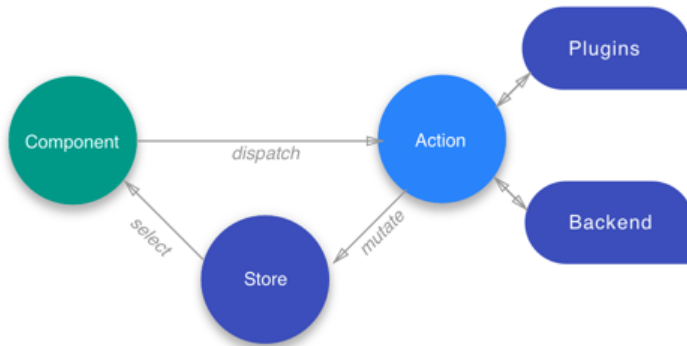


# NgXS

## Installation

```
npm install @ngxs/store
```

## Concept



## Création de l'état

```
import { State } from '@ngxs/store';
import { Injectable } from '@angular/core';
import { tap } from 'rxjs/operators';

export interface AppModel {
  title: string
}

@State<AppModel>({
  name: 'app',
  defaults: {
    title: '
  }
})
@Injectable()
export class AppState {

}
```

## Ajout dans le module

```
NgxsModule.forRoot([ AppState ])
```

## Création d'une action

```
export class GetTitle {
    static readonly type = '[App] Get Title'
    constructor(public title: string) {}
}
```

Ensuite, nous déclenchons une méthode dans le state lors que l'action est démarrée. Les paramètres du constructeur sont les données à envoyer lors de l'action

## Selector

Le selector récupère une partie de l'état. Nous mettons dans la classe AppState

```
@Selector()
static getTitle(state: AppModel) {
    return state.title
}
```

## Afficher dans le composant

```
@Select(AppState.getTitle) title: Observable<string>
```

## Effectuer une action

Tout d'abord, nous définissons la méthode de l'action dans AppState

```
@Action(GetTitle)
setTitle({ getState, setState }: StateContext<AppModel>, action: GetTitle) {
    const state = getState()
    setState({
        ...state,
        title: action.title
    })
}
```

Nous pouvons effectuer le même code dans une Observable pour une action asynchrone

## Déclencher l'action

Dans le composant:

1. Injection du Store

```
constructor(
```

```
    private store: Store  
  ) { }
```

## 2. L'action

```
this.store.dispatch(new GetTitle({ title: 'New 'App' })))
```