academeez

@ngrx/entity

Lists in our state

@ngrx/entity

- When you need to hold a list of items in your state there will probably be common repeating actions you would like to perform on that list
    - Set the list
    - Add item
    - Delete item
    - Update item
    - etc.
- If you have more than one list you will notice a lot of repeating code in the action, reducers of those lists.
- @ngrx/entity helps you deal with lists in your state while avoiding the need to duplicate your code

EntityAdapter

academeez

- Contains common methods for dealing with collection in the state
- You create an adapter for your collection using ht method **createEntityAdapter**
- You will get an object with a bunch of common methods you can perform in the reducer on your collection

```
export interface EntityStateAdapter<T> {
  addOne<S extends EntityState<T>>(entity: T, state: S): S;
  addMany<S extends EntityState<T>>(entities: T[], state: S): S;
  setAll<S extends EntityState<T>>(entities: T[], state: S): S;
  removeOne<S extends EntityState<T>>(key: string, state: S): S;
  removeMany<S extends EntityState<T>>(keys: string[], state: S): S;
}
```

@ngrx/entity - ex

- We have a list of todo items that we want to grab from a server and place in the state
  - https://nztodo.herokuapp.com/api/task/?format=json
- Manage that list using @ngrx/entity

Adapter

- The adapter contains methods to manage the list in the state
- We use those methods in the reducer in response to certain actions we define
- The adapter can also help us create the initial state using **getInitialState()**

Summary

academeez

- @ngrx/entity provides us with adapter which helps us perform common actions on a collection
- It will help us avoid repeating code and manage the collection in our state

academeez

Thank You

Next Lesson: @ngrx/data