sitepoint premium

Handout

Your First Meteor 1.2 Application

Lesson 11 - Step 1

David Turnbull

Security

Something we haven't talked about yet is **security**. This isn't the most exciting topic in the world, but if we don't think about security, then we can't share our projects with the world. As such, we'll spend the next couple of chapters talking about a couple of fundamental security topics.

It's worth mentioning, however, that security is one of those topics that can appear weird and confusing at first, but will suddenly click a little later down the line. If things don't sink in right away, continue building projects with Meteor but avoid sharing anything on the web. After you've built a few projects yourself, the details of securing those projects will start to become clearer.

Publications and Subscriptions

In this chapter, we're going to talk about publications and subscriptions, and the best way to explain these concepts is with a demonstration. To begin the demonstration, create two user accounts. Under each account, add three players to each list. Then log out of both accounts. Because of this, a total of six players should exist inside the database and those players should belong to a total of two users.

Next, use the `find` and `fetch` functions on the `PlayersList` collection. You'll notice that, as we've seen before, all of the data from the collection is returned. This means we can see all of the data that belongs to both of the users. But this is a problem, because unless we prevent users from accessing this data, every user will have the same unrestricted access to everything inside the database.T here's nothing stopping users from using the `find` and `fetch` functions. You might be wondering, though, why does this feature exist within Meteor? Why are we allowed unrestricted access to all this data? That's simply a matter of convenience. Being able to navigate through our database is extremely useful during the development process.

Securing The Application

It's just that before we share the application with the world, we'll have to prevent users from accessing all of this data, and precisely define what data should be available to users. The functionality that allows this unrestricted access to data is contained within a package known as autopublish, which is installed in every Meteor project by default. If we remove this package, users won't be able to access any data through the console, but will also break the application, so we'll have to take a couple of steps to get it working again.

To remove the autopublish package, run

```
meteor remove autopublish
```

If you're logged out of both user accounts when the package is removed, you won't notice anything different at first, but try using the `find` and `fetch` functions - you'll see that we can no longer navigate through the data that's inside the collection. It looks like the data has been deleted, but that's not the case, it has just been secured. The problem now, though, is that the data is too secure, because if we log into either of the accounts we'll see that none of the data is appearing inside the interface. This is what we'll be fixing throughout the rest of this chapter.