**Prisma Tutorial**

Which Database to use?

Initial Prisma installation and setup 2

Which Database to use?

What capabilities - what does my application need to do

Hosting

* What does the application need to do
* What is the cost
* Bottlenecks/hurdles/trouble

Interaction

* How it is accessed through the client in my code
* How is it accessed over a network and how much control I have over who can access it

CRUD applications

Create, Read, Update, Delete. These sorts of applications are when users access and create data like a forum or blog or indeed a marketplace application such as mine. Relational databases are a tried and tested solution for this level of storing data in tables that are joined together using a key. Uncomplicated, been around for a long time, large support network. ‘Standard applications’. There is also a lot of protection built into the database.

Firstly the documentation is all available on the prisma website with examples: <https://www.prisma.io/docs/getting-started>

Graphical user interface, application, Teams

Description automatically generated

We also install the @prisma/client which we will use later on in the process as it allows us to use the prisma client class to perform database queries within the application.

Looking into what has now been done by running the ‘yarn prism init’ command, we can see that a prisma folder has been added. In this folder we now have a schema file which is where we do almost all of our configuring.

Firstly, we have the generator definition which is used to generate new clients when we make changes to our scheme. Every time we make a change to a table or column etc within our database we need to generate a new client in order for the changes to be reflected in our db. This client generates models allowing us to interact with the db.

Secondly the datasource definition where we define which database we’re using and how to connect to it. We will be using mysql for this tutorial so the provider will need to be updated to reflect this.

Now lets look at the .env file. We currently only have one entry which is the database url which is defaulted to postgress here so lets update that as well.



And again with the updated versions of our username and password.



This username and password are the admin username and password for server access within the mysql workbench. These are set when creating the server on which the database will be stored.

One interesting point to make with Prisma is that it can perform something called ‘introspecting’. With most ORMs you would need to create the models for your database within the schema file and then perform a migration to create the database tables in your actual database. With Prisma, however, it is possible to perform the reverse of this and create a database separately from your application then use prisma to import (introspect) the models into your schema.prisma file which will either create new models or update existing ones for you. How nice. Handy for migrations to other databases. Saying all that, using migrations is the standard so lets go down this route in this tutorial.

Also Interestingly, when I tried to run mysql on the command line I got an error not recognising the command. This was due to a path not being set up for cmd to accept forming as a functional word. This was easily rectified in the advanced system preferences under the global paths section.

The next job is to open up a mysql command line and connect to our server. Then create the database and connect to that.

Text

Description automatically generated

If there’s a problem starting the mysql server there is a handy guide here:

<https://phoenixnap.com/kb/connect-to-mysql-windows-command-line#:~:text=Enter%20mysql.exe%20%2Duroot%20%2D,connect%20to%20the%20MySQL%20server>.

CREATE DATABASE nextjs\_prisma;

It’s SQL so the pesky semi-colon back in force.

USE nextjs\_prisma;

Now we can do anything we want through the command line to mess around with this database. What we really want to do though is to start storing some data. We’re going to need user details for our actual project so lets create a user model in our schema.

Now, prisma has a number of commands that can be run on the command line:

Text

Description automatically generated

Attempting to pull or generate at this stage would result in an error as there is nothing yet to pull from and there are no models in our schema to generate with.

Text

Description automatically generated

Yarn prisma db push will get the ball rolling. Now we have a db with a table called user.

If we run the yarn prisma studio command then we can look at our new db and the table we’ve created and also add an entry if we like. How nice.

You can retrieve an avatar from github by using the url https://github.com/{yourname}.png