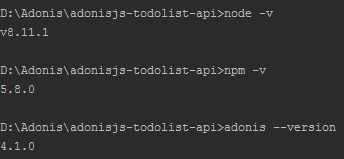
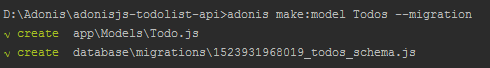
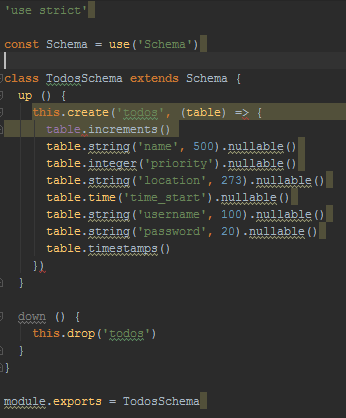
1. Pastikan Node, NPM dan Adonis sudah terinstall
2. Pada API ini saya menggunakan versi Node, NPM dan Adonis sebagai berikut :



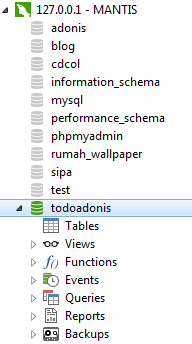
1. Siapkan schema database



1. Ubah schema tadi sesuai kebutuhan / request



1. Pastikan database mysql sudah aktif dan buat database baru (saya beri nama todoadonis) :



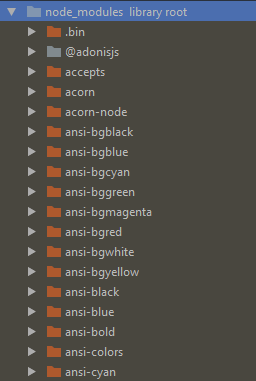
1. Pastikan configurasi database pada **.env**

DB\_CONNECTION=mysql  
DB\_HOST=127.0.0.1  
DB\_PORT=3306  
DB\_DATABASE=todoadonis  
DB\_USERNAME=root  
DB\_PASSWORD=

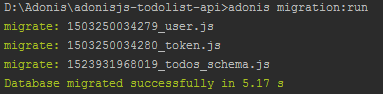
1. Agar dapat digunakan install Mysql module terlebih dahulu



Maka akan terbentuk file seperti berikut :

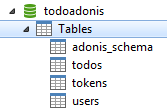


1. Jalan kan perintah berikut ini, agar table pada schema terbentuk



1. Hasilnya akan terbentuk table seperti berikut :

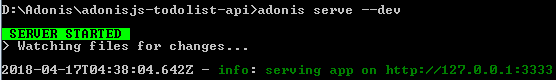
*\*note : terdapat 2 cara bisa menggunakan perintah diatas untuk membuat table atau import database todoadonis.sql..*



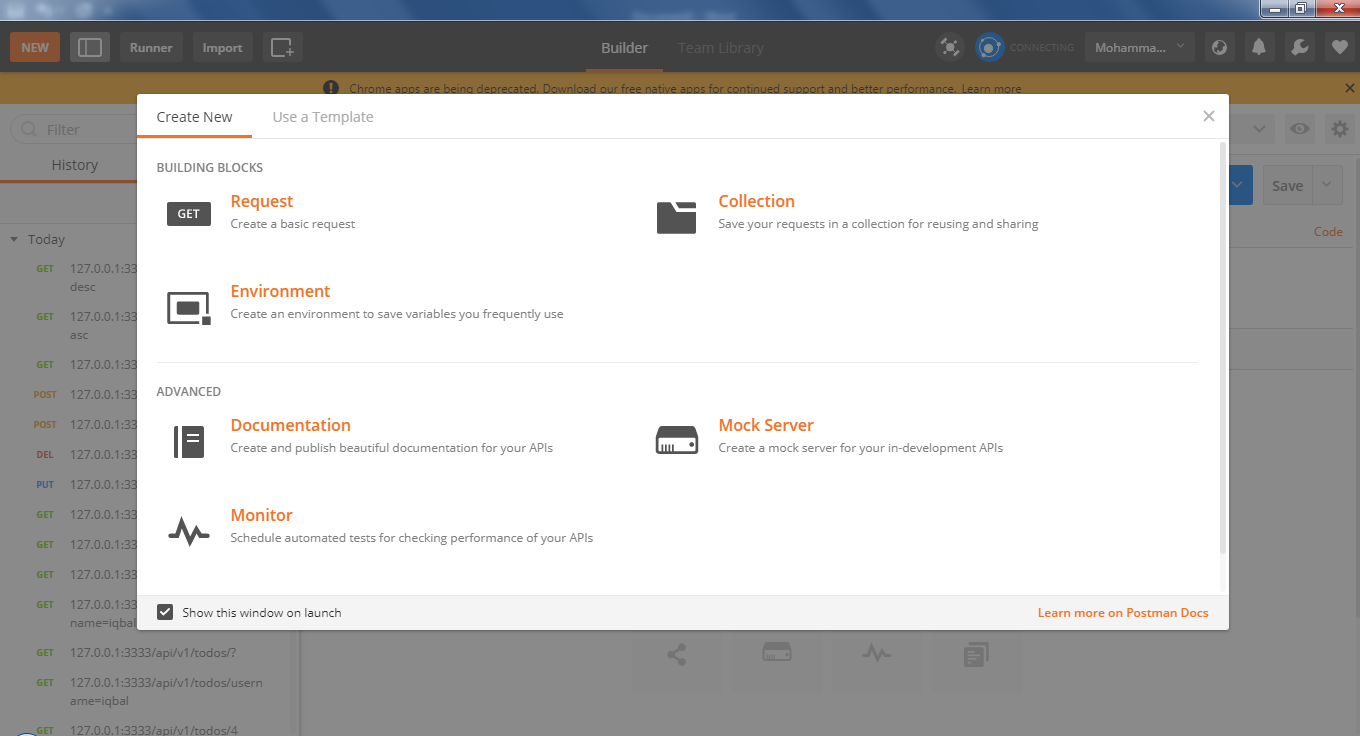
1. Routing yang saya gunakan untuk API ini adalah :

Route.post('todos', 'TodoController.store')  
Route.get('todos', 'TodoController.index')  
Route.get('todos/:username', 'TodoController.show')  
Route.get('todos/:username/:sorting', 'TodoController.priority')  
Route.put('todos/:id', 'TodoController.update')  
Route.delete('todos/:id', 'TodoController.delete')

1. Pastikan server Adonis sudah aktif



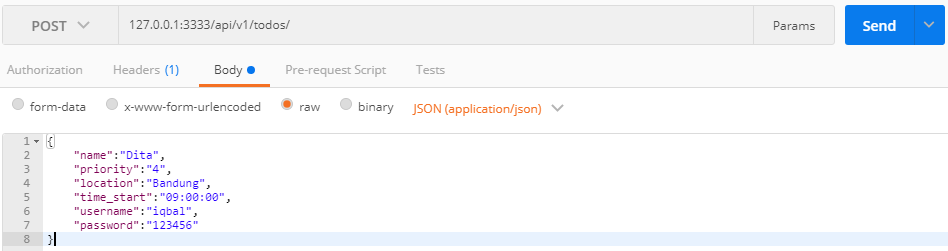
1. Aktifkan Postman untuk melakukan uji coba API



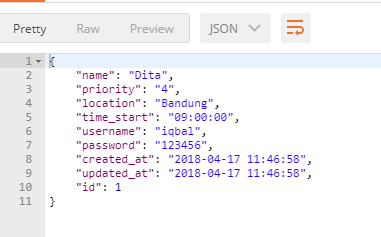
1. Uji coba pertamasimpan data

**Route.post('todos', 'TodoController.store')** (POST : 127.0.0.1:3333/api/v1/todos/)

Request :



Response :



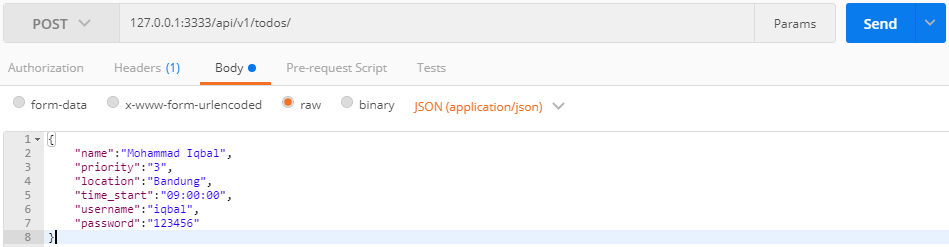
Hasil :



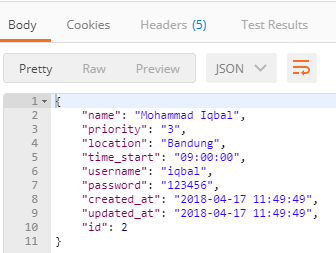
1. Uji Coba kedua simpan data

**Route.post('todos', 'TodoController.store')** (POST : 127.0.0.1:3333/api/v1/todos/)

Request :



Response :



Hasil :



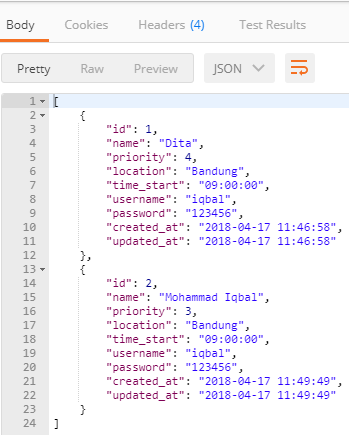
1. Selanjutnya untuk melihat semua data yang sudah tersimpan

Route.get('todos', 'TodoController.index') (GET : 127.0.0.1:3333/api/v1/todos/)

Request :



Response :



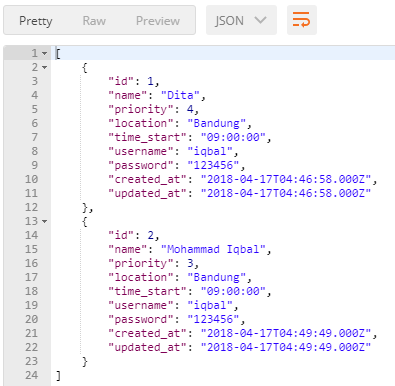
1. Selanjutnya untuk melihat semua data yang tersimpan berdasarkan username

Route.get('todos/:username', 'TodoController.show') (GET : 127.0.0.1:3333/api/v1/todos/:username)

Request :



Response :



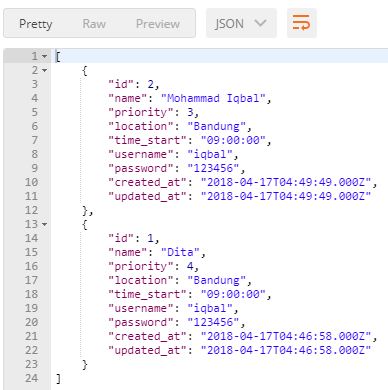
1. Selanjutnya untuk melihat semua data yang tersimpan berdasarkan username dan prioritas (sorting). Route.get('todos/:username/:sorting', 'TodoController.priority')

(GET : 127.0.0.1:3333/api/v1/todos/:username/:sorting)

Request :



Response :



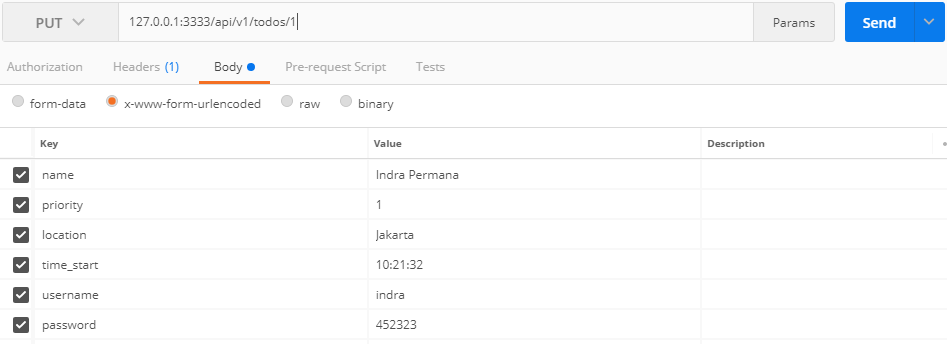
1. Selanjutnya untuk update data

Route.put('todos/:id', 'TodoController.update') (PUT : 127.0.0.1:3333/api/v1/todos/:id)

Data Sebelumnya :



Request :



Response :



Hasil :



1. Selanjutnya untuk delete data

Route.delete('todos/:id', 'TodoController.delete') (DELETE : 127.0.0.1:3333/api/v1/todos/:id)

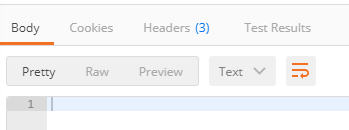
Data Sebelumnya :



Request :



Response :



Hasil :

