

CouchDB-Cheat-Sheet

Andreas Wenk – Version: 0.2.0 – August 2011

Zusammenfassung

Dieses Cheat-Sheet ist eine Übersicht der RESTful API von CouchDB [1]. Das Ziel ist es, die gebräuchlichsten Requests jeweils anhand eines Beispiels zu zeigen. Unter [2] ist eine CouchDB zu finden, in der die Beispiele nachgestellt werden können (Basic-Auth: ohne Daten ‘ok’ klicken).

Unter [3] ist die API-Referenz zu finden.

Damit die Listings so kurz als möglich sind, heisst die Datenbank, die als Beispiel dienen soll ‘kina’. Die Grundstruktur der Dokumente in dieser Datenbank ist folgendermaßen:

```
{
  "_id": "fab911e6e9ed703bf536dccffe000937",
  "_rev": "1-183f162a09532ccacf8223f6e2ba95f6",
  "lang": "javascript"
}
```

Alle Beispiel-Requests werden unter Verwendung von cURL [10] erstellt. Natürlich kann auch das von CouchDB mitgelieferte Webinterface ‘Futon’ genutzt werden. Allerdings rate ich davon ab, da bei der Verwendung von cURL die Betrachtung der HTTP-Response zu wesentlich mehr Verständnis führt.

1 Überblick

CouchDB ist eine dokumentbasierte Datenbank und gehört zur Gruppe der NoSQL Datenbanken. CouchDB ist ein Apache Open-Source Projekt und liegt aktuell in der Version 1.1.0 vor. Einige wichtige Merkmale von CouchDB sind:

- in Erlang [4] geschrieben
- schemalos
- API: HTTP Rest [5]
- implementiert ACID [6] über MVCC [7]
- CAP Theorem [8]:
 - A (Availability) - ja
 - P (Partition Tolerance) - ja
 - C (Consistency) -> Eventual Consistency
- MapReduce [9] für Views
- Asynchrone Replikation in beide Richtungen - Master / Master
- Revisions Management

2 Datenbanken

2.1 Alle Datenbanken im Cluster [GET] [_all_dbs]

```
curl -X GET http://127.0.0.1:5984/_all_dbs
["_replicator","_users"]
```

2.2 Datenbank erstellen [PUT]

```
curl -X PUT http://127.0.0.1:5984/kina
{"ok": true}
```

2.3 Datenbankinformationen erhalten [GET]

```
curl -X GET http://127.0.0.1:5984/kina
{"db_name": "kina",
 "doc_count": 0,
 "doc_del_count": 0,
 "update_seq": 0,
 "purge_seq": 0,
 "compact_running": false,
 "disk_size": 79,
 "instance_start_time": "1312577294992829",
 "disk_format_version": 5,
 "committed_update_seq": 0}
```

2.4 Datenbank löschen [DELETE]

```
curl -X DELETE http://127.0.0.1:5984/kina
{"ok": true}
```

2.5 Datenbank replizieren [POST] [_replicate] (alte API)

(params) source=[url], target=[url], cancel=[true], create_target=[true], continuous=[true], doc_ids=[doc_ids_array], filter=[filter_name], proxy=[url], query_params=[params]

2.5.1 lokal nach lokal

```
curl -X POST http://127.0.0.1:5984/_replicate \
-H "content-type:application/json" \
-d '{"source": "kina",
    "target": "anik",
    "create_target": true}'

{"ok":true, "session_id":"f75eb944bac70f40e77953f484afb64c",
 "source_last_seq":36, "history":
  [{"session_id":"f75eb944bac70f40e77953f484afb64c",
   "start_time":"Thu, 14 Apr 2011 20:36:12 GMT",
   "end_time":"Thu, 14 Apr 2011 20:36:12 GMT",
   "start_last_seq":0,
   "end_last_seq":36,
   "recorded_seq":36,
   "missing_checked":0,
   "missing_found":14,
   "docs_read":14,
   "docs_written":14,
   "doc_write_failures":0
  ]}
}
```

2.5.2 remote nach lokal (umgekehrt genauso)

(!) Bei der Nutzung von `--create_target` muss die Authentifizierung mit einem User der Zieldatenbank erfolgen.

```
curl -X POST http://127.0.0.1:5984/_replicate \
-H "content-type:application/json" \
-d '{"source": "kina",
    "target": "http://starsky:hutch@couchbuch.iriscouch.com/anik",
    "create_target": true}'

{"ok":true,"session_id":"ed5fa48d0a5ce9cee6941381754d796d",
 "source_last_seq":29,"replication_id_version":2,
 "history":
  [{"session_id":"ed5fa48d0a5ce9cee6941381754d796d",
   "start_time":"Wed, 31 Aug 2011 19:34:40 GMT",
   "end_time":"Wed, 31 Aug 2011 19:34:41 GMT",
   "start_last_seq":0,
   "end_last_seq":29,"recorded_seq":29,
   "missing_checked":0,
   "missing_found":14,
   "docs_read":14,
   "docs_written":14,
   "doc_write_failures":0
  ]}
}
```

2.5.3 remote nach remote

```
curl -X POST http://127.0.0.1:5984/_replicate \
-H "content-type:application/json" \
-d '{"source": "http://andywenk.cloudant.com/kina",
    "target": "http://starsky:hutch@couchbuch.iriscouch.com/kina",
    "create_target": true}'

{"ok":true,"no_changes":true}
```

2.6 Datenbank replizieren - filter

Filter werden in `_design` Dokumenten erstellt. Ein Filter kann später bei der Replikation genutzt werden, um nur Dokumente zu replizieren, die den im Filter hinterlegten Kriterien entsprechen.

Filter erstellen:

```
curl -X PUT http://127.0.0.1:5984/kina/_design/default \
-H "content-type: application/json" \
-d '{"filters":{
  "lang":
    "function(doc, req) {
      return \"javascript\" == doc.lang
    }"
  }
}'
```

Filter anwenden (davon ausgehend, es gibt drei Dokumente mit `lang = javascript`):

```
curl -X POST http://127.0.0.1:5984/_replicate \
-H "content-type:application/json" \
-d '{"source": "kina",
    "target": "anik",
    "create_target": true,
    "filter": "default/lang"}'

{"ok":true,"session_id":"3546d87233636b04440e6e023c3e3018",
 "source_last_seq":6,
 "replication_id_version":2, "history":
  [{"session_id":"3546d87233636b04440e6e023c3e3018",
   [...]
   "docs_read":3,
   "docs_written":3,
   "doc_write_failures":0
  ]}
}
```

Filter erstellen mit dynamischem Parameter (Auszug):

```
curl [...]
-d '{"filters":{
```

```

    "lang":
      "function(doc, req) {
        return req.query.lang == doc.lang
      }"
  }
},

```

Filter anwenden (Auszug):

```

curl [...]
-d '{"source": "kina",
  "target": "anik",
  "create_target": true,
  "query_params": {
    "lang": "javascript"
  }
},

```

Filter andwenden unter Angabe von genauen Dokument IDs (Auszug):

```

curl [...]
-d '{"source": "kina",
  "target": "anik",
  "create_target": true,
  "doc_ids": ["id_1", "id_2", "id_n"]
},

```

2.7 Datenbank replizieren mit der Datenbank [_replicate]

(*params*) source=[url], target=[url], cancel=[true], create_target=[true], continuous=[true], doc_ids=[doc_ids_array], filter=[filter_name], proxy=[url], query_params=[params]

Dokumentation unter [11] zu finden

2.8 Änderungen in der DB verfolgen [GET] [_changes]

(*params*) since=[sequence_number], style=all_docs, limit=[n], feed=[continuous,longpolling], heartbeat=[milliseconds], filter=filter_name, include_docs=[true,false], timeout=[milliseconds]

```

curl -X GET "http://127.0.0.1:5984/kina/_changes \
?feed=continuous&heartbeat=2000"
{"seq":3,"id":"kn0001","changes":
  [{"rev":"3-5570e8bbb34412db757c2df4bfa1099b"}]}
{"seq":4,"id":"fab911e6e9ed703bf536dccffe000937","changes":
  [{"rev":"1-183f162a09532ccacf8223f6e2ba95f6"}]}
[...]

```

2.9 Aktuelle tasks abfragen [GET] [_active_tasks]

```

curl -X GET http://127.0.0.1:5984/_active_tasks
[{"pid":"<0.7356.3>","changes_done":159756,
  "database":"kina","progress":6,
  "started_on":1359732010,"total_changes":2292848,
  "type":"database_compaction","updated_on":1359732068
}]

```

2.10 Compaction: Datenbank [POST] [_compact]

```

curl -X POST http://127.0.0.1:5984/kina/_compact \
-H "content-type:application/json"
{"ok":true}

```

2.11 Compaction: Design Dokumente [POST] [_compact/design-doc]

```

curl -X POST http://127.0.0.1:5984/kina/_compact/default \
-H "content-type:application/json"
{"ok":true}

```

2.12 Compaction: Views [POST] [_view_cleanup]

```

curl -X POST http://127.0.0.1:5984/kina/_view_cleanup \
-H "content-type:application/json"
{"ok":true}

```

3 Dokumente

Prinzipiell sollte beim Erstellen auf IDs zurückgegriffen werden, die CouchDB generiert. Allerdings macht es in manchen Fällen auch Sinn eigene IDs zu nutzen.

3.1 UUIDs ausgeben [GET] [_uuids]

(*params*) count=[n]

```

curl -X GET http://127.0.0.1:5984/_uuids?count=3
{"uuids":
  ["b7fc0ca7dffcc6d0f240c1e5bb000998",
   "b7fc0ca7dffcc6d0f240c1e5bb000fe8",
   "b7fc0ca7dffcc6d0f240c1e5bb001c9a"]
}

```

3.2 Alle Dokumente erhalten [GET] [_all_docs]

(*params*) descending=true, key=[key], st-artkey=[key], startkey_docid=[docid], endkey=[key], endkey_docid=[docid], group=[true,false], group_level=[0-n], inclusive_end=[true,false], limit=[n], reduce=[true,false], skip=[n], stale=ok, update_seq=[true,false]

```
curl -X GET http://127.0.0.1:5984/kina/_all_docs

{"total_rows":12,"offset":0,"rows":[
  {"id":"7341477ce373f9cc76f351e5980008bb",
    "key":"7341477ce373f9cc76f351e5980008bb",
    "value":{"rev":"2-f0bfca3976ad04bce05b2ade242519d7"}},
  {"id":"7341477ce373f9cc76f351e5980015cd",
    "key":"7341477ce373f9cc76f351e5980015cd",
    "value":{"rev":"2-d6f5f2cb326c1f68f95d2bfbef329280"}}]}
```

3.3 Mehrere Dokumente mit key(id) erhalten [POST] [_all_docs]

(*params*) siehe: Alle Dokumente erhalten

(!) gut nutzbar, um mit einem Request mehrere Dokumente zu erhalten

```
curl -X POST http://127.0.0.1:5984/kina/_all_docs?include_docs=true \
-H "content-type:application/json" \
-d '{"keys": [
  "fab911e6e9ed703bf536dccffe000937",
  "7341477ce373f9cc76f351e5980015cd"
]}'

{"total_rows":12,"offset":0,"rows":[
  {"id":"fab911e6e9ed703bf536dccffe000937",
    "key":"fab911e6e9ed703bf536dccffe000937",
    "value":{"rev":"1-183f162a09532ccacf8223f6e2ba95f6"},
    "doc":{"_id":"fab911e6e9ed703bf536dccffe000937",
      "_rev":"1-183f162a09532ccacf8223f6e2ba95f6",
      "lang":"javascript"}}},
  {"id":"7341477ce373f9cc76f351e5980015cd",
    "key":"7341477ce373f9cc76f351e5980015cd",
    "value":{"rev":"3-99b698d0caba3c9892a385dd0efe2a3d"},
    "doc":{"_id":"7341477ce373f9cc76f351e5980015cd",
      "_rev":"3-99b698d0caba3c9892a385dd0efe2a3d",
      "lang":"lisp"}}}
]}
```

3.4 Dokument erstellen mit CouchDB ID [PUT]

```
curl -X PUT http://127.0.0.1:5984/kina/ \
b7fc0ca7dffcc6d0f240c1e5bb000998

{"ok":true,"id":"b7fc0ca7dffcc6d0f240c1e5bb000998",
"rev":"1-967a00dff5e02add41819138abb3284d"}
```

3.5 Dokument erstellen mit eigener ID [PUT]

```
curl -X PUT http://127.0.0.1:5984/kina/kn0001 -d '{}'
```

```
{"ok":true,"id":"kn0001",
"rev":"3-5570e8bbb34412db757c2df4bfa1099b"}
```

3.6 Dokument erstellen [POST]

```
curl -X POST http://127.0.0.1:5984/kina/ \
-H "content-type: application/json" \
-d '{}'
```

```
{"ok":true,"id":"b7fc0ca7dffcc6d0f240c1e5bb000fe8",
"rev":"1-367b00dfc5e02axd41819138abb3284d"}
```

3.7 Dokument anfragen [GET]

(*params*) rev=[revision], revs=true, revs_info=true

```
curl -X GET http://127.0.0.1:5984/kina/ \
b7fc0ca7dffcc6d0f240c1e5bb000998

{"_id":"b7fc0ca7dffcc6d0f240c1e5bb000998",
 "_rev":"1-367b00dfc5e02axd41819138abb3284d",
 "inhalt":"hier steht was"}
```

3.8 Dokument erweitern / aktualisieren [PUT]

(*params*) rev=revision

(*header*) if-match=revision

```
curl -X PUT http://127.0.0.1:5984/kina/ \
fab911e6e9ed703bf536dccffe000937 \
-d '{"_rev":"1-183f162a09532ccacf8223f6e2ba95f6",
"lang": "php"}'
```

```
curl -X PUT http://127.0.0.1:5984/kina/ \
fab911e6e9ed703bf536dccffe000937 \
-H "if-match:1-183f162a09532ccacf8223f6e2ba95f6" \
-d '{"lang":"php"}'
```

```
{"ok":true,"id":"fab911e6e9ed703bf536dccffe000937",
"rev":"2-cf8bad0865d8296870352617ab3afdba"}
```

3.9 Dokument löschen [DELETE]

(*params*) rev=revision
(*header*) if-match=revision
(!) Dokument wird nur aus dem Index gelöscht und markiert
mit `--deleted:true`

```
curl -X DELETE http://127.0.0.1:5984/kina/ \
7341477ce373f9cc76f351e598001cdd \
?rev=1-9533bf3b3c5cda717ed000b186fdafe
```

```
curl -X DELETE http://127.0.0.1:5984/kina/ \
7341477ce373f9cc76f351e598001cdd \
-H "if-match:1-9533bf3b3c5cda717ed000b186fdafe"
{"ok":true,"id":"7341477ce373f9cc76f351e598001cdd",
"rev":"2-5c7fb5dfeaf6f7cea149922fa1cdaf96"}
```

3.10 Dokument-Attachment speichern [PUT]

(*params*) rev=revision
(*headers*) content-length:[bytes], content-type:[MIME-type document], if-match:revision
(!) generell könnte auch nur die Option `--data` für eine reine Textdatei
verwendet werden

```
curl -X PUT http://127.0.0.1:5984/kina/ \
b7fc0ca7dffcc6d0f240c1e5bb000998/ \
geburtstag.txt?rev=2-1726cd3e40ffd356591114f014b1ac22 \
--data-binary @geburtstag.txt \
-H "content-type: text/plain;charset=utf-8"
```

```
HTTP/1.1 201 Created
Server: CouchDB/1.1.0 (Erlang OTP/R14B03)
Location: http://127.0.0.1:5984/datenbankname/
b7fc0ca7dffcc6d0f240c1e5bb000998/geburtstag.txt
Etag: "2-1726cd3e40ffd356591114f014b1ac22"
Date: Mon, 07 Feb 2011 22:53:25 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 66
Cache-Control: must-revalidate
```

```
{"ok":true,"id":"b7fc0ca7dffcc6d0f240c1e5bb000998",
"rev":"2-1726cd3e40ffd356591114f014b1ac22"}
```

3.11 Dokument-Attachment anfragen [GET]

```
curl -X GET http://127.0.0.1:5984/kina/ \
b7fc0ca7dffcc6d0f240c1e5bb000998/geburtstag.txt
```

```
HTTP/1.1 200 OK
Server: CouchDB/1.1.0 (Erlang OTP/R14B03)
Etag: "2-1726cd3e40ffd356591114f014b1ac22"
Date: Sun, 28 Aug 2011 20:32:14 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 66
Cache-Control: must-revalidate
Accept-Ranges: none
```

3.12 Dokument-Attachment löschen [DELETE]

(*params*) rev=revision
(*headers*) if-match:revision

```
curl -X GET http://127.0.0.1:5984/kina/ \
b7fc0ca7dffcc6d0f240c1e5bb000998/geburtstag.txt
{"ok":true,"id":"b7fc0ca7dffcc6d0f240c1e5bb000998",
"rev":"3-f1d8176c2ea671ad75bd85a55dea4d05"}
```

3.13 Dokument permanent löschen [POST] [_purge]

(!) Referenz zum Dokument wird gelöscht; wird nicht repliziert. Um den
Plattenplatz auch frei zu bekommen, muss `--_compact` ausgeführt werden

```
curl -X POST http://127.0.0.1:5984/kina/_purge/ \
-H "content-type:application/json" \
-d '{"7341477ce373f9cc76f351e598001cdd":
["2-5c7fb5dfeaf6f7cea149922fa1cdaf96"]
}'
{"purge_seq":1,"purged":{"
7341477ce373f9cc76f351e598001cdd":
["2-5c7fb5dfeaf6f7cea149922fa1cdaf96"]
}}
```

3.14 Dokument Info [HEAD]

(*params*) rev=revision, revs=[true,false], revs_info=[true,false]

```
curl --head -X GET http://127.0.0.1:5984/kina/ \
fab911e6e9ed703bf536dcccfe000937?revs=true
```

```
HTTP/1.1 200 OK
Server: CouchDB/1.1.0 (Erlang OTP/R14B03)
Etag: "1-183f162a09532ccacf8223f6e2ba95f6"
Date: Sun, 28 Aug 2011 14:40:35 GMT
Content-Type: text/plain;charset=utf-8
Content-Length: 175
Cache-Control: must-revalidate
```

3.15 Dokument kopieren [COPY]

(i) Die ID 7341477ce373f9cc76f351e598001d4c wurde zuerst per `_uuids request` erfragt. Um in ein bestehendes Dokument zu kopieren, wird die ID des Zieldokuments angegeben (*header*) `Destination:docid` (*params*) `rev=revision`

```
curl -X COPY http://127.0.0.1:5984/kina/ \
  fab911e6e9ed703bf536dccffe000937 \
  -H "destination:7341477ce373f9cc76f351e598001d4c"
{"id": "7341477ce373f9cc76f351e598001d4c",
 "rev": "1-5a5db37cb3735e769ea8d133b28f04a1"}
```

3.16 Mehrere Dokumente anlegen [POST] [`_bulk_docs`]

(*options*) `all_or_nothing:[true,false (=non-atomic=default)]`, `docs:[key:value,...]`

```
curl -X POST http://127.0.0.1:5984/kina/_bulk_docs \
  -H "content-type:application/json" \
  -d '{"docs": [
    {"lang": "c"},
    {"lang": "c++"}
  ]}'
[{"id": "7341477ce373f9cc76f351e5980008bb",
 "rev": "1-f491c132ea24ecf492e0b5ae18467876"},
 {"id": "7341477ce373f9cc76f351e5980015cd",
 "rev": "1-9533bf3b3c5cda717ed000b186fdafae"}]
```

3.17 Mehrere Dokumente aktualisieren [POST] [`_bulk_docs`]

(*options*) `all_or_nothing:[true,false (=non-atomic=default)]`, `docs:[key:value,...]`

```
curl -X POST http://127.0.0.1:5984/kina/_bulk_docs \
  -H "content-type:application/json" \
  -d '{"docs": [
    {"_id": "7341477ce373f9cc76f351e5980008bb",
     "_rev": "1-f491c132ea24ecf492e0b5ae18467876",
     "lang": "erlang"},
    {"_id": "7341477ce373f9cc76f351e5980015cd",
     "_rev": "1-9533bf3b3c5cda717ed000b186fdafae",
     "lang": "lisp"}
  ]}'
[{"id": "7341477ce373f9cc76f351e5980008bb",
 "rev": "2-f0bfca3976ad04bce05b2ade242519d7"},
 {"id": "7341477ce373f9cc76f351e5980015cd",
 "rev": "2-d6f5f2cb326c1f68f95d2bfbef329280"}]
```

Literatur

- [1] <http://couchdb.apache.org>
- [2] <https://cloudant.com/futon/database.html?andywenk%2Fkina>
- [3] <http://wiki.apache.org/couchdb/Reference>
- [4] <http://www.erlang.org>
- [5] http://en.wikipedia.org/wiki/Representational_State_Transfer
- [6] <http://en.wikipedia.org/wiki/ACID>
- [7] http://en.wikipedia.org/wiki/Multiversion_concurrency_control
- [8] <http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>
- [9] <http://en.wikipedia.org/wiki/MapReduce>
- [10] <http://curl.haxx.se/>
- [11] <https://gist.github.com/832610>