Diferența dintre @GetMapping și @PostMapping în Spring (framework Java pentru dezvoltarea aplicațiilor web) ține de **tipul de cerere HTTP** pe care o tratează fiecare adnotare:

**🔹 @GetMapping**

* Este folosită pentru **cereri HTTP GET**.
* GET este utilizat în mod obișnuit pentru **a obține (citi)** date de la server.
* Nu modifică starea resurselor pe server (operație **idempotentă**).
* Datele se trimit prin **URL (query parameters)**.

**Exemplu:**

@GetMapping("/produse")

public List<Produs> getProduse() {

return produsService.getToateProdusele();

}

**🔹 @PostMapping**

* Este folosită pentru **cereri HTTP POST**.
* POST este utilizat pentru **a trimite date către server**, de obicei pentru **a crea o resursă nouă**.
* Modifică starea pe server (operație **non-idempotentă**).
* Datele se trimit în **body-ul cererii**, nu în URL.

**Exemplu:**

@PostMapping("/produse")

public Produs adaugaProdus(@RequestBody Produs produs) {

return produsService.salveazaProdus(produs);

}

**🔁 Pe scurt:**

| **Adnotare** | **Tip cerere HTTP** | **Folosit pentru** | **Trimitere date** |
| --- | --- | --- | --- |
| @GetMapping | GET | Citirea datelor | Prin URL (query params) |
| @PostMapping | POST | Trimiterea/crearea datelor | În body-ul cererii |

Dacă vrei, îți pot da și un exemplu complet cu controller + testare prin Postman. Spune-mi doar ce vrei să construim – poate un sistem simplu de produse, utilizatori etc.