

Université Toulouse Paul Sabatier - KEAT9AA1

Guilhem Saurel

2025-09-25

Dans un nouveau dossier:

```
git clone git@github.com:votre-nom/votre-dépôt.git
```

```
$ uv add --dev ipython
```

```
$ ./manage.py shell
```

```
In [1]: from high_level.models import Localisation
```

```
In [2]: l = Localisation.objects.first()
```

```
In [3]: l.nom
```

```
Out[3]: "Toulouse"
```

```
In [5]: l.local_set.get(nom="TLS-01").surface
```

```
Out[5]: 209
```

```
# high_level/tests.py
from django.test import TestCase

from .models import Machine

class MachineModelTests(TestCase):
    def test_machine_creation(self):
        self.assertEqual(Machine.objects.count(), 0)
        Machine.objects.create(nom="Mélangeur", prix=28_000, ...)
        self.assertEqual(Machine.objects.count(), 1)

$ ./manage.py test
```

Implémentez une méthode « `def costs(self):` » dans chaque modèle où ça a un sens :

- > `DebitEnergie`
- > `ApprovisionnementMatierePremiere`
- > `Local`
- > `RessourceHumaine`
- > `Machine`

Implémentez un scénario de test qui valide le calcul des coûts dans un cas connu. Par exemple:

- > un Local de 50 m²
- > dans la Localisation Labège à 2 000 €/m²
- > avec une Machine à 1 000 €, et une autre à 2 000 €
- > et en stock
 - 1000 kg de sucre à 10 €/kg
 - 50 m³ d'eau à 15 €/m³

On s'attend à ce que `Local.objects.first().costs()` vaille
110 750 €