

Exploring the Database

I'm running my Bookstack in a docker container, so for me if I want to explore my database the first step is to get a bash terminal within the container. Lets say you've named your bookstack database container `db_bookstack` - the command to enter a bash terminal within the container would be `docker exec -it db_bookstack bash`

Once you have a terminal, login to the database by running `mysql -u username -p` and enter the password to login when prompted. If you setup your bookstack with a docker container this should have been configured within your `docker-compose.yml` that you use to spin up your services.

Once your logged in, you'll see a mysql prompt like the below -

```
MariaDB [(none)]>
```

The prompt shows `(none)` because we haven't selected a database. Run `show databases;` to show the databases available to you and select one with `use database;`, see below for an example.

```
MariaDB [(none)]> show databases;
```

```
+-----+  
| Database      |  
+-----+  
| bookstack     |  
| information_schema |  
+-----+  
2 rows in set (0.000 sec)
```

```
MariaDB [(none)]> use bookstack;
```

```
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
MariaDB [bookstack]>
```

NOTE: The names of these databases do not reflect any actual databases

Now that we've selected the database we want to explore, let's see what it contains by running `show tables;` -

```
MariaDB [bookstack]> show tables;
```

```
+-----+
| Tables_in_bookstack |
+-----+
| activities          |
| api_tokens          |
| attachments         |
| books               |
| bookshelves         |
| bookshelves_books   |
| cache               |
| chapters            |
| comments            |
| email_confirmations |
| entity_permissions  |
| images              |
| joint_permissions   |
| migrations          |
| page_revisions      |
| pages               |
| password_resets     |
| permission_role      |
| role_permissions     |
| role_user           |
| roles               |
| search_terms        |
| sessions            |
| settings            |
| social_accounts     |
| tags                |
| user_invites        |
| users              |
| views               |
+-----+
```

```
29 rows in set (0.000 sec)
```

See something you want to explore further? Describe it by running `describe table;`, like below where I describe the `books` table.

```
MariaDB [bookstack]> describe books;
```

```
+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+
| id         | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| name       | varchar(191)    | NO   |     | NULL    |              |
| slug       | varchar(191)    | NO   | MUL | NULL    |              |
| description | text            | NO   |     | NULL    |              |
| created_at | timestamp       | YES  |     | NULL    |              |
| updated_at | timestamp       | YES  |     | NULL    |              |
| created_by | int(11)         | NO   | MUL | NULL    |              |
| updated_by | int(11)         | NO   | MUL | NULL    |              |
| restricted | tinyint(1)      | NO   | MUL | 0       |              |
| image_id   | int(11)         | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

Neat! We have all the tools we need to step through any database that we have access to explore, and now we just need to be able to list the actual content within the tables of the database. Below, we run `select` and `from` commands that refer to specific tables and elements within our database. After the commands are input, the tables and their values we selected are displayed. See below for an example.

```
MariaDB [bookstack]> select id, name, created_at, updated_at, created_by from books;
```

```
+---+-----+-----+-----+-----+
| id | name          | created_at      | updated_at      | created_by |
+---+-----+-----+-----+-----+
| 1  | Pi            | 2019-04-06 00:00:35 | 2019-05-17 15:41:22 | 1 |
| 4  | Docker        | 2019-04-06 00:02:12 | 2020-02-24 03:26:34 | 1 |
| 7  | BookStack     | 2019-04-06 04:41:38 | 2019-05-17 20:42:20 | 4 |
| 9  | Self-Hosted Applications | 2019-04-06 12:12:51 | 2019-07-06 14:17:31 | 4 |
| 13 | Networking    | 2019-04-19 06:48:57 | 2019-07-06 12:54:22 | 4 |
| 16 | Security      | 2019-05-04 15:02:19 | 2019-05-12 13:53:57 | 4 |
| 20 | Linux Admin   | 2019-07-06 12:57:22 | 2019-07-06 13:20:10 | 4 |
| 21 | Git           | 2019-07-19 03:11:03 | 2019-07-26 21:54:29 | 4 |
| 23 | Vim           | 2019-07-21 05:15:16 | 2019-08-30 08:22:47 | 4 |
| 24 | Ansible       | 2019-07-28 08:17:04 | 2019-08-30 07:58:16 | 4 |
| 28 | C++           | 2019-10-13 20:24:23 | 2020-03-01 14:31:16 | 4 |
+---+-----+-----+-----+-----+
```

