If you prefer to watch the video, click below.

{% youtube 6RRVU0-Vvm8 %}

{% apibox "Meteor.isClient" %} {% apibox "Meteor.isServer" %}

> `Meteor.isServer` *can be used to limit where code runs, but it does not prevent code from being sent to the client. Any sensitive code that you don't want served to the client, such as code containing passwords or authentication mechanisms, should be kept in the* `server` *directory.*

{% apibox "Meteor.isCordova" %} {% apibox "Meteor.isDevelopment" %} {% apibox "Meteor.isProduction" %}

{% apibox "Meteor.startup" %}

On a server, the function will run as soon as the server process is finished starting. On a client, the function will run as soon as the DOM is ready. Code wrapped in `Meteor.startup` always runs after all app files have loaded, so you should put code here if you want to access shared variables from other files.

The `startup` callbacks are called in the same order as the calls to `Meteor.startup` were made.

On a client, `startup` callbacks from packages will be called first, followed by `<body>` templates from your `.html` files, followed by your application code.

```
// On server startup, if the database is empty, create some initial data.
if (Meteor.isServer) {
  Meteor.startup(() => {
    if (Rooms.find().count() === 0) {
      Rooms.insert({ name: 'Initial room' });
    }
  });
}
```

{% apibox "Meteor.wrapAsync" %}

{% apibox "Meteor.defer" %}

{% apibox "Meteor.absoluteUrl" %}

{% apibox "Meteor.settings" %}

{% apibox "Meteor.release" %}