What is Meteor?

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.

- Meteor allows you to develop in **one language**, JavaScript, in all environments: application server, web browser, and mobile device.
- Meteor uses **data on the wire**, meaning the server sends data, not HTML, and the client renders it.
- Meteor **embraces the ecosystem**, bringing the best parts of the extremely active JavaScript community to you in a careful and considered way.
- Meteor provides **full stack reactivity**, allowing your UI to seamlessly reflect the true state of the world with minimal development effort.

Quick start

Install the latest official Meteor release following the steps in our docs.

Once you've installed Meteor, open a new terminal window and create a project:

```
meteor create myapp
```

Run it locally:

```
cd myapp
meteor npm install
meteor
# Meteor server running on: http://localhost:3000/
```

> *Meteor comes with npm bundled so that you can type `meteor npm` without worrying about installing it yourself. If you like, you can also use a globally installed npm to manage your packages.*

Meteor resources

1. The place to get started with Meteor is the tutorials page.
2. Meteor Examples is a list of examples using Meteor. You can also include your example with Meteor.
3. Once you are familiar with the basics, the Meteor Guide covers intermediate material on how to use Meteor in a larger scale app.
4. Visit the Meteor discussion forums to announce projects, get help, talk about the community, or discuss changes to core.
5. Meteor Slack Community is the best place to ask (and answer!) technical questions and also meet Meteor developers.
6. Atmosphere is the repository of community packages designed especially for Meteor.

What is the Meteor Guide?

This is a set of articles outlining opinions on best-practice application development using the Meteor platform. Our aim is to cover patterns that are common to the development of all modern web and mobile applications, so many concepts documented here are not necessarily Meteor specific and could be applied to any application built with a focus on modern, interactive user interfaces.

Nothing in the Meteor guide is *required* to build a Meteor application---you can certainly use the platform in ways that contradict the principles and patterns of the guide. However, the guide is an attempt to document best practices and community conventions, so we hope that the majority of the Meteor community will benefit from adopting the practices documented here.

The APIs of the Meteor platform are available at the docs site, and you can browse community packages on atmosphere.

Target audience

The guide is targeted towards intermediate developers that have some familiarity with JavaScript, the Meteor platform, and web development in general. If you are just getting started with Meteor, we recommend starting with the tutorials.

Example apps

If you want to see some examples, we have a repository dedicated with several examples provided by the community showing many concepts that can be used when implementing your application with Meteor. To know more you can here.

Guide development

Contributing

Ongoing Meteor Guide development takes place **in the open** on GitHub. We encourage pull requests and issues to discuss problems with any changes that could be made to the content. We hope that keeping our process open and honest will make it clear what we plan to include in the guide and what changes will be coming in future Meteor versions.

Goals of the project

The decisions made and practices outlined in the guide must necessarily be **opinionated**. Certain best practices will be highlighted and other valid approaches ignored. We aim to reach community consensus around major decisions but there will always be other ways to solve problems when developing your application. We believe it's important to know what the "standard" way to solve a problem is before branching out to other options. If an alternate approach proves itself superior, then it should make its way into a future version of the guide.

An important function of the guide is to **shape future development** in the Meteor platform. By documenting best practices, the guide shines a spotlight on areas of the platform that could be better, easier, or more performant, and thus will be used to focus a lot of future platform choices.

Similarly, gaps in the platform highlighted by the guide can often be plugged by **community packages**; we hope that if you see an opportunity to improve the Meteor workflow by writing a package, that you take it! If you're not sure how best to design or architect your package, reach out on the forums and start a discussion.