
DBCore (ALPHA)

DBCore is a code generator build around database schemas and an API specification. Included with DBCore are templates for generating a Go REST API and React UI.

Features and API specification

While the DBCore project can build any templates from your database. It also defines an API specification with useful functionality for rapidly standing up an API around your database.

Because DBCore does code generation, it can build well-typed code. The built-in Go API templates are a great example of this.

But since the API specification is language-agnostic, all these features are supported no matter what language you use to generate a DBCore API.

Major features include:

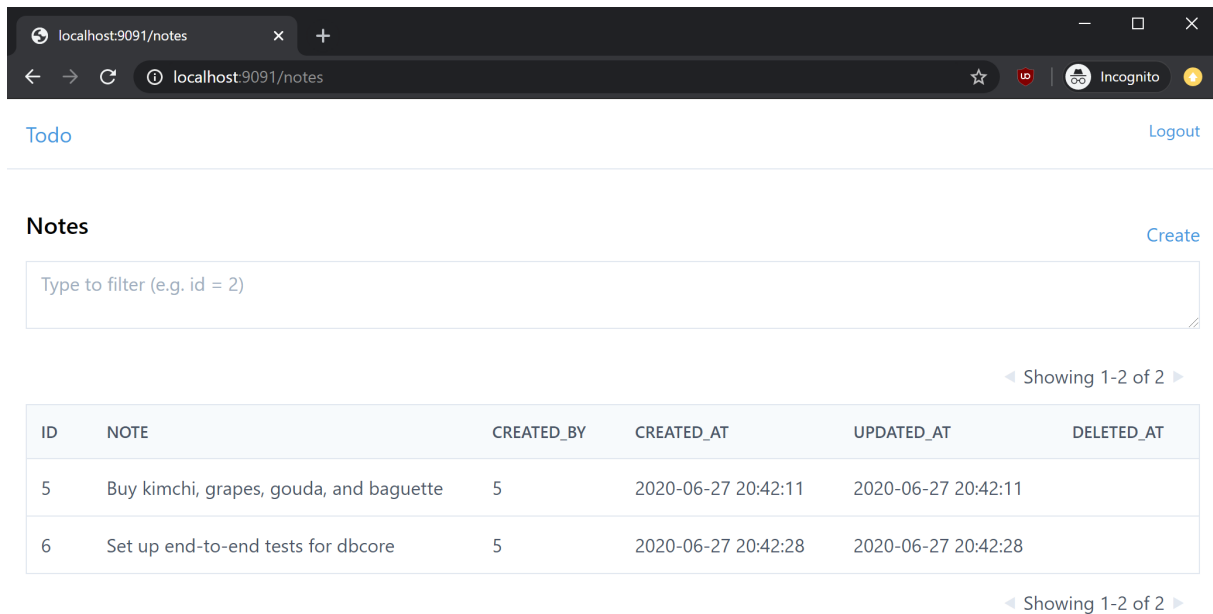
- Get one, get many, create, edit, delete endpoints
- Filtering, sorting, pagination
- JWT-based authentication, per-endpoint/method SQL filter-based authorization

Upcoming features include:

- Lua-based hooks and transformations
- SSO integration

See the docs site for more detail.

Example



localhost:9091/notes

← → ↻ ⓘ localhost:9091/notes ☆ | Incognito

[Todo](#) [Logout](#)

Notes [Create](#)

Type to filter (e.g. id = 2)

◀ Showing 1-2 of 2 ▶

ID	NOTE	CREATED_BY	CREATED_AT	UPDATED_AT	DELETED_AT
5	Buy kimchi, grapes, gouda, and baguette	5	2020-06-27 20:42:11	2020-06-27 20:42:11	
6	Set up end-to-end tests for dbcore	5	2020-06-27 20:42:28	2020-06-27 20:42:28	

◀ Showing 1-2 of 2 ▶

There's a built-in notes application with non-trivial authorization. Users belong to an org. Notes belong to a user. Notes that are marked public don't need a session. Otherwise they can only be viewed by other users within the same org. Only org admins or the notes creator can modify a note.

```
1 $ git clone git@github.com:eatonphil/dbcore
2 $ cd dbcore
3 $ make example-notes
4 $ cd ./examples/notes/api
5 $ ./main
6 INFO[0000] Starting server at :9090 pkg=server struct=Server
7 ... in a new window ...
8 $ curl -X POST -d '{"username": "alex", "password": "alex", "name": "Alex"}' localhost:9090/users/new
9 {"id":1,"username":"alex","password":"alex","name":"Alex"}
10 $ curl 'localhost:9090/users?limit=25&offset=0&sortColumn=id&sortOrder=desc' | jq
11 {
12   "total": 1,
13   "data": [
14     {
15       "id": 1,
16       "username": "alex",
17       "password": "alex",
18       "name": "Alex"
19     }
15 }
```

```
20   ]  
21 }
```

And to build the UI:

```
1 $ cd examples/notes/browser  
2 $ yarn start
```

Log in with any of the following credentials:

- admin:admin (Org 1)
- notes-admin:admin (Org 2)
- editor:editor (Org 2)

Dependencies

- Go
- PostgreSQL, MySQL or SQLite3
- .NET Core

Restrictions

There are a bunch of restrictions! Here are a few known ones. You will discover more and you may fix them!

- Only tables supported (i.e. no views)
- Only single-column foreign keys supported
- Only Go API, React UI templates provided