

基于 NodeJS 的跨平台聊天应用的设计与实现

赵 亮、赵绪辉

(渤海大学 信息学院, 锦州 121000) 渤海大学计算机科学与技术学院

(甘肃省陇南市武都区建国路国税局家属院 6 单元 101 15694267972)

摘 要: 当今互联网聊天软件层出不穷, 且主要面向移动端用户。本文将设计并使用基于 NodeJS 的 Meteor 框架开发一款跨平台的聊天软件。应用的通讯服务器采用 kadora 团队的 mup 来部署。前后端均有 JavaScript 语言来实现。前端框架使用 AngularJS 与 Ionic 以达到良好的移动端体验效果。数据库使用 NoSQL 类型的 MongoDB 数据库。该聊天应用将会实现 WEB、Android、iOS 三端实时聊天的功能。

关键词: 聊天应用; nodejs; Meteor

1、相关技术简介

1.1 Node.js

Node.js 是一个开源的、跨平台的 JavaScript 运行时环境, 用来构建各种各样的工具和应用程序。尽管 Node.js 不是一个 JavaScript 框架, 但是许多基础模块都是由 JavaScript 编写的, 并且开发者同样可以用 JavaScript 来编写 Node.js 的新模块。其解释器使用了 Google 的 V8 JavaScript 引擎。Node.js 拥有一个可异步操作的事件驱动构型。这些设计的选择旨在与铸就 Web 应用程序的良好扩展性和读写效率。虽然让 Javascript 运行于服务器端不是 Node 的独特之处, 但却是其一大功能。不得不承认, 浏览器环境限制了我们选择编程语言的自由。任何服务器与日益复杂的浏览器客户端应用程序间共享代码的愿望只能通过 Javascript 来实现。虽然还存在其他一些支持 Javascript 在服务器端 运行的平台, 但因为上述特性, Node 发展迅猛, 成为事实上的平台。

1.2 Meteor

Meteor 或称 MeteorJS 是一个基于 NodeJS 的免费并且开源的 JavaScript Web 框架。Meteor 出现的意义就在于快速全栈开发和跨平台开发。Meteor 本身解释 MongoDB 数据库并且使用数据分发规则和“订阅发布”来根据客户端数据变动实时反映到数据库而不用开发者去编写任何的同步代码。在客户端 Meteor 支持 jQuery 和 JavaScript 组件库。

1.3 AngularJS

AngularJS 是一款开源 JavaScript 库, 由 Google 维护, 用来协助单一页面应用程序运行的。它的目标是通过 MVC 模式 (MVC) 功能增强基于浏览器的应用, 使开发和测试变得更加容易。AngularJS 支持双向数据绑定, Angular 在呈现和数据中间, 可以简单创建双向的数据绑定。一旦创建双向绑定, 用户输入, 会由 Angular 自动传到一个变量中, 再自动读到所有绑定到它的内容, 更新它。效果上就是立即的数据同步。在代码中修改变量, 也会直接反应到呈现的外观上。不仅内容可以双向绑定, 其他诸如类别、宽度、高度等等, 都可以和变量与用户的输入, 绑定起来。

1.4 面向文档的分布式数据库 MongoDB

MongoDB 是一个为当代 Web 应用而生的 NoSQL 数据库。MongoDB 可以把关系型数据库的表理解为一个电子表格, 列表示字段, 每行的记录其实是按照列的字段顺序排列的值得元组。而存储在 MongoDB 中的文档被存储为键-值对的形式, 值却可以是任意类型且可嵌套。使用关系型数据库的时候, 我们需要把信息打散到不同的表中, 要通过关系表或者使用 join 拼接成复杂的 SQL 语句的方式才能获得需要的数据。现在我们可以更多地把信息放在一起, 也不需要提前预定信息的模式。MongoDB 还是用了高效的二进制 BSON 作为数据处理。BSON 是一个类 JSON 的格式, 选择 BSON 可以提供更快的遍历速度, 提供比 JSON 更多的内置数据类型。同时自带高可用及分区的解决方案, 分别为副本集 (Replica Set) 和分片 (sharding)。

1.5 Ionic

Ionic 是一款开源的前端框架, 帮助开发者设计基于 Html5 的 hybrid 移动应用。Ionic 提供了丰富的移动端定制组件。Ionic 同时支

持 AngularJS 和 SaSS 以帮助开发者更方便的设计应用界面样式。Ionic 提供的这些组件, 看起来和系统原生组件一模一样, 毫无违和感。响应速度是基于 HTML5 的移动应用的重点。Ionic 压缩 DOM 并使用硬件加速来提高响应速度。

2、系统设计

2.1 数据库连接

```
import { Mongo } from 'meteor/mongo';
export const Chats = new Mongo.Collection('chats');
export const Messages = new Mongo.Collection('messages');
```

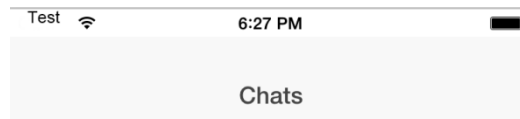
其中 chats 集合数据格式为:

```
Chats = [
  {
    Name: 用户名,
    Pictrue: 图片
  },
  ...
]
```

message 数据格式为:

```
Message = [
  {
    Text: 消息内容,
    Timestamp: 消息的时间戳
  }
]
```

2.2 聊天列表



2.3 信息展示

将 chats 和 message 集合从数据库模型读取到视图中，代码如下：

```
import { Chats, Messages } from '../lib/collections';

export default class ChatCtrl extends Controller {
  constructor() {
    ...some lines skipped...
    this.chatId = this.$stateParams.chatId;

    this.helpers({
      messages() {
        return Messages.find({ chatId: this.chatId });
      },
      data() {
        return Chats.findOne(this.chatId);
      }
    });
  }
}
```



2.4 聊天界面

Ionic 模板代码如下：

```
<ion-footer-bar keyboard-attach class="bar-stable footer-chat
item-input-inset">
  <button class="button button-clear button-icon button-positive icon
ion-ios-upload-outline"></button>

  <label class="item-input-wrapper">
    <input ng-model="chat.message"
      dir="auto"
      type="text"/>
  </label>

  <span ng-if="chat.message.length > 0">
    <button ng-click="chat.sendMessage()" class="button button-clear
button-positive">Send</button>
  </span>
  <span ng-if="!chat.message || chat.message.length === 0">
    <button class="button button-clear button-icon button-positive icon
ion-ios-camera-outline"></button>
  <i class="buttons-seperator icon ion-android-more-vertical"></i>
  <button class="button button-clear button-icon button-positive icon
```

```
ion-ios-mic-outline"></button>
</span>
</ion-footer-bar>
```



2.5 发送消息

```
sendPicture() {
  MeteorCameraUI.getPicture({}, (err, data) => {
    if (err) return this.handleError(err);

    this.callMethod('newMessage', {
      picture: data,
      type: 'picture',
      chatId: this.chatId
    });
  });
}
```

```
sendMessage() {
  if (!_isEmpty(this.message)) return;
```

```
this.callMethod('newMessage', {
  text: this.message,
  type: 'text',
  chatId: this.chatId
});
delete this.message;
}
```

2.6 创建和删除 chat 的 Ionic 模板设计

```
<ion-nav-buttons side="right">
  <button ng-click="chats.showNewChatModal()" class="button
button-clear button-positive button-icon ion-ios-compose-outline"><
  </button></ion-nav-buttons>
```

3、结束语

经测试，该聊天应用在 Web 端和移动端均表现良好，特别是在移动端，使用了 Ionic 构建的 UI 界面在 iOS 系统下表现与原生应用无异。Meteor 与 Ionic 的结合，使得应用程序一次编写，处处能用，免除了开发者进行跨平台编写的工作，提高了开发效率。Ionic 的原生样式组件同样减轻了开发基于 HTML5 移动端应用时 UI 设计的工作量。语言方面来讲，前后端均为 JavaScript，数据库也是 JavaScript (MongoDB)；数据库方面，使用 MongoDB 的自由度非常大，不需要在开发初期设计好数据模型，完全可以在开发进程中不断演进。Meteor 的所见即所得的特性也提高了调试效果和排错效率。在项目初期，使用 Meteor+Ionic 的方案，可以极大的提高产品原型的构建速度，最小化人力资源的浪费。