

Ringkasan: Pada tutorial kali ini, saya belajar menggunakan logging untuk melakukan debugging suatu method serta menjalankan query sql untuk menginput data, update, delete data di database

Pertanyaan

1. Jika menggunakan Object sebagai parameter pada form POST, bagaimana caranya melakukan validasi input yang optional dan input yang required? Apakah validasi diperlukan? Asumsikan input pada form Anda tidak menggunakan attribute required sehingga butuh validasi di backend.

Jawab: Validasi input yang optional dan input yang required dapat dilakukan dengan menggunakan if statement serta menggunakan BindingRequest dan @Valid untuk pengecekan apabila terjadi error. Validasi diperlukan.

2. Menurut Anda, mengapa form submit biasanya menggunakan POST method dibanding GET method? Apakah perlu penanganan berbeda di header atau body method di controller jika form di post dikirim menggunakan method berbeda?

Jawab: Dengan tujuan agar hasil form yang diinput muncul pada url setelah di-submit. Diperlukan dan dapat dengan RequestMethod.

3. Apakah mungkin satu method menerima lebih dari satu jenis request method, misalkan menerima GET sekaligus POST?

Jawab: Tidak karena GET dan POST merupakan method yang berbeda

Latihan Menambahkan Delete

Pada latihan ini, saya menambahkan method deleteStudent dan anotasi @Delete untuk menjalankan query apabila method deleteStudent dipanggil. Apabila npm ditemukan, maka akan memanggil method deleteStudent dan menghapus object student dari database setelah itu ke-direct ke halaman delete.html. Apabila tidak maka akan ke-direct ke not-found.html

Latihan Menambahkan Update

Pada latihan ini, saya membuat method updateStudent pada StudentMapper dengan param StudentModel serta anotasi @Update untuk menjalankan query apabila method updateStudent

dipanggil. Setelah itu saya mengimplementasikan method `updateStudent` pada `StudentService` dan `StudentServiceDatabase` untuk menjalankan method `updateStudent`. Setelah itu, saya mengedit `form-update`. Selain itu saya menambahkan path untuk `update` pada `StudentController` untuk melakukan pencarian dari `npm` yang ingin diupdate. Apabila `npm` ditemukan maka akan ke `direct` ke `form-update.html`. Sebaliknya apabila tidak maka akan ke `direct` ke `not-found.html`. Jika `update` berhasil dilakukan maka akan ke `direct` ke halaman `success-update.html`. `Update` pada latihan kali ini langsung mengupdate data pada database.

Latihan Menggunakan Object sebagai Parameter

Pada latihan ini, parameter dari `update` diubah menjadi objek `StudentModel` dengan menggunakan `@ModelAttribute`. Dengan `@ModelAttribute`, kita tidak perlu menggunakan parameter `npm` dan membuat object `student` dalam method dan dapat langsung memanggil `updateStudent` untuk melakukan `update`.