

METEOR BEFORE IMPACT

Niko Köbler

[@dasniko](#)

Heiko Spindler

[@brainbrix](#)

[Qualitects Group](#)

**WHAT IS
METEOR?**

NODE.JS

MONGO DB

WELL-KNOWN & PRODUCTIVITY-PROVEN

JAVASCRIPT LIBRARIES

PACKAGED INTO ONE

POWERFUL PLATFORM!



Uncle Bob Martin

@unclebobmartin

 Follow

I was very impressed by this video. Meteor could be the shape of things to come. meteor.com

11:19 PM - 27 Mar 2014

39 RETWEETS 62 FAVORITES



7 PRINCIPLES

1. DATA ON WIRE
2. ONE LANGUAGE
3. DATABASE EVERYWHERE
4. LATENCY COMPENSATION
5. FULL STACK REACTIVITY
6. EMBRACE THE ECOSYSTEM
7. SIMPLICITY EQUALS
PRODUCTIVITY

1.

DATA

ON

WIRE

2.

**ONE
LANGUAGE**

3.

DATABASE

EVERYWHERE

4.

LATENCY

COMPENSATION

5.

**FULL
STACK**

REACTIVITY

6.

**EMBRACE
THE
ECOSYSTEM**

7.

SIMPLICITY

EQUALS

PRODUCTIVITY

QUICKSTART

Install Meteor:

```
$ curl https://install.meteor.com | /bin/sh
```

Create a project:

```
$ meteor create myapp
```

Run it locally:

```
$ cd myapp  
$ meteor  
=> Meteor server running on: http://localhost:3000/
```

**STRUCTURE
&
ARCHITECTURE**

GENERATED JS

```
if (Meteor.isClient) {  
  someFunction = function() {  
    // your code goes here  
  };  
  ...  
}  
  
if (Meteor.isServer) {  
  Meteor.startup(function() {  
    // code to run on server at startup  
  });  
  ...  
}
```

FOLDER STRUCTURE

```
/myapp/...
```

```
/myapp/lib/...  
/myapp/somefolder/...
```

```
/myapp/server/lib/...  
/myapp/server/someLib.js  
/myapp/server/main.js
```

```
/myapp/client/lib/...  
/myapp/client/someLib.js  
/myapp/client/main.js
```

```
/myapp/public/...
```


**MONGO DB,
MINIMONGO
&
COLLECTIONS**

SYNCHRONIZED COLLECTIONS

```
Messages = new Meteor.Collection("messages");
```

```
    Messages.find();  
    Messages.findOne();  
    Messages.insert();  
    Messages.update();  
    Messages.remove();  
    ...
```

**PUBLISH/SUBSCRIBE
&
METHOD CALLS**

PUBLISH/SUBSCRIBE

```
// server: publish the messages collection
Meteor.publish("messages", function () {
  return Messages.find();
});
```

```
// client: subscribe to the published messages
Meteor.subscribe("messages");
```

METHOD CALLS

```
Meteor.methods({
  foo: function (arg1, arg2) {
    // .. do stuff ..
    if (you want to throw an error)
      throw new Meteor.Error(404, "Can't find my pants");
    return "some return value";
  },

  bar: function () {
    // .. do other stuff ..
    return "baz";
  }
});
```

```
// async call
Meteor.call('foo', 1, 2, function (error, result) { ... } );
```

```
// sync call
var result = Meteor.call('foo', 1, 2);
```

ACCOUNTS & SECURITY

ACCOUNTS

```
$ meteor add accounts-ui  
$ meteor add accounts-*
```

* = password, facebook, twitter, google, github, ...
OAuth2

```
{{> login-buttons}}
```

ALLOW/DENY

```
Messages.allow({
  insert: function (userId, msg) {
    // only logged-in users can insert a new message that they own
    return (userId && msg.owner == userId);
  },
  fetch: ['owner']
});
```

```
Messages.deny({
  remove: function (userId, msg) {
    //can't remove locked messages
    return msg.locked;
  },
  fetch: ['locked']
});
```


**TEMPLATING,
LIVE HTML**

&

HOT CODE REPLACEMENT

SPACEBARS TEMPLATE

```
<head>
  <title>myapp</title>
</head>

<body>
  {{> hello}}
</body>

<template name="hello">
  <h1>Hello World!</h1>
  {{greeting}}
  <input type="button" value="click" />
</template>
```

LIVE UPDATE

```
Template.hello.greeting = function() {  
    return Session.get("welcome_message");  
};  
  
// somewhere in the code...  
Session.set("welcome_message", "Welcome to myapp.");
```

DEPLOYMENT & PACKAGING

DEPLOYMENT

ON METEOR INFRASTRUCTURE

```
$ meteor deploy myapp.meteor.com
```

```
$ meteor deploy www.myapp.com
```

ON OWN INFRASTRUCTURE

```
$ meteor bundle myapp.tgz
```

ECOSYSTEM

PACKAGE MANAGEMENT

ATMOSPHERE

NPM



Werkzeuge 06.09.2013 - 10:58

Schlagwörter: node.js, MongoDB, Meteor, JavaScript

Niko Köbler, Heiko Spindler

Einstieg in die Entwicklung von Web-Apps mit Meteor

Vor dem Einschlag



JavaScript-Webframeworks gibt es wie Staub auf dem Mond. Jedoch ist jedes nur auf dem Gebiet gut, für das es geschaffen wurde. Meteor versteht sich als eine Plattform, die eine Auswahl der besten Frameworks mit MongoDB und Node.js kombiniert und Entwicklern so das Zusammensuchen abnehmen möchte. Ihre Hauptmerkmale sind eine einheitliche Basis in JavaScript-Code für Client und Server, das automatisches Propagieren der Änderungen von Code und Daten, ein vergleichsweise einfaches API-Design und intelligentes Packaging und Deployment von Applikationen.

[Meteor](#) ist eine JavaScript-Plattform, die von der Datenbank über den Server bis hin zum Client alles abdecken möchte und dabei den Schwerpunkt auf das Aktualisieren von Daten in Echtzeit über alle Clients hinweg legt. Als Datenbank kommt [MongoDB](#) zum Einsatz, während auf dem Server [Node.js](#) Verwendung findet – alle Ressourcen lassen sich folglich mit einer einzigen Sprache nutzen.

Weiterhin setzt Meteor auf viele bekannte JavaScript-Bibliotheken, wie Amplify, AppCache, Backbone.js, Handlebars, jQuery und Underscore.js, und kombiniert sie unter einer eigenen API. Auch die Verwendung von CoffeeScript in Meteor-Applikationen ist möglich. Zurzeit liegt das Framework in der Version 0.6.5 vor, ein genauer Termin für eine Version 1.0 ist noch nicht bekannt. Die Entwickler selbst sagen "mehr als einen Monat, weniger als ein Jahr", was ihnen viel Spielraum lässt.

Artikelanfang

[Die erste App](#)

[Mechanismen und Methoden](#)

[Absichern und verteilen](#)

[Fazit](#)

Mit DVD



DEVELOPER

€ 12,90

Österreich € 14,30
Schweiz CHF 23,90
Dänemark € 14,90
Niederlande € 14,90

1/2014

**Auf der
Heft-DVD:
Über 5 GByte
für Entwickler**

Visual Studio Express
2013 für Webentwicklung,
Wakanda, Frameworks wie
Ember.js und AngularJS,
Gitter.com und andere
nützliche Tools, Codebeispiele,
Leseproben alterer JavaScript-
Publikationen, 40 Episoden des Software-
Archiv-Club-Podcasts.

JavaScript heute

Grundlagen:

**Sprachparadigmen, Laufzeitumgebungen
und Entwurfsmuster**

Frameworks:

**Einsatz in professionellen Projekten
Ember.js 1.1, AngularJS und Co.**

Sprachen und Standards:

**Was ECMAScript 6 bringt
Schneller mit asm.js
CoffeeScript und
andere Alternativen**

Server-side JavaScript:

Node.js und sein Ökosystem

Datenträger enthält
Info- und
Lehrprogramme
gemäß § 14 JuSchG

Qualitätssicherung:

**Mocha, Grunt und
Headless Browser Testing
Tipps für sichere Anwendungen**



Vertriebspartner
heise Developer
www.heise.de/developer



AMAZON

GO AND BUILD YOUR OWN!

www.meteor.com

github.com/dasniko/meteor-reactive-xmpl

github.com/dasniko/meteor-chat

github.com/brainbrix/PokerQuiz