

[Pro](#) [Équipes](#) [Tarification](#) [Documentation](#)**npm**[Inscription](#)**Recherche**

mongoose-unique-validateur DT

4.0.0 - Publié il y a 5 mois

[Code](#) Bêta[3 Dépendances](#)[192 personnes à charge](#)[25 Versions](#)

mongoose-unique-validateur

build unknown

le monogone-unique-validateur est un plugin qui ajoute la validation pré-sauvegardée pour des champs uniques dans un schéma Mongoose.

Cela rend la gestion des erreurs beaucoup plus facile, puisque vous obtiendrez une erreur de validation Mongoose lorsque vous tenterez de violer une **une contrainte unique**, plutôt qu'une erreur E11000 de MongoDB.

Utilisation

Fils: `yarn add mongoose-unique-validator`

MNP: `npm install --save mongoose-unique-validator`

Appliquez ensuite le plugin à votre schéma :

```
var mongoose - nécessité('mongoose');
```

```
var uniqueValidator « require » («mongoose-unique-validator») );

var mySchema .Schema(/- mettre votre définition de schéma ici));
.plugin uniqueValidator));
```

Exemple

Disons que vous avez un schéma d'utilisateurs. Vous pouvez facilement ajouter la validation pour les contraintes uniques dans ce schéma en appliquant la uniqueValidator plugin à votre schéma utilisateur:

```
var mongoose - nécessité('mongoose'));
var uniqueValidator « require » («mongoose-unique-validator») );

// Définissez votre schéma comme normal.
var userSchema et mongoose.Schema({
  nom d'utilisateur : . type type: String, requis : true, unique
  e-mail : {type : String, index : true, unique : true, require
  Mot de passe : {type : Chaîne, requise : true}
-));

// Appliquer le plugin uniqueValidator à userSchema.
userSchema.plugin uniqueValidator));
```

Maintenant, quand vous essayez de sauvegarder un utilisateur, le validateur unique vérifiera les entrées de la base de données en double et les signalera juste comme toute autre erreur de validation:

```
userutilisateur var et nouvel utilisateur ({nom d'utilisateur : '
user.save(fonction (err){
  console.log(err));
});
```

-

```

message: "Validation échouée",
Nom: "ValidationError",
Erreurs: {
  Nom d'utilisateur : {
    message: "Erreur, attendu 'nom d'utilisateur' |
    Nom : 'ValidatorError',
    gentil : "unique",
    chemin: "nom d'utilisateur",
    Valeur : 'JohnSmith'
  }
}

```

Trouver et mettre à jour

En cas d'utilisation `findOneAndUpdate` et les méthodes connexes, la mangose ne fonctionne pas automatiquement en validation. Pour déclencher cela, vous devez passer un objet de configuration. Pour des raisons techniques, ce plugin exige que vous définissiez également le contexte option à `query`.

```
{ runValidators: true, context: 'query' }
```

Un exemple complet:

```

User.findOneAndUpdate(
  { - e-mail : 'old-email-example.com' },
  { - e-mail : 'new-email-example.com' },
  - { runValidators: vrai, contexte: 'query' },
  fonction(err){
    // ...
  }
)

```

Types d'erreurs personnalisés

Vous pouvez passer par un type d'erreur personnalisé dans le cadre de l'option `options argument`:

```
userSchema.plugin(uniqueValidator, {et type: 'mongoose-unique-val:
```

Après avoir exécuté l'exemple ci-dessus, la sortie sera:

```
-
  message: "Validation échouée",,
  Nom: "ValidationError",,
  Erreurs:{
    Nom d'utilisateur :{
      message: "Erreur, attendu "nom d'utilisateur" |
      Nom : 'ValidatorError',
      nature : "mongoose-unique-validateur",,
      chemin: "nom d'utilisateur",,
      Valeur : 'JohnSmith'
    }
  }
-
-
-
```

Vous pouvez également spécifier un type d'erreur personnalisé par défaut en supplantant le plugin `defaults.type` variable:

```
uniqueValidator.defaults.type - 'mongoose-unique-validator'
```

Messages d'erreur personnalisés

Vous pouvez passer à travers un message d'erreur personnalisé dans le cadre de l'option `options argument`:

```
userSchema.plugin(uniqueValidator, {message: 'Error, 'PATH
```

Vous avez accès à toutes les étalons d'erreur Mongoose standard:

- {PATH}
- {VALUE}
- {TYPE}

Vous pouvez également spécifier un message d'erreur personnalisé par défaut en remplaçant le plugin `defaults.message` variable:

```
uniqueValidator.defaults.message - 'Error, 'PATH' s'attendait à ce
```

Cas insensible

Pour les correspondances insensibles au cas cas, inclure la `uniqueCaseInsensitive` option dans votre schéma. Les questions traiteront `john.smith@gmail.com` et `John.Smith@gmail.com` en tant que doublons.

```
var userSchema = mongoose.Schema({
  nom d'utilisateur : { type: String, requis : true, unique: true },
  e-mail : { type: Corde, index : true, unique : true, requis : true },
  Mot de passe : { type: Chaîne, requise : true }
});
```

Conditions supplémentaires

Pour des conditions de contrainte uniques supplémentaires (ex: n'imposer que des contraintes uniques sur les enregistrements non mustitués), l'option MongoDB `partialFilterExpression` peuvent être utilisés.

Remarque: l'option `index` doit être passé en tant qu'objet contenant `unique: true`, ou bien `partialFilterExpression` seront ignorés.

```
var userSchema = mongoose.Schema({
  nom d'utilisateur : { type: String, requis : true, unique: true },
  e-mail : {
    Type: Chaîne,
    requis : true,
    partialFilterExpression: { nom d'utilisateur: 'John' }
  }
});
```

```
    Indice:{
      unique : vrai,
      partial FiltreExpression : supprimé {deleted: false}
    },
    Mot de passe : {type : Chaîne, requise : true}
  -));
```

Réveil

Parce que nous nous appuyons sur des opérations asynchroniques pour vérifier si un document existe dans la base de données, il est possible pour deux requêtes d'exécuter en même temps, les deux obtiennent 0, puis les deux insèrent dans MongoDB.

En dehors du verrouillage automatique de la collection ou de forcer une seule connexion, il n'y a pas de solution réelle.

Pour la plupart de nos utilisateurs, ce ne sera pas un problème, mais c'est un cas de bord que d'être conscient.

Mots clefs

la mangouste unique validateur

Install

```
> npm i mongoose-unique-validator
```

Repository

📁 github.com/blakehaswell/mongoose-unique-validateur

➤–Essayez RunKit

⬇ Téléchargements hebdomadaires

7 368



Page d'accueil

github.com/blakehaswell/mongoose-unique-validator-readme

Version

4.0.0

Licence

MIT

Taille non emballée

15,5 kB

Fiches totales

4

Questions

15

Demandes de tirage

2

Dernière publication

Il y a 5 mois

Collaborateurs

[🚩 Signaler les logiciels malveillants](#)

Appui

[Aide](#)

[Avis](#)

[État d'avancement](#)

[Contact npm](#)

Entreprise

[À propos](#)

[Blog](#)

[Presse](#)

Termes et politiques

[Politiques](#)

[Conditions d'utilisation](#)

[Code de conduite](#)

[Confidentialité](#)