API

Créer une entité

Paramètres d'URL

Nom	Indication
unflatten	Reconstruire des structures ("xx__yy__zz": 1 donnera {'xx': {'yy': {'zz': 1}}})

```
POST /api/entity/<slug>/?unflatten=1 HTTP/1.1
Content-Type: application/json

{
  "champ1": <valeur1>,
  "champ2": <valeur2>,
  "journal__backoffice_url": "<journa_valeur2>,
}
```

Cas passant

L'entité est créée, un identifiant lui est attribué et est renvoyé dans le champ id, les attributs journal__ sont stockées comme une entrée de journal associée à l'entité enlevant le préfixe journal__.

```
201 Created
Content-Type: application/json

{
   "err": 0,
   "id": 1234,
   "data": {
       "champ1": <valeur1>,
   }
}
```

Cas non-passant

Slug d'entité inexistant

```
404 Not found
Content-Type: application/json

{"err": 1}
```

Schéma invalide

```
404 Not found
Content-Type: application/json

{"err": 1, "errors": ["clé x manquant", "clé y invalide"]}
```

Modifier une entité

On peut soit écraser le contenu actuel avec PUT, soit mettre à jour l'existant avec PATCH. Comme pour la création les clés avec le préfixe journal__ servent à alimenter l'historique et le paramètre unflatten indique de renormaliser les données.

Requête

```
PUT /api/entity/1234/?unflatten=1 HTTP/1.1
Content-Type: application/json
```

```
{
   "champ1__0__champ2": 1,
   "champ1__0__champ3": 2,
   "journal__backofice_url": "https://..."
}
```

Cas passant

```
200 Ok
Content-Type: application/json

{
   "err": 0,
   "data": {
      "champ1": [
         {
            "champ2": 1,
            "champ3": 2
         }
      ]
   }
}
```

Cas non-passant

- slug d'entité inexistant
- schéma invalide (voir plus haut)