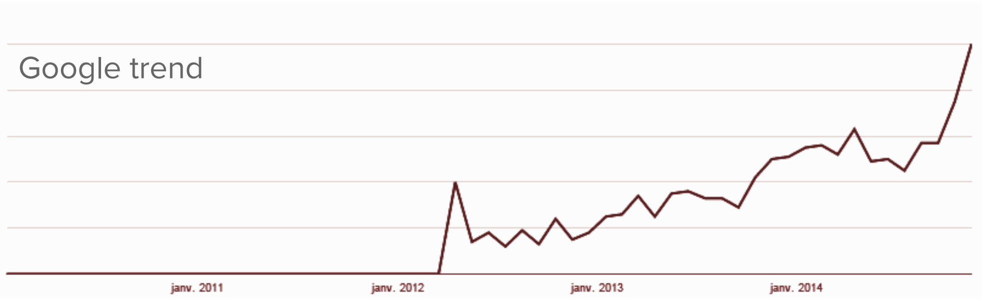
BOTIFY

METEOR

nov. 2014 - Nicolas

Twitter.com/botify

Meteor is a **full stack** open source platform for building web and mobile apps in pure **JavaScript**.



★ 20k

Twitter.com/botify

Principles of Meteor

One Language Everywhere

Both client and server parts
are written in JavaScript.

Database Everywhere

You can use the **same methods**
to access your database
from the **client** or the **server**.

Full Stack Reactivity

Realtime is the default.

All layers, from database to template, update themselves automatically when necessary.

Latency Compensation

On the client,
Meteor prefetches data making it look like
server answer instantly.

Modular & Unified Packages

Packages work identically
on browsers, servers and mobile devices

Packages can be replaced easily to fit needs

Rich Ecosystem

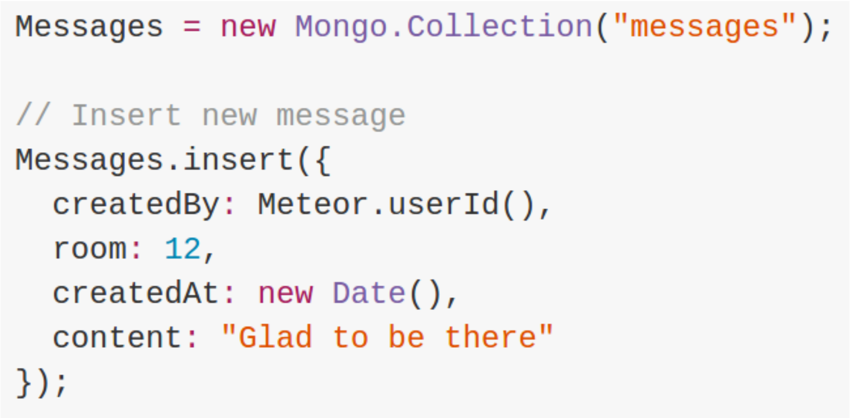
- Easy deployment
- Meteor servers to run prototypes or production apps
- Testing environment
- Bug reporting
- 3k+ packages availables

Let’s start

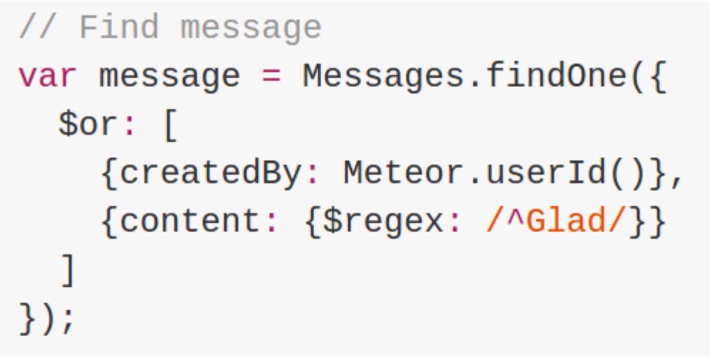
https://github.com/zallek/meteor-introduction

MongoDB Collections

Accessible on both clients & server
using the same MongoDB syntax.



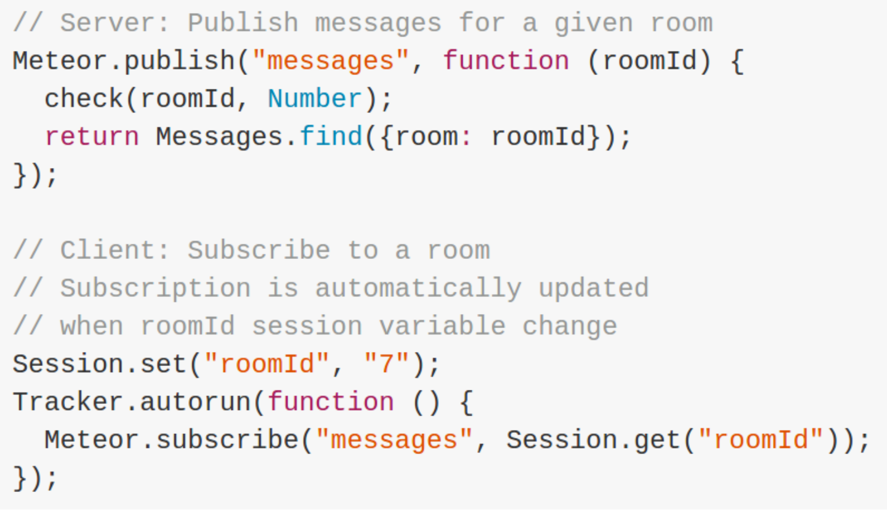
```
Messages = new Mongo.Collection("messages");

// Insert new message
Messages.insert({
  createdBy: Meteor.userId(),
  room: 12,
  createdAt: new Date(),
  content: "Glad to be there"
});
```



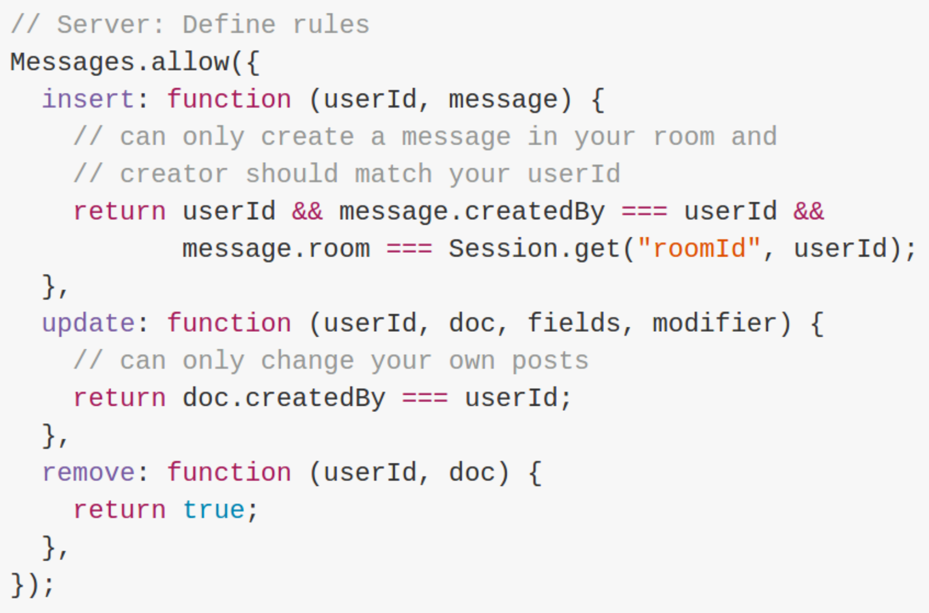
```
// Find message
var message = Messages.findOne({
  $or: [
    {createdBy: Meteor.userId()},
    {content: {$regex: /^Glad/}}
  ]
});
```

Publish/Subscribe

Servers publish data, clients can subscribe to



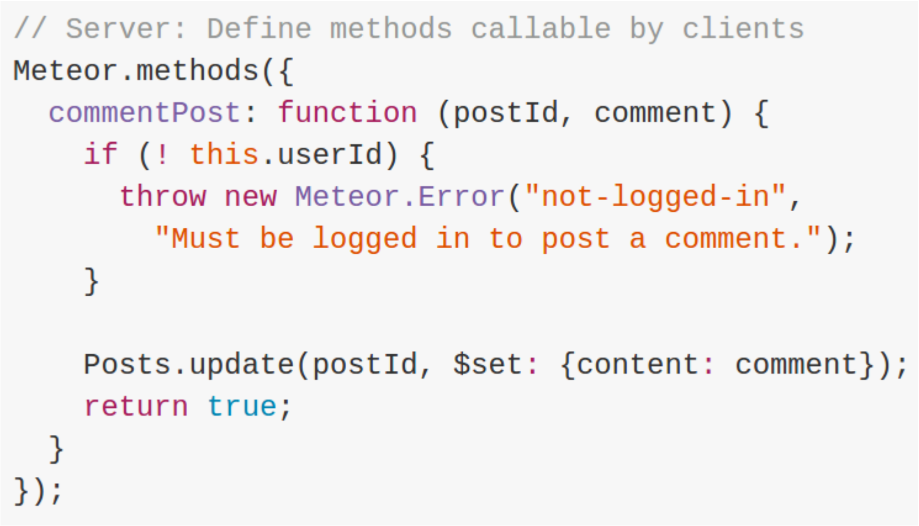
```
// Server: Publish messages for a given room
Meteor.publish("messages", function (roomId) {
  check(roomId, Number);
  return Messages.find({room: roomId});
});

// Client: Subscribe to a room
// Subscription is automatically updated
// when roomId session variable change
Session.set("roomId", "7");
Tracker.autorun(function () {
  Meteor.subscribe("messages", Session.get("roomId"));
});
```



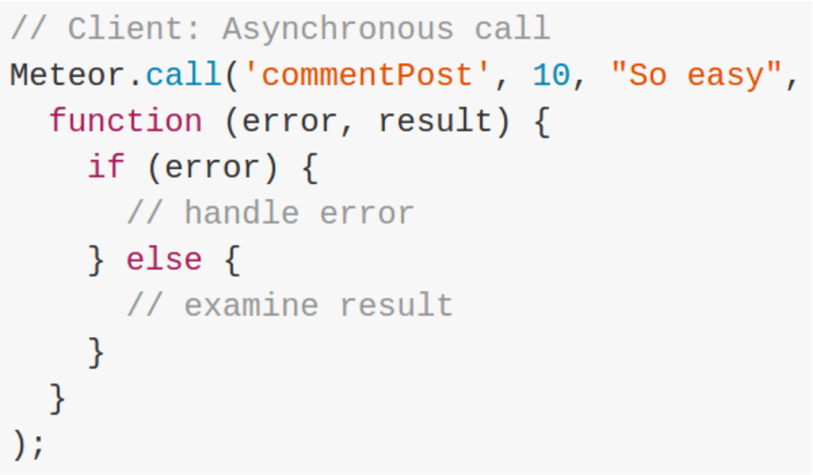
```
// Server: Define rules
Messages.allow({
  insert: function (userId, message) {
    // can only create a message in your room and
    // creator should match your userId
    return userId && message.createdBy === userId &&
           message.room === Session.get("roomId", userId);
  },
  update: function (userId, doc, fields, modifier) {
    // can only change your own posts
    return doc.createdBy === userId;
  },
  remove: function (userId, doc) {
    return true;
  },
});
```

Server methods

API methods can be defined for complex data processing



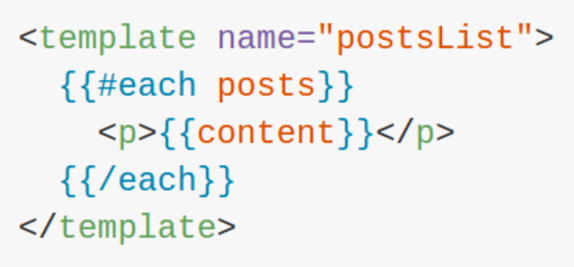
```
// Server: Define methods callable by clients
Meteor.methods({
  commentPost: function (postId, comment) {
    if (! this.userId) {
      throw new Meteor.Error("not-logged-in",
        "Must be logged in to post a comment.");
    }

    Posts.update(postId, $set: {content: comment});
    return true;
  }
});
```



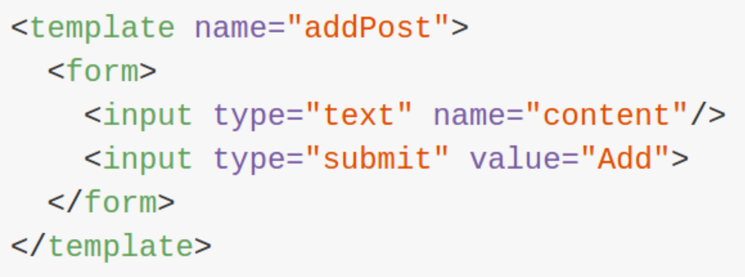
```
// Client: Asynchronous call
Meteor.call('commentPost', 10, "So easy",
  function (error, result) {
    if (error) {
      // handle error
    } else {
      // examine result
    }
  }
);
```

Templating

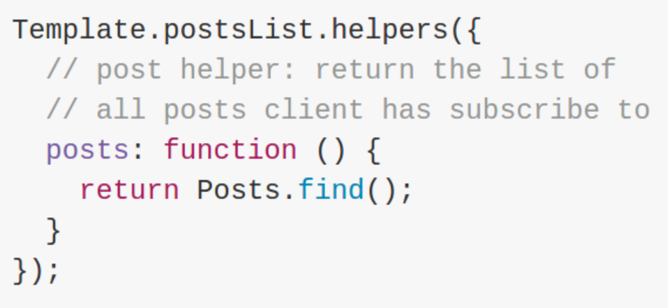
By default, Meteor use **Blaze** and **Spacebars** to render templates



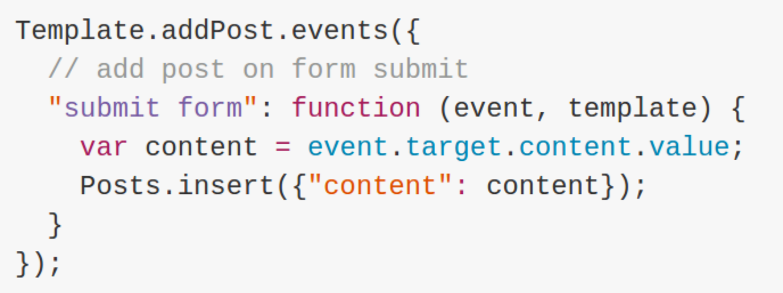
```
<template name="postsList">
  {{#each posts}}
    <p>{{content}}</p>
  {{/each}}
</template>
```



```
<template name="addPost">
  <form>
    <input type="text" name="content"/>
    <input type="submit" value="Add">
  </form>
</template>
```



```
Template.postsList.helpers({
  // post helper: return the list of
  // all posts client has subscribe to
  posts: function () {
    return Posts.find();
  }
});
```



```
Template.addPost.events({
  // add post on form submit
  "submit form": function (event, template) {
    var content = event.target.content.value;
    Posts.insert({"content": content});
  }
});
```

Want more ?

More features

- Built-in User Accounts
    - Password accounts
    - Facebook, Google, Twitter (...) accounts
- Custom Reactive Data
- Easy OAuth encryption

New in 1.0

- Mobile App Development
- Redis support (beta)
- Testing
- Bug Reporting
- Improve security
- Improve documentation

Future

- Compatibility with SQL Databases
- Improve performance
- Internationalization
- Widgets system
- …

https://trello.com/b/hjBDflxp/meteor-roadmap

Pros & Cons

Pros

- Productivity
- Easy clients synchronisation
- Modular
- Frequently updated

Cons

- Scalability
    - Static files delivery
    - Clients data synchronisation
    - Persistants connections to each client

https://meteorhacks.com/does-meteor-scale.html

Thank you

Twitter.com/botify

Get informed

github.com/meteor/meteor

meteorhacks.com

meetup.com/Meteor-Paris