgreich war.
Retrieve (sKey As Variant [, iCas As Integer])	Collection	Die Methode wird verwendet, um einen Wert vom MC-Server abzurufen. sKey kann ein einzelner Schlüssel oder ein Array von Schlüsseln sein. Setzen Sie den optionalen Parameter 'iCas', wenn Sie wissen möchten, ob das Objekt geändert wurde, seit es zuletzt vom <i>rufenden</i> Client geholt wurde. Die Funktion gibt als Funktionswert eine Auflistung (Datentyp: Collection) von Schlüssel-Wert-Paaren zurück.
Flush ([iDelay As Integer])	-	Mit dem Befehl werden <i>alle</i> Schlüssel-Wert-Paare auf dem MC-Server gelöscht. Es wird ein optionaler Parameter <i>iDelay</i> akzeptiert, der die Zeit festlegt, nach der die Schlüssel-Wert-Paare tatsächlich gelöscht werden.
Exec (sCommand As String [, vData As Variant, bRaw As Boolean])	Variant	Die Funktion bietet <u>direkten Zugriff</u> auf den MC-Server. Der verwendete Befehl wird mit 'sCommand' angegeben. Der erste optionale Parameter 'vData' repräsentiert die zu speichernden Daten. Der zweite optionale Parameter gibt an, ob die Daten im Original vorliegen.

Tabelle 22.10.0.2.1 : Methoden der Klasse Memcached

22.10.0.3 Konstanten der Klasse Memcached

Die folgenden Konstanten beziehen sich u.a. auf die Ausgaben der Methode Memcached.Exec(...)!

Konstante	Wert	Beschreibung
Stored	0	Gibt an, dass ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel erfolgreich gespeichert wurde.
NotStored	1	Gibt an, dass ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel nicht gespeichert werden konnte.
Exists	2	Gibt an, dass ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel, das Sie mit einem Cas-Befehl speichern möchten, seit dem letzten Abruf geändert wurde.
NotFound	3	Gibt an, dass ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel nicht gefunden wurde.
Deleted	4	Gibt an, dass ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel erfolgreich gelöscht wurde.
Touched	5	Gibt an, dass die Ablaufzeit für ein Schlüssel-Wert-Paar mit dem angegebenen Schlüssel aktualisiert wurde.

Tabelle 22.10.0.3.1 : Konstanten der Klasse Memcached

22.10.0.4 Virtuelle Klasse _Memcached_Key (gb.memcached)

Die Klasse Memcached können Sie wie ein Nur-Lese-Array Memcached[] verwenden:

```
Dim hMemcached As Memcached
Dim hMemcachedKey As _Memcached_Key
hMemcachedKey = hMemcached [ sKey As String ]
```

Es wird ein virtuelles Memcached-Key-Objekt zurückgegeben, das mit dem Schlüssel 'sKey' (Datentyp String) verknüpft ist. Dies ist die *bevorzugte* Methode für den lesenden und schreibenden Datenzugriff. Die Klasse *_Memcached_Key* verfügt über diese beiden Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Value	Variant	Gibt den Wert zu dem angegebenen Schlüssel zurück oder setzt den Wert.

Eigenschaft	Datentyp	Beschreibung
CasID	Integer	Liefert einen ID zu einem bestimmten Schlüssel.

Tabelle 22.10.0.4.1 : Eigenschaften der Klasse `_Memcached_Key`

Wenn Sie ein neues Schlüssel-Wert-Paar speichern, das noch *nicht* auf dem MC-Server existiert, dann liefert zum Beispiel `hMCClient[sCurKey].CasID` den Wert 0.

Die Klasse `_Memcached_Key` besitzt diese fünf Methoden:

Methode	Rückgabotyp	Beschreibung
<code>Cas (sData As String)</code>	Boolean	Gibt True zurück, wenn die Änderung des Schlüssel-Wert-Paares für den verwendeten Schlüssel erfolgreich war.
<code>Delete</code>	Boolean	Gibt True zurück, wenn das Löschen des Schlüssel-Wert-Paares für den verwendeten Schlüssel erfolgreich war.
<code>Decrement ([iStep As Integer])</code>	Integer	Gibt den um den Wert des Parameters <code>iStep</code> verminderten Wert des Zählers für den verwendeten Schlüssel zurück.
<code>Increment ([iStep As Integer])</code>	Integer	Gibt den um den Wert des Parameters <code>iStep</code> erhöhten Wert des Zählers für den verwendeten Schlüssel zurück.
<code>Touch (iExpire As Integer)</code>	Boolean	Gibt True zurück, wenn das Setzen der Ablaufzeit für den Wert mit dem verwendeten Schlüssel erfolgreich war.

Tabelle 22.10.0.4.2 : Methoden der Klasse `_Memcached_Key`

Hinweise:

- Der kleinste Funktionswert der Methode `Decrement ([iStep As Integer])` ist 0. Alle weiteren Aufrufe der Funktion liefern dann ebenfalls 0.
- Der Parameter `iExpire` der Methode `Touch (iExpire As Integer)` legt die Ablaufzeit der Daten in Sekunden fest. Der Standardwert ist 0 und bedeutet: Der Wert läuft nie ab. Unabhängig davon kann der Wert aber zu jeder Zeit aus dem Cache gelöscht werden kann, um Platz für andere Schlüssel-Wert-Paare zu machen!

Bevor Sie die Komponente `gb.memcached` praktisch einsetzen, sollten Sie das wissen:

- Es gibt keine Zugriffskontrolle: Jeder MC-Client, der mit einem MC-Server verbunden ist, kann alle gespeicherten Daten auf dem MC-Server lesen und auch Daten schreiben.
- Alle Daten werden nur im Speicher abgespeichert. Mit dem Beenden eines Programms, das die Komponente `gb.memcached` nutzt, stehen die Daten nicht mehr zur Verfügung.
- Auf die Daten können Sie nur mit einem eindeutigen Schlüssel zugreifen. Iterieren ist nicht möglich!
- Daten können zur Laufzeit des Memcached-Clients entweder dauerhaft oder zeitlich begrenzt abgespeichert werden. Für den letzten Fall übernimmt der Server `Memcached` das Löschen automatisch.
- Der Daten-Typ für einen Schlüssel ist String.
- Der Daten-Typ für einen Wert ist entweder Variant oder Objekt.