

Migrating to Meteor 1.5

This guide is quite short and we think you'll find the upgrade from 1.4 to 1.5 quite painless. We encourage reading full history and comparing the full differences between the versions you are upgrading from and to.

If you find details which are not covered here, please discuss it using the “Discuss” button above, or if you have something super important, open a pull-request to this article using the “Edit on GitHub” button above!

Migrating from a version older than 1.4?

If you're migrating from a version of Meteor older than Meteor 1.4, there may be important considerations not listed in this guide (which specifically covers 1.4 to 1.5). Please review the older migration guides for details:

- Migrating to Meteor 1.4 (from 1.3)
- Migrating to Meteor 1.3 (from 1.2)

`MAIL_URL` should be reviewed

Due to an upgrade in the underlying dependency for the `email` package, it is necessary to check that your `MAIL_URL` is using the correct scheme (e.g. `smtps://` or `smtp://`).

Previously, Meteor would automatically assume that any `MAIL_URL` using port 465 was to be encrypted and automatically changed `smtp://` to `smtps://`. However, this is not always desired, and not always a safe assumption for Meteor.

If your `MAIL_URL` is TLS/SSL-only (and does not need `STARTTLS`), be sure that the `MAIL_URL` starts with `smtps://` and not `smtp://`.

Again, generally speaking, this applies to applications whose `MAIL_URL` already includes `:465`. If an application's mail provider supports `STARTTLS` (i.e. if the `MAIL_URL` uses `:587` and *sometimes* `:25`), the application can continue to use `smtp://` (without the `s`) and the TLS/SSL upgrade will be made by the mail server, if supported.

Unfortunately, the e-mail ecosystem is confusing. More information can be found in the Nodemailer docs.