PW 2019 – Sebastien Pittion – TP 3 : Finalisation de l'application

Aide (Installation)

Terminer d'abord les TP 1 et 2.

S'assurer que le code est bien présent sur un dépôt Git distant.

Il est grand temps de s'occuper du bouton « Random » introduit au TP1. Il s'agit d'appeler une méthode **asynchrone** nommée **getRandomMovie** (ne prenant aucun argument) lors du clic sur le bouton en question. Cette méthode devra appeler une autre méthode (à coder, appelant elle-même la méthode **askTMDB** - introduite au TP2 - se basant sur l'API discover pour récupérer des films selon les critères de votre choix) dont la signature est la suivante :

```
private discoverMovies(): Promise<Movie[]>;
```

L'implémentation de **getRandomMovie** devra piocher un film au hasard parmi ceux retournés par **disoverMovies** et présenter une alerte Ionic possédant les caractéristiques suivantes :

- Son **header** contiendra le **title** du film pioché.
- Son **message** contiendra l'**overview** du film pioché.
- Un bouton « Cancel », sans **handler**, permettra de simplement fermer l'alerte.
- Un bouton « Show details », via son **handler**, permettra d'appeler ma méthode **showDetails** - introduite au TP1 - en passant en paramètre le film précédemment pioché au hasard.

A ce stade, vous pourrez alors naviguer vers les détails d'un film récupéré au hasard en cliquant sur le bouton « Random » situé dans le header de votre application.

Ensuite, en vous inspirant de la manière dont vous avez utilisé l'alerte Ionic, modifier la méthode **askTMDB** pour :

- Présenter un loader Ionic avant que l'appel à la méthode **get** du **HttpClient** soit fait.
- Fermer (**dismiss**) le loader en question avant de retourner les résultats de l'appel à cette méthode **get**.

Attention : bien attendre (**await**) aux bons endroits, comme dans les exemples de code de la documentation.

Ainsi, chaque fois que vous interrogerez l'API TMDB, un « écran » de chargement sera affiché pour bien se rendre compte qu'une requête HTTP est faite à l'arrière-plan.

Enfin, dans le fichier TypeScript de la page **details**, rajouter la méthode suivante :

```
getImageUrl(image: string): string {
  return `https://image.tmdb.org/t/p/w780${image}`;
}
```

Puis, en combinant l'exemple du cours et un appel à **getImageUrl**, afficher dans les détails les **backdrop** et **poster** du film sélectionné en complément de son **title**, **overview** et de sa **release_date** (date pipe).