mongo

Übersicht

MongoDB

- mongod
- mongo
- mongoimport

Datenformate

- JSON
- BSON

Komponenten

neue Bibliotheken

- libmongoc
  - bson.h
  - mongo.h
- json-glib
- icu

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

mongoDB

mongod

Parameter

- `mongod --auth --db-path` *`DB-PFAD`*

mongo

nicht-interaktiver Modus

- `mongo --eval` *`JAVASCRIPT-CODE`*

mongoimport

nicht-interaktiver Modus

- `mongoimport --db` *`DB`* `--collection` *`COLLECTION`* `--file` *`DATEI`*

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

Datenformate JSON / BSON

JSON text

```
01 { "string" : "xxx",
02   "number" : 1
03   "array" : [ 1, 1.0, 0x1000000 ],
04   "object" : {
05     "boolean" : true
06     "null" : null
07   }
08 }
```

BSON document

```
01 string : 2    xxx
02 number : 1    1.000000
03 array : 4
04     0 : 1     1.000000
05     1 : 1     1.000000
06     2 : 1     4294967296.000000
07 object : 3
08     boolean : 8   true
09     null : 10     BSON_NULL
```

Datentypen

JSON	BSON	ID
string	UTF-8 string	2
number	Floating point 32-bit Integer 64-bit Integer	1 16 17
boolean	Boolean	8
null	Null value	10
object	Embedded document	3
array	Array	4
	Binary data ObjectId UTC datetime Regular expression JavaScript code Symbol JavaScript code w/ scope Timestamp	5 7 9 11 13 14 15 17

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

libmongoc [bson.h]

Codeschnippsel

```
01 #include <mongo.h>
02
03 int status;
04 bson b;
05
06 bson_init (&b);
07
08 bson_append_... (&b, ...);
09   ...
10
11 if (bson_finish (&b) != BSON_OK)
12   ...
13
14 bson_destroy (&b);
```

BSON Dokument

JSON	libbson API		ID
string	bson_append_string bson_append_string_n	b, c, c b, c, c, i	2
number	bson_append_int bson_append_long bson_append_double	b, c, i b, c, i_64 b, c, d	1 16 17
boolean	bson_append_bool	b, c, i	8
null	bson_append_null	b, c	10
object	bson_append_start_object bson_append_finish_object	b, c b	3
array	bson_append_start_array bson_append_finish_array	b, c b	4

bson *b const char *c int i int64_t i_64 double d

Rückgabewert

- BSON_OK = 0
- BSON_ERROR
- bson -> err

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

libmongoc [bson.h]

Codeschnippsel

```
01 #include <mongo.h>
02
03 bson_type type;
04
05 while ((t=bson_iterator_type (&b)
06   != BSON_EOO))
07   ...
08
09
10 while (bson_iterator_more (&i)) {
11   type = bson_iterator_next (&b)
12 }
13
14
```

"unsichere" Varianten

- mit Postfix _raw

Iteratoren

Typ	libbson API	
c i	bson_iterator_string bson_iterator_string_len	 I
i i_64 d	bson_iterator_int bson_iterator_long bson_iterator_double	I I I
i	bson_iterator_bool	I
c	bson_iterator_key	I
t	bson_iterator_type	I
i	bson_iterator_more	I
t	bson_iterator_next	I
v	bson_iterator_subiterator	I, J

const char *c int i int64_t i_64 double d
bson_iterator *I, *J bson_type t void v

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

libmongoc [mongo.h]

Codeschnippsel

```
01 #include <mongo.h>
02
03 #define HOST "localhost"
04 #define PORT 27017
05
06 mongo c[1];
07 mongo_init (conn);
08
09 if (mongo_connect (c, HOST, PORT) != MONGO_OK)
10    ...
11
12 mongo_cursor cursor[1];
13 mongo_cursor_init (cursor, c,
14 "DB.COLLECTION");
15
16 while (mongo_cursor_next (cursor) == MONGO_OK)
17    bson_print (&cursor->current);
18
19 mongo_cursor_destroy (cursor);
20
21 mongo_destroy (conn);
22
```

Connection

	libmongocAPI	
i	mongo_connect	m, c, i
v	mongo_destroy	m

mongo *m const char *c int i void v

Cursor

	libmongocAPI	
v	mongo_cursor_init	C, m, c
v	mongo_cursor_set_query	C, b
i	mongo_cursor_next	C
i	mongo_cursor_destroy	C

mongo_cursor *C
mongo *m const char *c int i void v

Rückgabewert

- MONGO_OK = 0
- MONGO_ERROR
- mongo -> err

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

libmongoc [mongo.h]

Codeschnippsel

```
01 ...
02 bson query[1];
03 bson_init (query);
04 bson_append_start_object (query, "$query");
05 bson_append_string (query, "key", "avoir");
06 bson_append_finish_object (query);
07 bson_finish (query);
08
09 mongo_cursor cur[1];
10 mongo_cursor_init (cur, conn, "lug.frz");
11 mongo_cursor_set_query (cursor, query);
12
13 while ((mongo_cursor_next (cur)) == MONGO_OK)
14   ...
15
16
17 bson out[1];
18 mongo_simple_int_command (conn, "admin",
19                           "listDatabases", 1, out);
20 bson_print (out);
```

Command

	libmongocAPI	
i	mongo_simple_int_command	m, c, c, i, b
i	mongo_simple_str_command	m, c, c, c, b
i	mongo_run_command	m, c, c, b, b

mongo *m bson *b const char *c int i

Queries

BSON-Dokument

```
{  $query   : { ... },
   $orderby : { ... }
   ...  }
```

Kommandos

BSON-Dokument

```
{  count      : ... }
{  dropDatabase  :  1 }
 ...
```

- einige auf "spezieller" Datenbank "admin"

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

json-glib

JsonNode

Typ	libbson API	
JsonNodeType	json_node_get_node_type	N
GType	json_node_get_value_type	N
JsonObject*	json_node_get_object	N
JsonArray*	json_node_get_array	N
const gcharr* gchar*	json_node_get_string json_node_dup_string	N N
gboolean	json_node_get_boolean	N
gint64 gdouble	json_node_get_int json_node_get_double	N N

JsonNodeType

- JSON_NODE_OBJECT
- JSON_NODE_ARRAY
- JSON_NOE_VALUE
- JSON_NODE_NULL

JsonArray

Typ	libbson API	
JsonNode*	json_array_get_element	N, i
...	json_array_get_..._elementt	N, i
guint	json_array_get_length	N
GList*	json_array_get_elements	N
void	json_array_foreach_element	N, F

- JsonForeachElement (*F) (A, i, N, p)

JsonObject

Typ	libbson API	
JsonNode*	json_object_get_member	N, c
...	json_object_get_..._member	N, c
guint	json_object_get_length	N
GList*	json_object_get_elements	N
void	json_object_foreach_member	N, G

- JsonForeachMember (*G) (O, c, N, p)

JsonNode *N
const gchar *c

JsonArray *A
guint i

JsonObject *O
gpointer p

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

icu

Codeschnippsel

```
01 #include <unicode/utypes.h>
02 #include <unicode/ucol.h>
03 #include <unicode/ustring.h>
04 ...
05 UChar *utf2uchar (gchar *src) {
06   UErrorCode status;
07   UChar *dest;
08   int32_t size;
09
10   u_strFromUTF8 (NULL, 0, &size,
11                  src, -1, &status);
12   dest = g_new (UChar, size);
13
14   u_strFromUTF8 (dest, size, &size,
15                  src, -1, &status);
16
17   return dest;
18 }
19 ...
```

Codeschnippsel

```
01 int icu_strcmp (UCollator *collator,
02                 gchar* s, gchar* t)
03 {
04   UCollationResult result;
05
06   UChar *source = utf2uchar (s);
07   UChar *target = utf2uchar (t);
08
09   result = ucol_strcoll (collator,
10 source, -1, target, -1);
11
12   g_free (source);
13   g_free (target);
14
15   switch (result) {
16     case UCOL_LESS:
17       return -1;
18     case UCOL_GREATER:
19       return 1;
20     default:
21       return 0;
22   }
23 }
```

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.

mongo

icu

Kollatoren

Codeschnippsel

```
01 UErrorCode status = U_ZERO_ERROR;
02 UCollator *col;
03 UChar rules[100];
04 const gchar *locale = "";
05
06 col = ucol_open (locale, &status);
07 /* oder ?*/
08 col = ucol_openRules (rules, -1, UCOL_OFF, UCOL_TERTIARY, NULL, &status);
09
10 ucol_setAttribute (col,UCOL_NORMALIZATION_MODE, UCOL_ON, &status);
11
12 if (U_FAILURE (status))
13   ...
14
15 ...
16
17 ucol_close (collator);
```

PUBLIC DOMAIN

To the extent possible under law, the person who associated CC0 with this work has waived all copyright and related or neighboring rights to this work. This work is published from: Germany.