

Mongoose Population

Jogesh K. Muppala



THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

MongoDB and Relations

- NoSQL databases like MongoDB do not explicitly support relations like the SQL databases
- All documents are normally expected to be self-contained
- However you can store references to other documents within a document by using ObjectIds
- Mongoose does not have joins

Referencing Documents Example

- Consider the dishes documents that used in the exercise:
 - Comments are posted by users
 - We already have a user document containing user's information
 - Referencing the user document within the comment sub-document using ObjectIds

Modified Comment Schema

```
var commentSchema = new Schema({  
  rating: { type: Number, min: 1, max: 5, required: true },  
  comment: { type: String, required: true },  
  postedBy: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: 'User'  
  }  
}, { timestamps: true});
```

Mongoose Population

- Population is the process of automatically replacing specified paths within a document with documents from another collection
- This is where the cross-reference with ObjectIds helps

Populating the Documents

```
Dishes.find({})  
  .populate('comments.postedBy')  
  .exec(function (err, dish) {  
    if (err) throw err;  
    res.json(dish);  
  });
```

Exercise: Mongoose Populations

- Extending the User schema to track more information about the users
- Using Mongoose population to cross-reference users within a comment
- Populating the comments with the users' information upon querying