

2.– 5. September 2013  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## Meteor vor dem Einschlag

Ein flexibles JavaScript Framework

Heiko Spindler

# METEOR BEFORE IMPACT

Niko Köbler  
@dasniko

Heiko Spindler  
@brainbrix

# WHAT IS METEOR?

**NODE.JS**

**MONGO DB**

**WELL-KNOWN & PRODUCTIVITY-PROVEN**

**JAVASCRIPT LIBRARIES**

**PACKAGED INTO ONE  
POWERFUL PLATFORM!**

# 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**

# 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 & COLLECTIONS

# SYNCHRONIZED COLLECTIONS

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

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

# 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']
});
```

**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);
```

**TEMPLATING,  
LIVE HTML**

**&**

**HOT CODE REPLACEMENT**

# HANDLEBARS 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.");
```

# ACCOUNTS & SECURITY

# ACCOUNTS

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

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

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

# DEPLOYMENT & PACKAGING

# DEPLOYMENT

## ON METEOR INFRASTRUCTURE

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

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

## ON OWN INFRASTRUCTURE

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

# ECOSYSTEM

# METEORITE & ATMOSPHERE

**GO  
AND BUILD  
YOUR OWN!**

[www.meteor.com](http://www.meteor.com)

[github.com/brainbrix/PokerQuiz](https://github.com/brainbrix/PokerQuiz)



**STRUCTURE & ARCHITECTURE**  
**MONGO DB & COLLECTIONS**  
**PUBLISH/SUBSCRIBE & METHOD CALLS**  
**TEMPLATING & LIVE-UPDATE**  
**ACCOUNTS & SECURITY**  
**DEPLOYMENT & PACKAGING**  
**ECOSYSTEM**