

Documentazione helper interna

Tipi di classe

- Model

Una classe il cui nome termina per Model, per esempio UtenteModel, rappresenta il modello di un utente all'interno del sistema. Generalmente dovrebbe essere composto solo da attributi, costruttore e getter (non setter). Non setter perchè tutti i valori di un utente risiedono nel database. L'oggetto utente è istanziato tramite costruttore.

- Service

Una classe il cui nome termina per Service, per esempio UtenteService, rappresenta le funzionalità della utente. In spring boot viene annotata con `@Service` sopra la definizione della classe. Può contenere direttamente la funzionalità o richiamare funzionalità definite altrove (per la dependency injection).

- RestController

Una classe il cui nome termina per Controller, per esempio UtenteController, rappresenta le funzionalità della classe utenti richiamabili esternamente tramite richieste HTTP. L'annotazione sulla definizione della classe è `@RestController`.

L'url dell'endpoint è definito tramite `@RequestMapping("url")`, per esempio `@RequestMapping("api/Utente")`. Le funzioni in questa classe sono accessibili tramite richieste http, per dire di che tipo di richieste si usano le annotazioni:

@PostMapping per una POST

@GetMapping per una GET

@PutMapping per una PUT

Per testare il rest API vi consiglio PostMan.

Altre annotazioni utili

@Repository indica che è una classe che parla con un database.

@AutoWired gestisce la dependency injection, la useremo penso sempre solo per costruttori. In automatico fornisce al costruttore un oggetto del tipo richiesto dal costruttore. In pratica il costruttore della classe non viene "manualmente" chiamato da nessuna parte.

@Qualifier("nome") Per disambiguare tra diverse possibili classi per @AutoWired si può qualificare/dare un nome via annotazione alle classi.