



Meteor

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

 <https://www.facebook.com/tutorialspointindia>

 <https://twitter.com/tutorialspoint>

About the Tutorial

Meteor is a full-stack JavaScript platform for building web and mobile apps. Meteor makes it easier to create real-time apps, since it alone offers a full ecosystem to work with, instead of combining couple of different tools and frameworks to get the same effect.

Audience

This tutorial will be useful for any JavaScript developer who wants to learn Meteor framework. The tutorial is explained in a crisp manner with simple and easy-to-grasp code samples. Beginners will benefit the most from this tutorial. Readers can also use it as a reference while working with Meteor framework.

Prerequisites

Meteor is a JavaScript framework, hence readers will need to have basic knowledge of JavaScript and HTML. They will also need to be familiar with NodeJS and MongoDB although it will also be easy to understand everything without previous knowledge.

If the readers have never used MongoDB, any knowledge about databases should suffice. Since Meteor is a full-stack framework, any previous experience in creating web or mobile apps will be helpful.

Copyright & Disclaimer

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial.....	i
Audience	i
Prerequisites	i
Copyright & Disclaimer.....	i
Table of Contents	ii
1. METEOR – OVERVIEW	1
2. METEOR – ENVIRONMENT SETUP	2
3. METEOR – FIRST APPLICATION	3
4. METEOR – TEMPLATES.....	4
Simple Template	4
Block Template	5
5. METEOR – COLLECTIONS.....	7
Create a Collection	7
Add Data	7
Find Data.....	7
Update Data	9
Delete Data	10
6. METEOR – FORMS.....	13
Text Input.....	13
Radio Buttons	14
Checkbox.....	16
Select Dropdown.....	17
7. METEOR – EVENTS	19

8.	METEOR – SESSION	21
9.	METEOR – TRACKER	23
10.	METEOR – PACKAGES.....	25
	Adding Packages	25
	Removing Packages	25
	Updating Packages	25
	Checking Current Packages	25
	Package Maintenance	25
11.	METEOR – CORE API.....	26
12.	METEOR – CHECK.....	27
	Installing Check Package	27
	Using Check	27
	Match Test	27
13.	METEOR – BLAZE.....	29
	Render Method	29
	Render with Data	30
	Remove Method	31
14.	METEOR – TIMERS.....	34
	Timeout	34
	Interval	35
15.	METEOR – EJSON.....	37
	Install EJSON	37
	Stringify	38

16. METEOR – HTTP	40
Install Package	40
CALL Method.....	40
GET Method	40
POST Method	41
PUT Method.....	42
DEL Method	42
17. METEOR – EMAIL.....	44
18. METEOR – ASSETS	45
19. METEOR – SECURITY	47
Autopublish and Autosecure.....	47
Use Server Side Methods	47
Additional Security.....	47
20. METEOR – SORTING	48
21. METEOR – ACCOUNTS.....	52
Authentication Example.....	52
22. METEOR – METHODS	57
Using Methods	57
Handling Errors	58
23. METEOR – PACKAGE.JS.....	60
Creating a Package	60
Adding a Package	60
Package Files	60
Testing Package (package-name-test.js).....	61

package.js File	62
package-name.js File	62
Using a Package.....	63
24. METEOR – PUBLISH & SUBSCRIBE	65
Removing Autopublish	65
Using Publish and Subscribe	66
Filtering Client Data.....	67
25. METEOR – STRUCTURE.....	69
26. METEOR – DEPLOYMENT.....	70
27. METEOR – RUNNING ON MOBILE.....	71
28. METEOR – TODO APP	73
29. METEOR – BEST PRACTICES.....	78

1. Meteor – Overview

According to Meteor official documentation –

Meteor is a full-stack JavaScript platform for developing modern web and mobile applications. Meteor includes a key set of technologies for building connected-client reactive applications, a build tool, and a curated set of packages from the Node.js and general JavaScript community.

Features

- **Web and Mobile** – Meteor offers a platform for developing Web, Android, and IOS apps.
- **Universal Apps** – The same code for web browsers and mobile devices.
- **Packages** – Huge number of packages that are easy to install and use.
- **Meteor Galaxy** – Cloud service for Meteor app deployment.

Advantages

- Developers only need JavaScript for server and client side development.
- Coding is very simple and beginner friendly.
- Meteor apps are real time by default.
- Official and community packages are huge time saver.

Limitations

- Meteor isn't very suitable for large and complex applications.
- There is a lot of magic going on when working with Meteor, so developers might find themselves limited in some way.

2. Meteor – Environment Setup

In this chapter, we will learn how to install Meteor on Windows operating system. Before we start working with Meteor, we will need **NodeJS**. If you don't have it installed, you can check the links provided below.

Prerequisite

NodeJS is the platform needed for Meteor development. If you do not have NodeJS environment setup ready, then you can check out our [NodeJS Environment Setup](#).

Install Meteor

Download the official meteor installer from [this page](#).

If any error occurs during the installation, try running the installer as an administrator. Once the installation is complete, you will be asked to create a Meteor account.

When you finish installing Meteor installer, you can test if everything is installed correctly by running the following code in the command prompt window.

```
C:\Users\username>meteor
```

Following will be the output –

```
run: You're not in a Meteor project directory.  
To create a new Meteor project:  
  meteor create <project name>  
For example:  
  meteor create myapp  
For more help, see 'meteor --help'.
```


3. Meteor – First Application

In this chapter, we will learn how to create your first Meteor application.

Step 1 - Create the App

To create the app, we will run the **meteor create** command from the command prompt window. The app's name will be **meteorApp**.

```
C:\Users\username\Desktop\Meteor>meteor create meteorApp
```

Step 2 - Run the App

We can run the app by typing the **meteor** command.

```
C:\Users\username\Desktop\meteorApp>meteor
```

This command will start several processes, which can be seen in the following image.

```
=> Started proxy.  
=> Started MongoDB.  
=> Started your app.  
  
=> App running at: http://localhost:3000/  
Type Control-C twice to stop.
```

Step 3 - Verify the Result

Now, we can open the **http://localhost:3000/** address to see how our first Meteor App looks like.



4. Meteor – Templates

Meteor templates are using three top level tags. The first two are **head** and **body**. These tags perform the same functions as in regular HTML. The third tag is **template**. This is the place, where we connect HTML to JavaScript.

Simple Template

Following example shows how this works. We are creating a template with **name = "myParagraph"** attribute. Our **template** tag is created below the **body** element, however, we need to include it before it is rendered on the screen. We can do it by using **{{> myParagraph}}** syntax. In our template, we are using double curly braces (**{{text}}**). This is meteor template language called **Spacebars**.

In our JavaScript file, we are setting **Template.myParagraph.helpers({})** method that will be our connection to our template. We are only using **text** helper in this example.

meteorApp.html

```
<head>
  <title>meteorApp</title>
</head>

<body>
  <h1>Header</h1>
  {{> myParagraph}}
</body>

<template name = "myParagraph">
  <p>{{text}}</p>
</template>
```

meteorApp.js

```
if (Meteor.isClient) {
  // This code only runs on the client
  Template.myParagraph.helpers({
    text: 'This is paragraph...'
  });
}
```

After we save the changes, following will be the output –



Block Template

In the following example, we are using `{{#each paragraphs}}` to iterate over the `paragraphs` array and return template `name = "paragraph"` for each value.

meteorApp.html

```
<head>
  <title>meteorApp</title>
</head>

<body>
  <div>
    {{#each paragraphs}}
      {{> paragraph}}
    {{/each}}
  </div>
</body>

<template name = "paragraph">
  <p>{{text}}</p>
</template>
```

We need to create `paragraphs` helper. This will be an array with five text values.

meteorApp.js

```
if (Meteor.isClient) {  
  // This code only runs on the client  
  Template.body.helpers({  
    paragraphs: [  
      { text: "This is paragraph 1..." },  
      { text: "This is paragraph 2..." },  
      { text: "This is paragraph 3..." },  
      { text: "This is paragraph 4..." },  
      { text: "This is paragraph 5..." }  
    ]  
  });  
}
```

Now, we can see five paragraphs on the screen.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>