

# TUTORIAL

Build a simple app in minutes.  
Learn advanced concepts and techniques.

[Home \(/\)](#) > [Tutorials \(/tutorials\)](#) > [Todo App \(/tutorials/blaze/creating-an-app\)](#) > Collections

[← BACK TO MAIN \(/TUTORIALS\)](#)

## Storing tasks in a collection

Collections are Meteor's way of storing persistent data. The special thing about collections in Meteor is that they can be accessed from both the server and the client, making it easy to write view logic without having to write a lot of server code. They also update themselves automatically, so a view component backed by a collection will automatically display the most up-to-date data.

You can read more about collections in the Collections article (<http://guide.meteor.com/collections.html>) of the Meteor Guide.

Creating a new collection is as easy as calling `MyCollection = new Mongo.Collection("my-collection");` in your JavaScript. On the server, this sets up a MongoDB collection called `my-collection`; on the client, this creates a cache connected to the server collection. We'll learn more about the client/server divide in step 12, but for now we can write our code with the assumption that the entire database is present on the client.

To create the collection, we define a new `tasks` module that creates a Mongo collection and exports it:

```
3.1 Create tasks collection  
imports/api/tasks.js » (https://github.com/meteor/simple-todos/commit/bbe752d39cf04a51376c1cd089b053615cf08df9)
```

```
1 import { Mongo } from 'meteor/mongo';  
2  
3 export const Tasks = new Mongo.Collection('tasks');
```

Notice that we place this file in a new `imports/api` directory. This is a sensible place to store API-related files for the application. We will start by putting "collections" here and later we will add "publications" that read from them and "methods" that write to them. You can read more about how to structure your code in the Application Structure article (<http://guide.meteor.com/structure.html>) of the Meteor Guide.

We need to import that module on the server (this creates the MongoDB collection and sets up the plumbing to get the data to the client):

```
3.2 Load tasks collection on the server  
server/main.js » (https://github.com/meteor/simple-todos/commit/85ec00cdd88b93b68d2f3c517060594a8f015062)
```

```
import '../imports/api/tasks.js';
```

Let's update our client-side JavaScript code to get our tasks from a collection instead of a static array:

```
3.3 Load tasks from Tasks collection  
imports/ui/body.js » (https://github.com/meteor/simple-todos/commit/f6b008980d6f04b93e05bf777bdb0378308f4b57)
```

```
1 import { Template } from 'meteor/templating';  
2  
3 import { Tasks } from '../api/tasks.js';  
4  
5 import './body.html';  
6  
7 Template.body.helpers({  
8   tasks() {  
9     return Tasks.find({});  
10  },  
11 });
```

When you make these changes to the code, you'll notice that the tasks that used to be in the todo list have disappeared. That's because our database is currently empty — we need to insert some tasks!

### Inserting tasks from the server-side database console

Items inside collections are called documents. Let's use the server database console to insert some documents into our collection. In a new terminal tab, go to your app directory and type:

```
meteor mongo
```

This opens a console into your app's local development database. Into the prompt, type:

```
db.tasks.insert({ text: "Hello world!", createdAt: new Date() });
```

In your web browser, you will see the UI of your app immediately update to show the new task. You can see that we didn't have to write any code to connect the server-side database to our front-end code — it just happened automatically.

Insert a few more tasks from the database console with different text. In the next step, we'll see how to add functionality to our app's UI so that we can add tasks without using the database console.

[← PREVIOUS \(/TUTORIALS/BLAZE/TEMPLATES\)](#)

[NEXT > \(/TUTORIALS/BLAZE/FORMS-AND-EVENTS\)](#)

### Todo App

Build your first Meteor app

#### 1. Creating an app

[\(/tutorials/blaze/creating-an-app\)](#)

#### 2. Templates

[\(/tutorials/blaze/templates\)](#)

#### 3. Collections

[\(/tutorials/blaze/collections\)](#)

#### 4. Forms and events

[\(/tutorials/blaze/forms-and-events\)](#)

#### 5. Update and remove

[\(/tutorials/blaze/update-and-remove\)](#)

#### 6. Running on mobile

[\(/tutorials/blaze/running-on-mobile\)](#)

#### 7. Temporary UI state

[\(/tutorials/blaze/temporary-ui-state\)](#)

#### 8. Adding user accounts

[\(/tutorials/blaze/adding-user-accounts\)](#)

#### 9. Security with methods

[\(/tutorials/blaze/security-with-methods\)](#)

#### 10. Publish and subscribe

[\(/tutorials/blaze/publish-and-subscribe\)](#)

#### 11. Testing

[\(/tutorials/blaze/testing\)](#)

#### 12. Next steps

[\(/tutorials/blaze/next-steps\)](#)



© 2017 Meteor Development Group Inc.

Open source downloads supported by [KeyCDN](https://www.keycdn.com)

(<https://www.keycdn.com>) and [MaxCDN](https://www.maxcdn.com)

(<https://www.maxcdn.com>)

[Home \(/\)](#)

[GitHub](#)

(<https://github.com/meteor/meteor>)

[Twitter \(https://twitter.com/meteorjs\)](https://twitter.com/meteorjs)

[YouTube](#)

(<https://www.youtube.com/user/MeteorVideos>)

[Facebook](#)

(<https://www.facebook.com/meteorjs/>)

[Developers \(/developers\)](#)

[Install \(/install\)](#)

[Tutorials \(/tutorials\)](#)

[Guide \(http://guide.meteor.com/\)](http://guide.meteor.com/)

[Documentation](#)

(<http://docs.meteor.com/#/basic/>)

[Apps Showcase \(/showcase\)](#)

[Forums \(https://forums.meteor.com/\)](https://forums.meteor.com/)

[Meetups](#)

(<http://www.meetup.com/topics/meteor/>)

[Meteor FAQ \(/meteor-faq\)](#)

[Speaker Kit \(/speaker-kit\)](#)

[Articles \(/articles\)](#)

[Solutions \(/solutions\)](#)

[Meteor Hosting \(galaxy\)](#)

[Official Partners \(/partners\)](#)

[Become a Partner \(/become-a-partner\)](#)

[Submit RFP \(/submit-rfp\)](#)

[Galaxy FAQ \(/galaxy-faq\)](#)

[GraphQL \(http://graphql.com\)](http://graphql.com)

[System Status \(http://status.meteor.com/\)](http://status.meteor.com/)

[Company](#)

(<https://www.meteor.io/>)

[Blog](#)

(<https://blog.meteor.com/>)

[Careers](#)

(<https://www.meteor.io/jobs/>)

[Culture](#)

(<https://www.meteor.io/culture/>)

[Press \(/press\)](#)

[Contact Us](#)

([/contact](#))

[Policies \(/policy\)](#)

[Apollo GraphQL](#)

(<http://www.apollographql.com>)