|  |  |
| --- | --- |
| TOPIC/TITLE  level 2 DB encryption | |
| Keywords/Questions  Encryption  Mongoose-encryption  Method used  UserSchema as a plugin  Changing option  How does it work | Notes : |
|  |
| Cryptii.com |
|  |
| https://www.npmjs.com/package/mongoose-encryption |
|  |
| Secret String Instead of Two Keys |
|  |
| const encrypt = require('mongoose-encryption'); |
| We have to modify our schema |
| const userSchema = new mongoose.Schema (){  email : String,  password : String  }); |
|  |
| Then create a const to use it at a decryption key |
| const secret = "Thisisourlittlesecret." |
|  |
| userSchema.plugin(encrypt, { secret: secret }); |
| It has to be placed before your mongoose model  Plugins are a tool for reusing logic in multiple schemas. Suppose you have several models in your database and want to add a loadedAt property to each one. Just create a plugin once and apply it to each Schema: |
| https://mongoosejs.com/docs/plugins.html |
|  |
| Right now all our fields are encrypted and we only the password to be encrypted. IN order to change that we have to change some options to only encrypt cetain fields |
| userSchema.plugin(encrypt, { secret: secret, encryptedFields: ['password'] }); |
| You can add different field simply like this |
| userSchema.plugin(encrypt, { secret: secret, encryptedFields: ['password',’email’] }); |
|  |
| It will encrypt when you call save and it will decrypt when you call find. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |