1. Create new to do list (@PostMapping("/todos")):

Ini adalah endpoint untuk membuat tugas baru dalam daftar. Metode createTodo mengambil data tugas dari body permintaan (@RequestBody Todo todo) dan menyimpannya menggunakan todoRepository.save(todo).

2. Get all existing to do list (@GetMapping("/todos")):

Ini adalah endpoint untuk mendapatkan semua tugas dalam daftar. Metode getAllTodos dapat mengambil parameter opsional seperti creator (pencipta tugas) dan date (tanggal). Berdasarkan parameter-parameter ini, metode akan mencari tugas-tugas yang sesuai dengan kriteria tertentu dalam todoRepository.

3. Update existing to do list (@PutMapping("/todos/{id}")):

Endpoint ini memungkinkan Anda memperbarui tugas yang sudah ada dalam daftar. Metode updateTodo mengambil ID tugas dari path (@PathVariable Long id) dan data yang diperbarui dari body permintaan (@RequestBody Todo updatedTodo). Metode ini akan mencari tugas dengan ID yang diberikan, memperbarui properti-properti tugas dengan data yang baru, dan kemudian menyimpan perubahan menggunakan todoRepository.save(todo).

4. Delete to do list (@DeleteMapping("/todos/{id}")):

Ini adalah endpoint untuk menghapus tugas dari daftar. Metode deleteTodo mengambil ID tugas dari path dan mencari tugas dengan ID tersebut dalam todoRepository. Jika ditemukan, tugas akan dihapus menggunakan todoRepository.delete(todo).

5. Get to do list by ID (@GetMapping("/todos/{id}")):

Endpoint ini memungkinkan Anda untuk mendapatkan detail tugas berdasarkan ID-nya. Metode getTodoById mengambil ID tugas dari path dan mencari tugas dengan ID tersebut dalam todoRepository.

6. Mark to do list as complete (@PatchMapping("/todos/{id}/complete")):

Ini adalah endpoint untuk menandai tugas sebagai selesai. Metode markTodoAsCompleted mengambil ID tugas dari path, mencari tugas dengan ID tersebut, mengubah status tugas menjadi selesai, dan kemudian menyimpan perubahan.

7. Get complete to do list (@GetMapping("/todos/completed")):

Endpoint ini memungkinkan Anda untuk mendapatkan daftar tugas yang telah selesai.

8. Get overdue to do list (@GetMapping("/todos/overdue")):

Endpoint ini mengembalikan daftar tugas yang jatuh tempo belum tercapai (overdue). Metode ini menggunakan tanggal saat ini untuk membandingkan dengan tanggal jatuh tempo tugas.

9. Get to do list by priority (@GetMapping("/todos/priority/{priority}")):

Endpoint ini memungkinkan Anda untuk mendapatkan daftar tugas berdasarkan prioritas yang diberikan.

10. Update to do list priority (@PutMapping("/todos/{id}/priority")):

Ini adalah endpoint untuk memperbarui prioritas tugas. Metode updateTodoPriority mengambil ID tugas dari path dan parameter prioritas dari query (@RequestParam String priority). Metode ini mencari tugas berdasarkan ID dan memperbarui prioritasnya.

Seluruh implementasi ini bergantung pada objek todoRepository, yang diasumsikan sebagai antarmuka untuk mengakses dan berinteraksi dengan penyimpanan data (misalnya, basis data) yang menyimpan informasi tentang tugas-tugas. Kode ini menyiratkan adanya model Todo yang mungkin mencakup properti seperti title, description, dueDate, creator, completed, dan priority.