

## 021PROTD2265613 PROGRAMMAZIONE JAVA

### VERIFICA DELLE COMPETENZE ACQUISITE

Il progetto dovrà avere queste caratteristiche:

**## SPRING INITIALIZER** : Sostituire *YOURLASTNAME* col proprio nome

Spring version: 2.7.8

name: *YOURLASTNAME*

artifact: *YOURLASTNAME*

group: it.cgmconsulting

package: it.cgmconsulting.*YOURLASTNAME*

### ## DATABASE MYSQL

Nel file application.yaml dare al database il nome ***video\_store\_exam***

### ## SPRING SECURITY **NON** E' DA IMPLEMENTARE

### ## APPLICAZIONE: I CONTROLLER DA CREARE:

1. `@PatchMapping("/update-film/{filmId}") ---> @RequestBody FilmRequest`
2. `@GetMapping("/find-film-in-store/{filmId}") ---> List<FilmStoreResponse>:  
film_id, title, store_name`
3. `@GetMapping("/find-films-by-language/{languageId}") --> List<FilmResponse>  
film_id, title, description, release_year, language_name`
4. `@PutMapping("/add-film-to-store/{storeId}/{filmId}")`
5. `@GetMapping("/count-customers-by-store/{storeName}") -->  
CustomerStoreResponse: store_name, total_customers (N.B. sono da considerare 'clienti' di un determinato store tutti quelli che hanno effettuato almeno un noleggio).`
6. `@PutMapping("/add-update-rental") add: inserimento rental; update:  
aggiornamento solo della data restituzione (rental_return). In questo`

caso non vi sono suggerimenti sui parametri da passare al controller. **N.B.** Il film è noleggiabile solo se almeno una copia è presente in uno store e non risulta a sua volta già noleggiata.

7. @GetMapping("/count-rentals-in-date-range-by-store/{storeId}")  
@RequestParam Date start, Date end --> return: conteggio dei noleggi in un determinato store in un determinato arco temporale (comprende anche i film non ancora restituiti).
8. @GetMapping("/find-all-films-rent-by-one-customer/{customerId}") return: List<FilmRentResponse> : film\_id, title, store\_name (considerare anche i film non ancora restituiti).
9. @GetMapping("/find-film-with-max-number-of-rent") Trovare il film o i film con il maggior numero di noleggi (anche se non ancora restituiti).  
List<FilmMaxRentResponse> : film\_id, title, numero di noleggi totale.
10. @GetMapping("/find-films-by-actors") ---> @RequestParam: Collection di staff\_id di attori --> return List<FilmResponse>: questa lista dovrà contenere i film a cui hanno lavorato tutti gli attori i cui identificativi sono stati passati come parametro. La lista dovrà essere ordinata alfabeticamente per titolo del film.