

4. Entity - Post.ts

#mikroorm #entity #backend

How to implement an Entity

(Later we will [Convert the Post entity_\(class\) to a GraphQL type](#): )

We use the `@Entity()`, `@PrimaryKey()` and `@Property()` attributes imported from `@mikro-orm/core` to define the **columns** of the db table (**entity**) Posts as follows:

entities/Posts.ts


```
import { Entity, PrimaryKey, Property } from "@mikro-orm/core";

@Entity()
export class Post {
  @PrimaryKey()
  id!: number;

  @Property({ type: 'date' }) // explicitly set type for MikroORM otherwise it
  // infers is as "jsonb" type
  createdAt = new Date();

  @Property({ type: 'date', onUpdate: () => new Date() })
  updatedAt = new Date();

  @Property({ type: 'text' })
  title!: string;
}
```

- `entities: [Post, User],` in `mikro-orm.config.ts` should be updated everytime we add a new entity
- after creating the entity and updating the above parameter we run `mikro-orm migration:create` to update the database create tables and create the new tables if necessary. (see [3. MikroORM](#) )