

```
// pokemons.service.ts

async getPokemons(offset: number, limit: number): Promise<PokemonDTO[]> {
  const cachedData = await cache.getItem(` ${cacheName}-${args.join('-')}`);

  if (cachedData) {
    return cachedData;
  }

  // requests and data treatments
  // ...

  return result;
}
```

CAS PRATIQUE

Implémentation d'une annotation

```
// pokemons.service.ts

@cached("pokemons")
async getPokemons(offset: number, limit: number): Promise<PokemonDTO[]> {}
```



```
// cached.decorator.ts
// ....  
  
export function cached(cacheName: string) {
  return function actualDecorator(originalMethod: any, context: ClassMethodDecoratorContext) {
    async function replacementMethod(this: any, ...args: any[]) {
      const cachedData = await cache.getItem(` ${cacheName}-${args.join('-')}`);
      if (cachedData) {
        return cachedData;
      }
      const result = originalMethod.call(this, ...args);
      await cache.setItem(` ${cacheName}-${args.join('-')}`, result, { ttl: 86400 });
      return result;
    }
    return replacementMethod;
  }
}
```