**GET műveletek paraméterezése**

REST esetén az alkalmazásban található erőforrásokat az url-ek azonosítják; ki tudjuk használni az url előnyeit: url 1)paraméterek használata és az 2)url bizonyos részeinek dinamikus változtathatósága -> ezeket mind támogatja a Spring MVC

**A Controllerbe hogyan kell ezt a paraméterezhetőséget megvalósítani?**

**1 - cél: Az Url-ben át lehessen adni egy prefix értéket**

**@RequestParam** annotációval látjuk el a metódusunk aktuális paraméterét

az **Optional jellegből** fogja tudni a Spring, hogy ezt nem kötelező kitölteni; ugyanezt elérhetjük az annotációnak beállított required=false-al

az urlből Stringként kiszedett adatot átalakítja automatikusan int-é, ha a prefix paramétert intként adjuk meg



**2 - cél: URL részletek kezelése**

cél: x-edikként azonosított alkalmazott a /x-edik címen legyen

**@PathVariable**

**@Getmappingnek placeholderként kell átadni az id-t**, ahol azt deifináljuk, hogy milyen urlen legyen elérhertő a metódus; a Spring tudni fogja, hogy itt valamilyen értéket vár amit majd tovább kell adnia a metódusnak

**itt is működik a típus konverzió**



**12FACTOR hivatkozás (=hogyan fejlesszünk alkalmazásokat) hogy érvényesül a RESTFUL API-knál?**

**API -FIRST**

Először legyen meg az API leírás, és utána legyen implementálva mögé a megoldást (=contract first;kétséges)

az alkalmazások nem látnak be egymás mélyebb rétegeibe, csak interfacen keresztül – laza csatolás

**a RESTFUL API-nk alaposan dokumentált és tesztelt legyen;** eszközök és szabványok erre: API Blueprint, Swagger, OpenAPI(JSON formátumban írja le az API erőforrásait, az elvégezhető műveleteket ,formátumokat stb.)

**STATELES – PROCESSES**

**az állapot csak egy kérésnek belül létezhet** (=csak egy kérés időtartamáig van állapot); a http response után már **nem tárolhatjuk az állapotot a memóriában az újabb request bejövetelével;** ha az állapot eltűnik, lehet a felhasználó számára eltűnik az adat; de csak pl. egy termék, nem a teljes kosár

nem kell clusterezni a nodeokat – ha több gép futtatja több példányban az alkalmazásunkat, elegendő csak egy terhelés elosztást végezni, a nodeoknak nem kell az állapotaikat egymás között megosztaniuk; nem lesz párhuzamossági probléma se

**ugyanakkor szeretnénk a kéréske közötti állapotot megőrizni valahogy, de nem az alkalmazáson; erre jó a backing service**, mint pl. az adatbázisban való tárolás vagy elosztott cacche-ben(=eloszottt alkalmazásokban)

cachelni lehetne magában az alkalmazásban is, de a12factor szerint helyezzük ezt ki külön backend servicebe

ahhoz, hogy update végrehajtódjon, csak egy node kelljen

**Konkurencia**

Ha állapotmentesen dolgozunk (springben = azzal, hogy a Controller, Service, Repository osztályainkban ne legyen állapotot tároló atrit és értékeket kizárólag paraméterként adunk át)

horizontális skálázás – kiszolgáló **threadek** különböző számítógépeken

ha mégis kell/lehetséges **konkurencia kezelés**, akkor azt is backend service szintre kell vinni (vagy **adatbázis lockolat** kell használni, vagy az elosztott cachekből lehet lockokat definiálni)

**GET műveletek paraméterezése - gyakorlat**

*alkalmazás módosítása, hogy azokat listázza, amiknek a neve megegyezik a prefixxel, amit url paraméterben adunk meg*

1. meglévő listázós metódus módosítása -> @RequestParam és Optional; paraméter átadása a servicenek
2. service metódus kapja meg a prefixet és stream filterelés
3. új rest végpont megírása, ami azonosító alapján kéri le az alkalmazottat
4. @Getmapping(”/{id}”) és findEmployeeById metódus @PathVariable-vel

delegálás a service findemployeebyid() metódusának és annak kell még az implementációja

**Kérdések**

*REST esetén az alkalmazás erőforrásait mik azonosítják?*

*A Spring MVC mit támogat az urlek kapcsán amivel a műveleteket paraméterezni tudjuk?*

*Hogy adunk át az URLben prefix értéket? Hogy tesszük opcionálissá a használatát?*

*Hogy adunk át az urlben placeholderként változót\_? Jellemzően mit adunk így át?*

*Hogy működik ezeknél az átadásoknál a típuskonverzió?*

*A 12Factor hivatkozásban az API FIRST elv hogyan valósul meg REST alkalmazásokban?*

*A 12Factor hivatkozásban a STATELESS – PROCESSES elv hogyan valósul meg REST alkalmazásokban?*