Enhance Web Development

Angular 2

Reactive State avec @ngrx/store

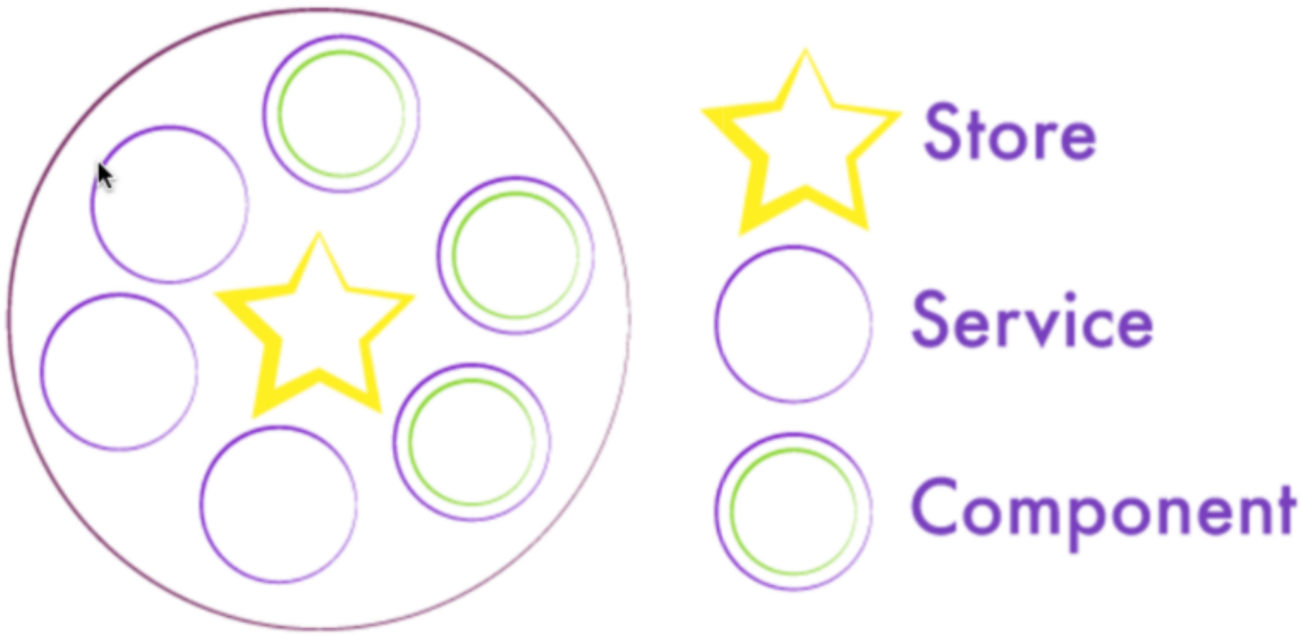
- Redux et @ngrx/store
- Store
- Reducers
- provideStore
- Actions
- store.select
- store.dispatch

Redux et @ngrx/store

- RxJS donne un gestionnaire de state pour les application Angular 2 inspiré de Redux
- @ngrx/store conserve les mêmes principes que redux
- On peut "subscribe" à nos état et donc utiliser | async et les opérations observables

Store

- Le store peut être considéré comme la base de donnée de l'application
- Toutes les manipulations d'état se fait dans le reducer qui est enregistré sur le store
- Il prend un reducers et opère un observable pour résulter un nouveau state
- Les store peuvent mettre en place des pre & post reducers



Un arbre unique d'état

Un arbre unique d'état

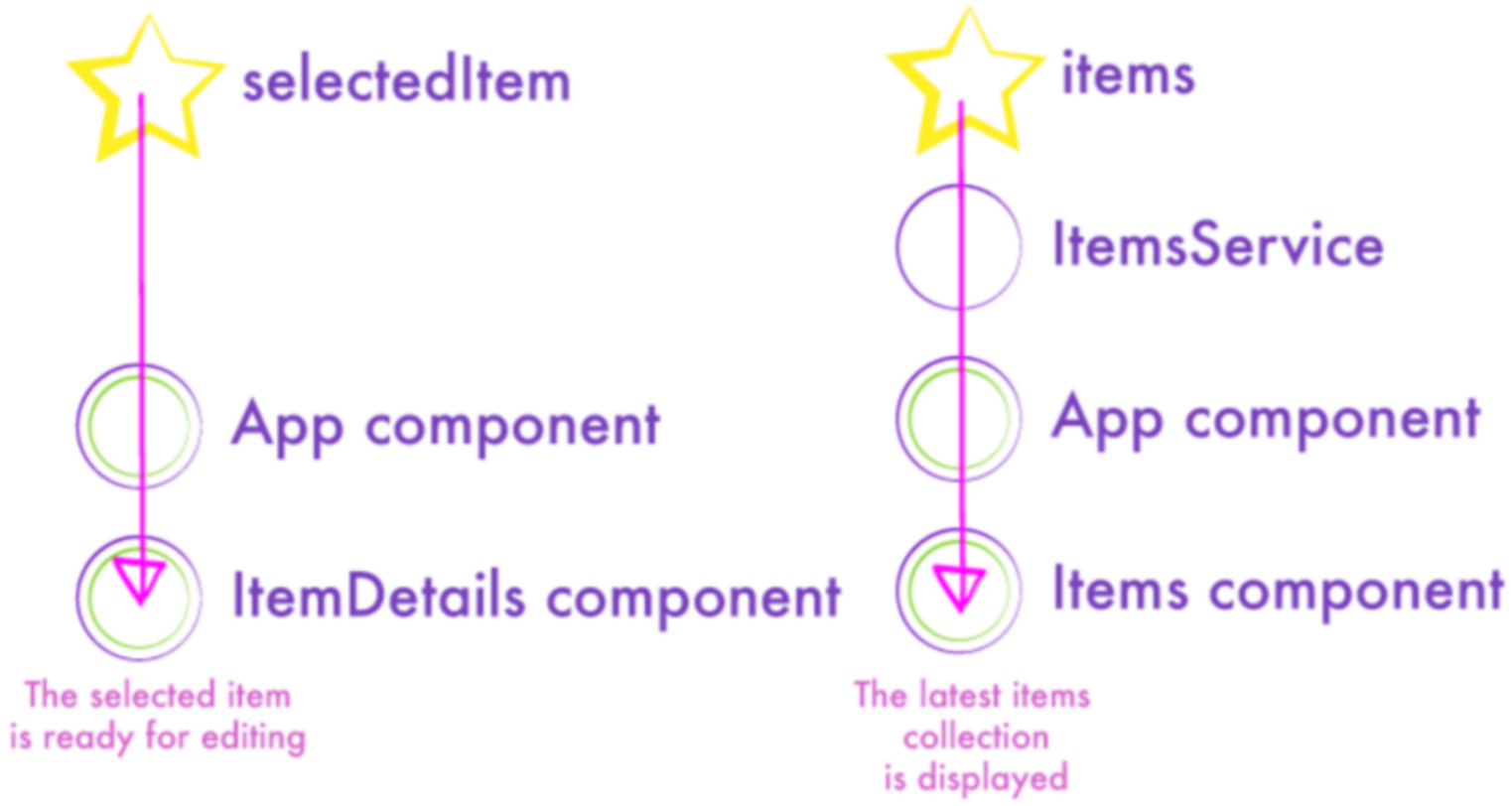
```
export interface Item {   id: number;
  name: string;
  description: string;
};
export interface AppStore {
  items: Item[];
  selectedItem: Item;
};
```

Un arbre unique d'état

Reducers

- Une méthode qui prend en paramètre le state courant et une action
- Retourne le nouveau state basé sur l'action
- Pure functions, pure functions, pure functions !

Les states sont propagés vers le bas



Les states sont propagés vers le bas

```
export const selectedItem = (state: any = null, {type, payload}) => {
  switch (type) {
    case 'SELECT_ITEM':
      return payload;
    default:
      return state;
  }
}
```

Reducer

provideStore

- Rend un reducer disponible dans notre application Angular 2

```
import {App} from './src/app';
import {provideStore} from '@ngrx/store';
import {items} from './src/common/stores/items.store';
import {selectedItem} from './src/common/stores/selectedItem.store';

bootstrap(App, [
  provideStore({items, selectedItem})
]);
```

provideStore

store.select

- Retourne un Observable du type que l'on a stocké
- On peut utiliser **combineLatest** pour aggréger plusieurs store

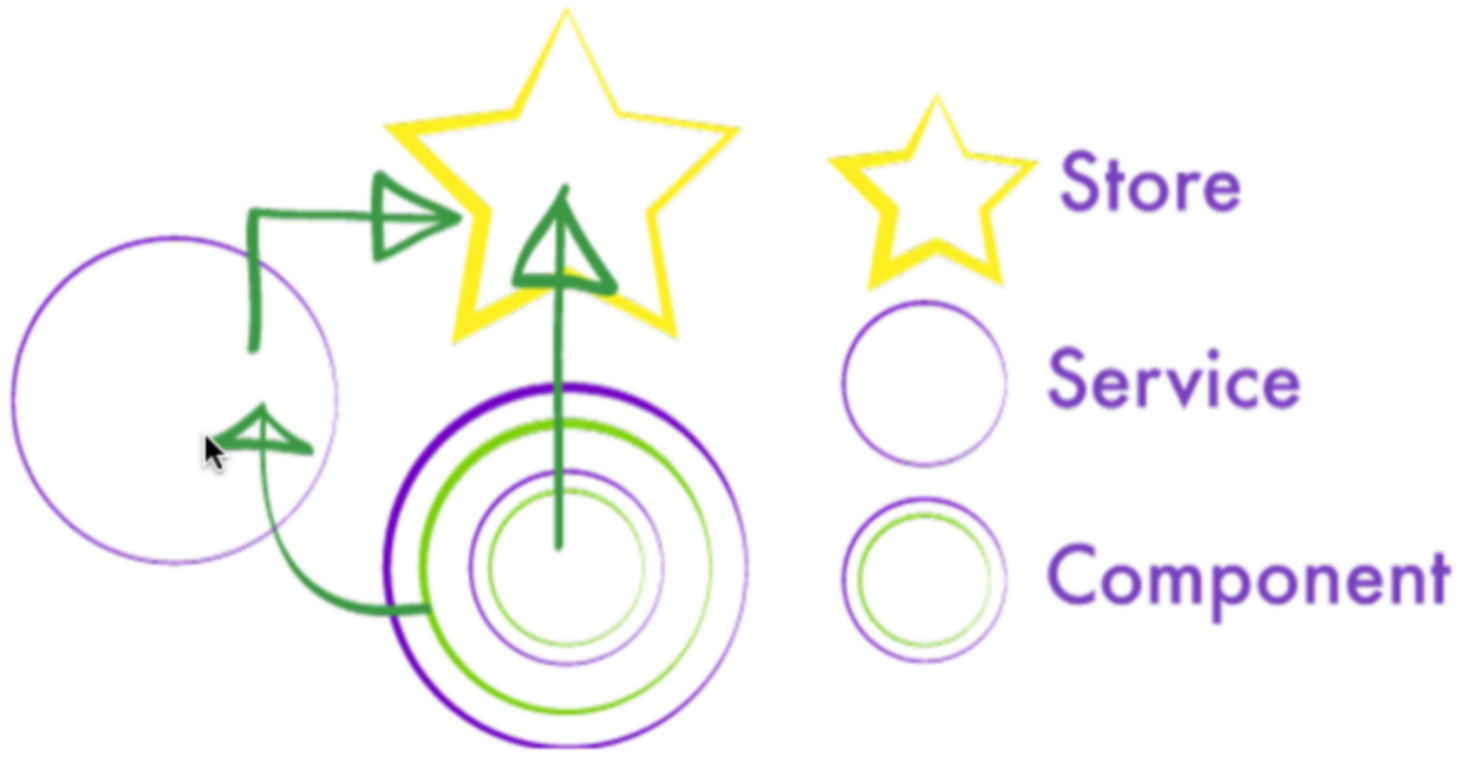
```
// items component
this.selectedItem = store.select('selectedItem')

// items template
<item-detail
(saved)="saveItem($event)"
(cancelled)="resetItem($event)"
[item]="selectedItem | async">Select an Item</item-detail>
```

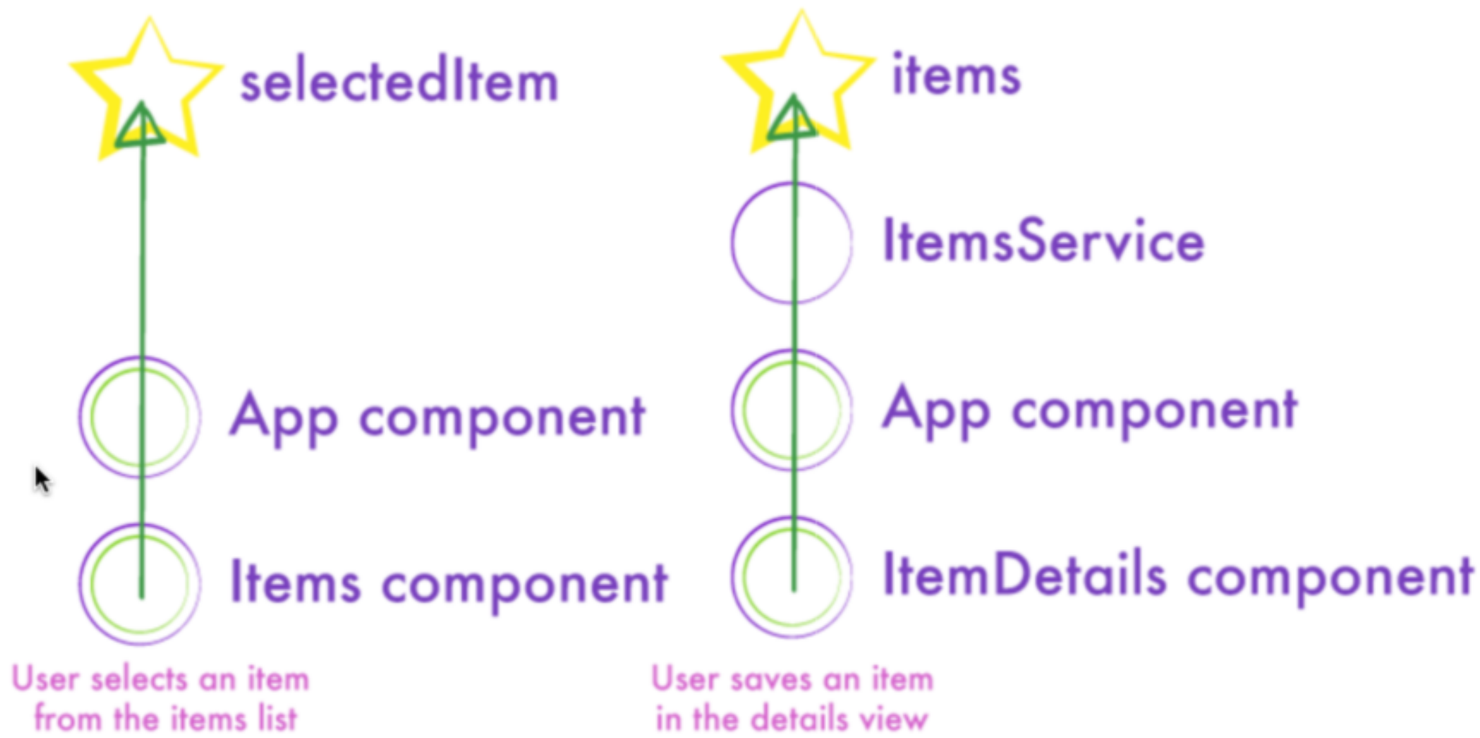
store.select

Actions

- Généralement ce sont les services dans une application Angular 2 qui vont dispatch vers le reducer/store
- Type + Payload?
- Basé sur le type, le reducer va effectuer l'actions avec la payload et retourner un nouveau state



Flux d'évènement ascendant



Flux d'évènement ascendant

store.dispatch

- Envoi une action au store, qui appelle le reducer associé pour mettre à jours le state
- Appelé depuis un service (ou un composant)

```
selectItem(item: Item) {
  this.store.dispatch({type: 'SELECT_ITEM', payload: item})
}
```

store.dispatch