EX - Attrici e Attori

Consegna

Repo: ex-attrici-e-attori-ts

Nota: a differenza di quanto visto finora negli esempi, per accedere all'API utilizzare l'url base:
http://localhost:3333
al posto di:
https://freetestapi.com/api/v1
Ad esempio:
http://localhost:3333/users
per chiamare l'endpoint /users
Clicca qui per la guida su come installare il Server API Locale!

📌 Milestone 1

Crea un **type alias `Person`** per rappresentare una persona generica.
Il tipo deve includere le seguenti proprietà:

- `id`: numero identificativo, **non modificabile**
- `name`: nome completo, stringa **non modificabile**
- `birth_year`: anno di nascita, numero
- `death_year`: anno di morte, numero **opzionale**
- `biography`: breve biografia, stringa
- `image`: URL dell'immagine, stringa

📌 Milestone 2

Crea un **type alias `Actress`** che oltre a tutte le proprietà di `Person`, aggiunge le seguenti proprietà:

- `most_famous_movies`: una tuple di **3 stringhe**
- `awards`: una stringa
- `nationality`: una stringa tra un insieme definito di valori.
  Le nazionalità accettate sono: *American, British, Australian, Israeli-American, South African, French, Indian, Israeli, Spanish, South Korean, Chinese*.

📌 Milestone 3

Crea una funzione **`getActress`** che, dato un `id`, effettua una chiamata a:

```
GET /actresses/:id
```

La funzione deve restituire l'oggetto Actress, se esiste, oppure `null` se non trovato.
Utilizza un **type guard** chiamato **`isActress`** per assicurarti che la struttura del dato ricevuto sia corretta.

📌 Milestone 4

Crea una funzione **`getAllActresses`** che chiama:

```
GET /actresses
```

La funzione deve restituire un array di oggetti `Actress`.
Può essere anche un array vuoto.

📌 Milestone 5

Crea una funzione **`getActresses`** che riceve un array di numeri (gli id delle attrici).
Per ogni id nell'array, **usa la funzione `getActress`** che hai creato nella Milestone 3 per recuperare l'attrice corrispondente.
L'obiettivo è ottenere una lista di risultati **in parallelo**, quindi dovrai usare `Promise.all`.
La funzione deve restituire un array contenente elementi di tipo `Actress` oppure `null` (se l'attrice non è stata trovata).

🎯 BONUS 1

Crea le funzioni:

- **`createActress`**
- **`updateActress`**

Utilizza gli **Utility Types**:

- `Omit`: per creare un'attrice senza passare `id`, che verrà generato casualmente.
- `Partial`: per permettere l'aggiornamento di qualsiasi proprietà tranne `id` e `name`.

🎯 BONUS 2

Crea un tipo `Actor`, che estende `Person` con le seguenti differenze rispetto ad `Actress`:

- `known_for`: una tuple di **3 stringhe**
- `awards`: array di una o due stringhe
- `nationality`: le stesse di `Actress` più:
  Scottish, *New Zealand, Hong Kong, German, Canadian, Irish*.

Implementa anche le versioni `getActor`, `getAllActors`, `getActors`, `createActor`, `updateActor`.

🎯 BONUS 3

Crea la funzione **`createRandomCouple`** che usa `getAllActresses` e `getAllActors` per restituire un'array che ha **sempre due elementi**: al primo posto una `Actress` casuale e al secondo posto un `Actor` casuale.