



Introduction to MeteorJS

by Andrew McPherson

a bit about me.



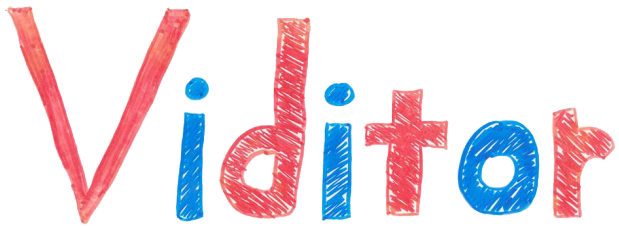
Andrew McPherson
I like videogames.

1. Aspiring developer and entrepreneur.
2. Born and raised here in Chattanooga.
3. Student of computer science at UTC.

github.com/arcym

twitter.com/andrewmcp333

a bit about my projects.



My current project is
“the online video editor”

- Originally it was built through a lot of NodeJS with MongoDB and SocketIO.
- But I had a lot of issues in maintaining consistency of data between clients.
- Then I stumbled upon MeteorJS!

a bit about meteor.

Meteor is a javascript framework that offers...

- client data synchronization.
- reactive updating templates.
- external module interoperability.
- just really simple syntax.

It doesn't want to replace existing frameworks, but integrate upon them.

**Want to see a bit of
syntax?**

Starting the project

To initialize...

```
meteor create NoogaDocs
```

```
> NoogaDocs created!
```

```
cd NoogaDocs
```

```
ls
```

```
> .meteor
```

```
> NoogaDocs.css
```

```
> NoogaDocs.html
```

```
> NoogaDocs.js
```

To execute...

```
meteor
```

```
> Started proxy.
```

```
> Started database.
```

```
> Started application.
```

```
> App at 127.0.0.1:3000
```

Trying it out

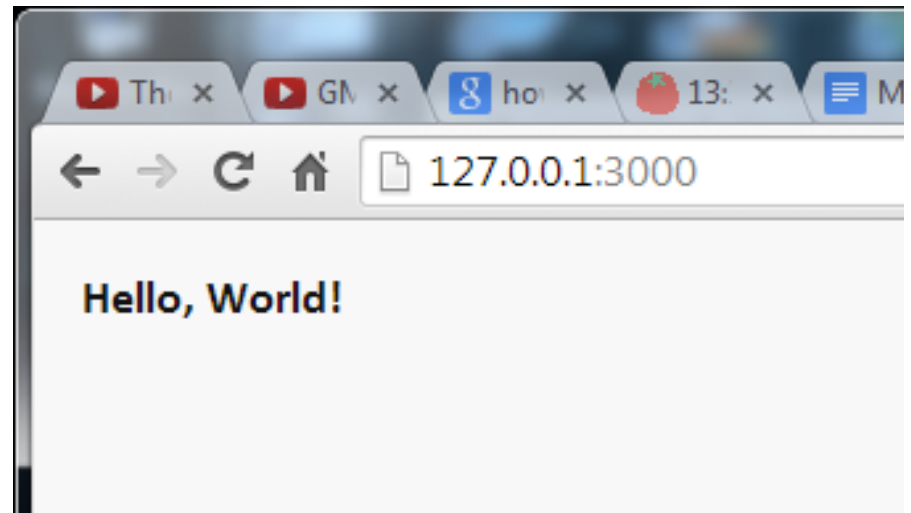
In the .html

```
<head>
  <title>NoogDocs</title>
</head>

<body>
  <div id="greeting">
    Hello, {{name}}!
  </div>
</body>
```

In the .js

```
if(Meteor.isClient)
{
  Template.name = "World";
}
```



Defining a template

In the .html

```
<template name="greet">
  <div id="greeting">
    Hello, {{name}}!
  </div>
</template>
```

```
<body>
  {{> greet}}
</body>
```

In the .js

```
if(Meteor.isClient)
{
  var who = "World"
  Template.greet.name = who
}
```


Using more functions

In the .js

```
if(Meteor.isClient)
{
  Template.greet.name = function()
  {
    var who = "World";
    return who;
  }

  //this is just another way of doing it.
}
```

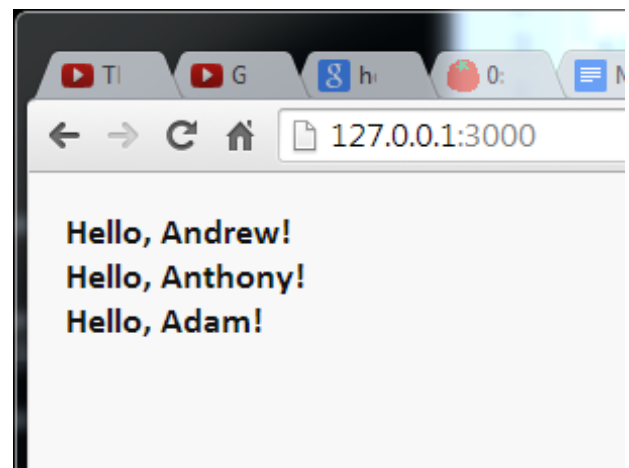
Accessing the data

In the .js

```
if(Meteor.isClient)
{
  Template.greet.name = function()
  {
    var who = this || "World";
    return who;
  }
}
```

In the .html

```
{{> greet "Andrew"}}
{{> greet "Anthony"}}
{{> greet "Adam"}}
```



Iterating via helpers

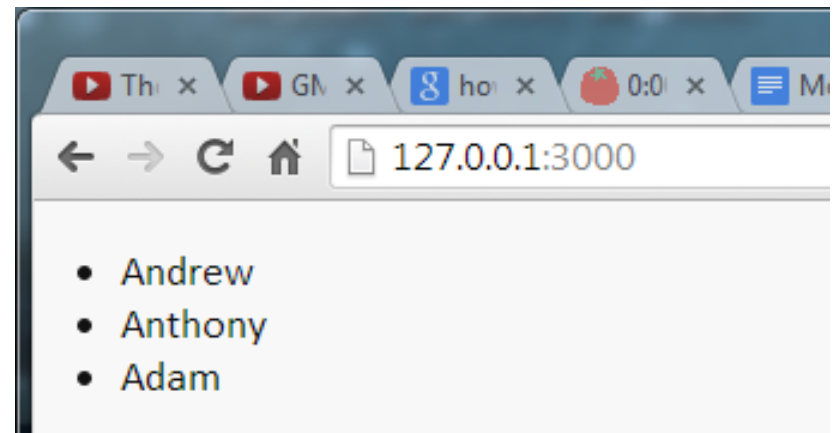
In the .html

```
<template name="list">
  <ul>
    {{#each persons}}
      <li>{{name}}</li>
    {{/each}}
  </ul>
</template>

<body>
  {{> list}}
</body>
```

In the .js

```
Template.list.persons = [
  {name: "Andrew"},
  {name: "Anthony"},
  {name: "Adam"}
];
```



Including a database

In the .js

```
var Persons = new Meteor.Collection("persons");
```

```
if(Meteor.isClient)
{
    Template.list.persons = Persons.find({});
}
```

```
if(Meteor.isServer)
{
    Persons.insert({name: "Andrew"});
    Persons.insert({name: "Anthony"});
    Persons.insert({name: "Adam"});
}
```

Binding to events

In the .js

```
var Persons = new Meteor.Collection("persons");
```

```
if(Meteor.isClient)
```

```
{
```

```
  Template.list.persons = Persons.find({});
```

```
  Template.list.events = {
```

```
    "click li": function() {
```

```
      Persons.remove(this._id);
```

```
    }
```

```
  }
```

```
}
```



So let's try a **live demo!**

Hopefully I won't mess this up! (May the programming gods have mercy on my soul)

some issues with meteor

- potential networking latency
- best for single page apps
- still in development :\
 -
 -
 -
- not much else than that.

some resources for meteor

Meteor Documentation: docs.meteor.com

Meteor Examples: meteor.com/learn-meteor

NoogaDocs Github: github.com/arcym/noogadocs

“Useful Meteor Resources”

discovermeteor.com/blog/useful-meteor-resources

“How to learn Javascript properly”

javascriptissexy.com/how-to-learn-javascript-properly

Thank you so much!