It is absolutely possible to define a key in a Mongoose schema to be of type mongoose.Schema.Types.ObjectId, even if the object data is not yet in the database and comes from the client. Here's how it works:

1. Pre-defining the key type:

When you define your Mongoose schema, you can specify the type of each key using the Schema.Types object. For example:

JavaScript

```
const mongoose = require('mongoose');

const ProductSchema = new mongoose.Schema({
  _id: { type: mongoose.Schema.Types.ObjectId, required: true },
  // other schema properties
});
```

Here, the _id key is explicitly defined as a mongoose.Schema.Types.ObjectId. This tells Mongoose that this key will hold a unique identifier for each document in the collection.

2. Client-side object data:

When you create a new document from client-side data, you don't necessarily need to provide the value for the _id key. Mongoose will automatically generate a unique ObjectId for you when you save the document.

Here's an example:

JavaScript

```
const Product = mongoose.model('Product', ProductSchema);

const newProduct = new Product({
  name: 'T-shirt',
  price: 20,
  // ... other product information
});
```

In this example, the client only provides data for the name and price keys. Mongoose will

automatically generate an _id for the new product when it is saved.

3. Saving the document:

When you call the save() method on the document, Mongoose will perform the following steps:

- Validate the document data against the schema definition.
- Generate an ObjectId and assign it to the _id key if it is not explicitly provided.
- Insert the document into the database collection.

Benefits of pre-defining the key type:

There are several benefits to pre-defining the key type as mongoose.Schema.Types.ObjectId:

- **Data consistency:** It ensures that the _id key always holds a valid ObjectId, which is crucial for ensuring data integrity within your application.
- **Type safety:** It prevents accidental assignment of non-ObjectId values to the _id key, which could lead to errors.
- **Automatic generation:** It simplifies client-side code by removing the need to explicitly generate and provide an ObjectId.

In summary:

Even if the object data comes from the client, it is perfectly valid and recommended practice to pre-define the key type as mongoose.Schema.Types.ObjectId in your Mongoose schema. This ensures data consistency, type safety, and simplifies client-side code. Mongoose will automatically handle generating and assigning an ObjectId when you save the document, making it a seamless and efficient process.

Sources

1. https://github.com/YoussefMo7amed/learning