

基于 Web 的即时通信系统

——使用手册

一、数据库安装

- a) 安装 Mysql 数据库，在本实验中我安装的是 Mysql5.7.10 版本，具体安装可以上网搜索教程
- b) 数据库账户密码设置为 root/root，当然也可以使用自己喜欢的账户密码，但在后面开发中要改变相应的配置。
- c) 项目中的数据库配置

```
var mysql=require('mysql');  
  
function connectServer(){  
    var client=mysql.createConnection({  
        host:'localhost',  
        user:'root',  
        password:'root',  
        database:'webchat'  
    });  
    return client;  
}
```

若账户密码不同，则在代码中也要做相应修改

- d) 数据库操作——创建数据库 webchat
- e) 数据库操作——创建表 user friendship message

create database webchat;

```
create table user(  
    username varchar(20) not null primary key,  
    password varchar(20) not null,  
    nickname varchar(50) not null,  
    email varchar(50) not null  
);
```

```
create table firendship(  
    own_username varchar(20) not null,  
    other_username varchar(20) not null,  
    index(own_username),  
    foreign key(own_username) references user(username)  
);
```

```
create table message(  
    messageid int not null primary key auto_increment,  
    sender varchar(20) not null,  
    receiver varchar(20) not null,  
    send_time datetime not null,  
    ifread boolean not null default '0',
```

```

content varchar(255) not null,
index(sender),
foreign key(sender) references user(username),
index(receiver),
foreign key(receiver) references user(username)
);

```

f) 完成后的数据库表信息

```

+-----+
| Tables_in_webchat |
+-----+
| firendship         |
| message            |
| user               |
+-----+

```

```

mysql> select * from user;
+-----+
| username | password | nickname | email |
+-----+

```

```

mysql> select * from message;
+-----+
| messageid | sender | receiver | send_time | ifread | content |
+-----+

```

```

mysql> select * from firendship
-> ;
+-----+
| own_username | other_username |
+-----+

```

g) 至此数据库已配置好

二、开发环境配置，nodejs+express4 环境配置

- 下载并安装 Node.js，直接在 www.nodejs.org 官网下载并安装即可
- 打开命令提示符进入 node.js 安装目录，输入命令 `npm install express -v` 安装 express，对于 express4 以上的版本，需要额外输入命令 `npm install express-generator -v`
- 安装完后进入项目所在目录，使用命令 `npm install` 安装项目所依赖的包，也可根据下列的文件自己逐个安装

```

{
  "dependencies": {
    "socket.io": "~1.3.5",
    "body-parser": "~1.12.4",
    "cookie-parser": "~1.3.5",
    "debug": "~2.2.0",
    "express": "~4.12.4",
    "jade": "~1.9.2",
    "morgan": "~1.5.3",
    "serve-favicon": "~2.2.1",
    "markdown-js": ">= 0.0.1",
    "mysql": "latest",
    "xss": "0.2.2"
  }
}

```

- 安装完后就可以在本地实现项目的运行了
进入项目所在目录，输入命令 `npm start` 开启项目，如下图所示

```

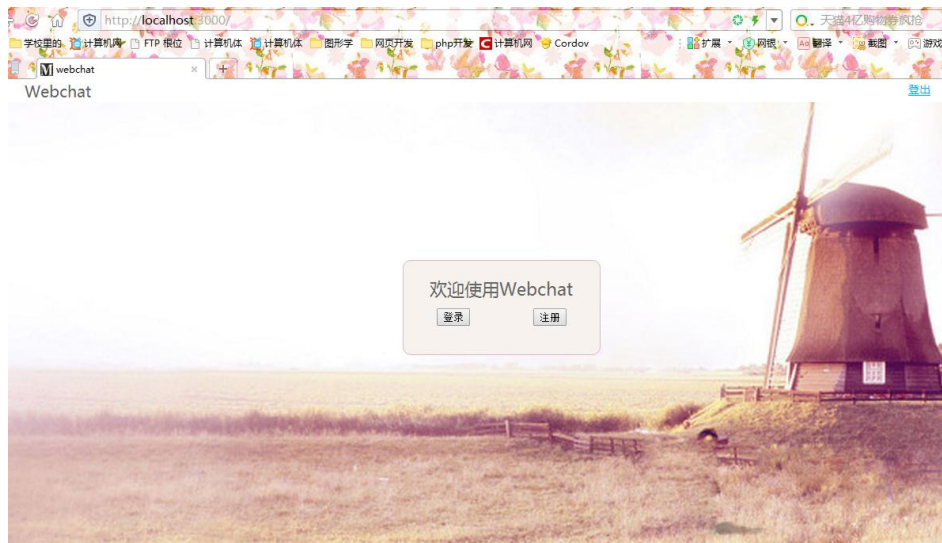
C:\Users\DELL>d:
D:\>cd gitproject
D:\gitproject>cd webchat
D:\gitproject\Webchat>npm start
> node-js-getting-started@0.0.1 start D:\gitproject\Webchat
> node index.js

```

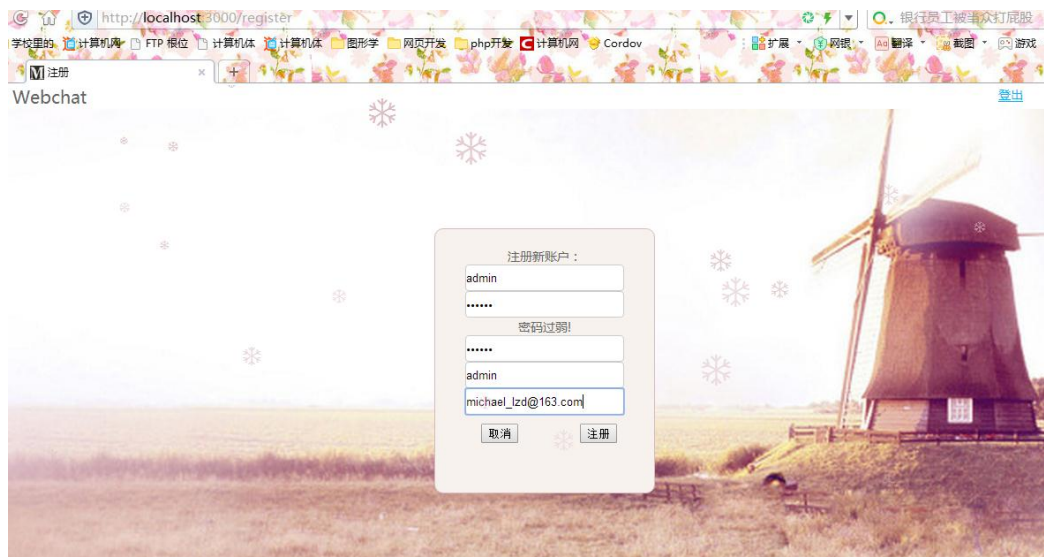
代表项目开始运行

三、项目操作流程

- a) 在浏览器中输入 localhost:3000 打开网站



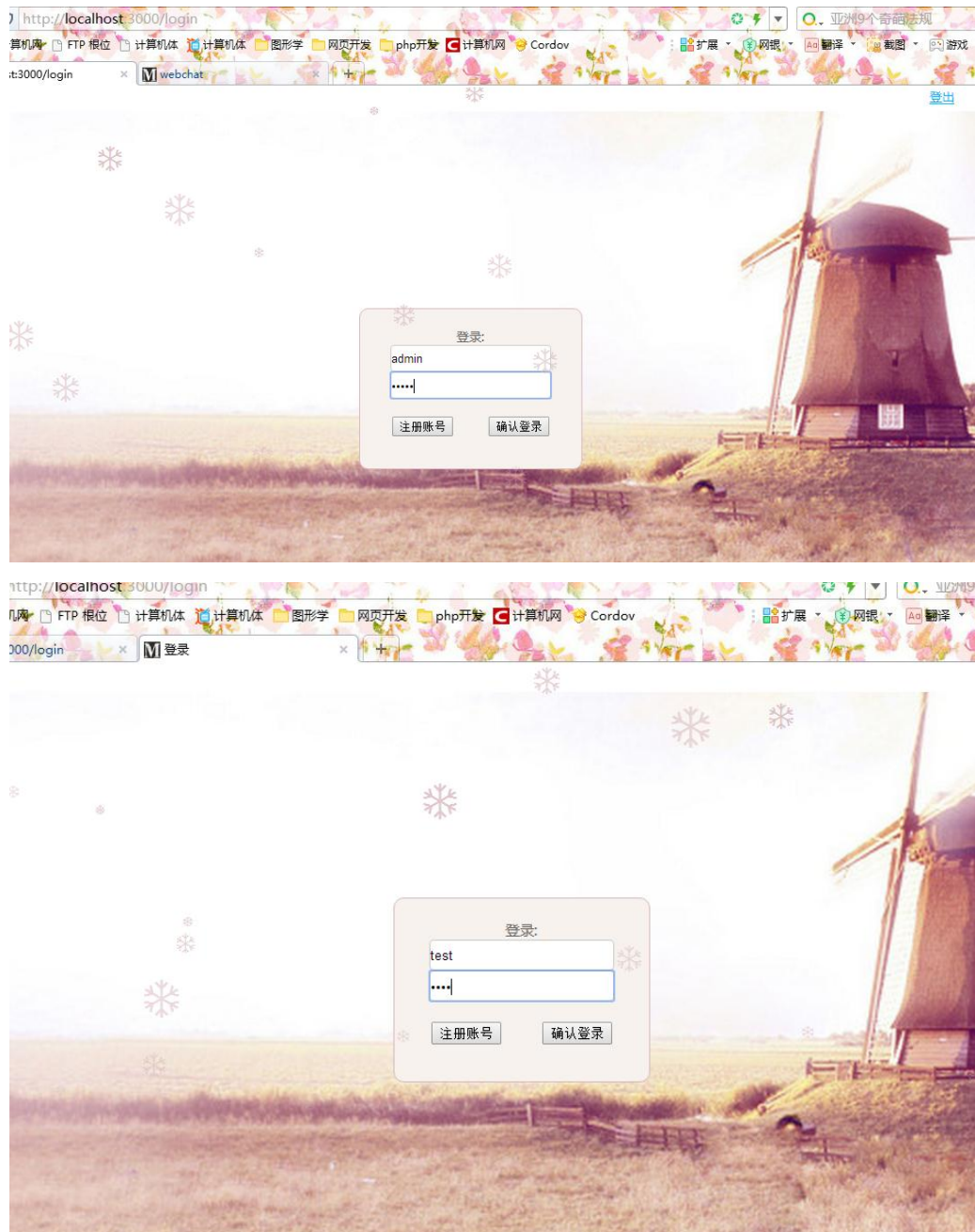
- b) 首先注册两个账号，admin 和 test



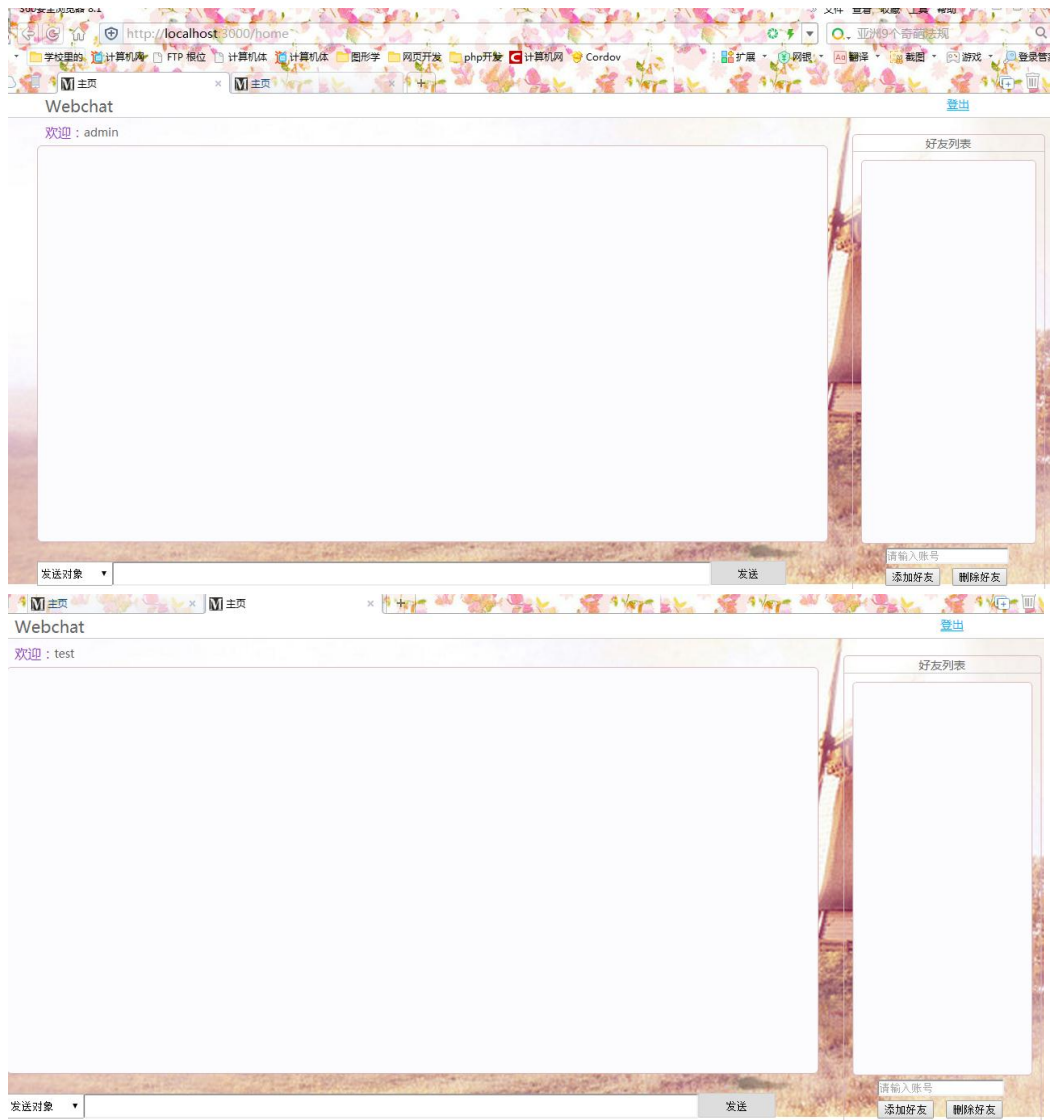
c) 注册后的数据库表信息

```
mysql> select * from user;
+-----+-----+-----+-----+
| username | password | nickname | email |
+-----+-----+-----+-----+
| admin    | 123456   | admin    | michael_lzd@163.com |
| test     | 123456   | test     | michael_lzd@163.com |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

d) 注册后账户会直接进入主页，在这里我们通过登录界面进入主页



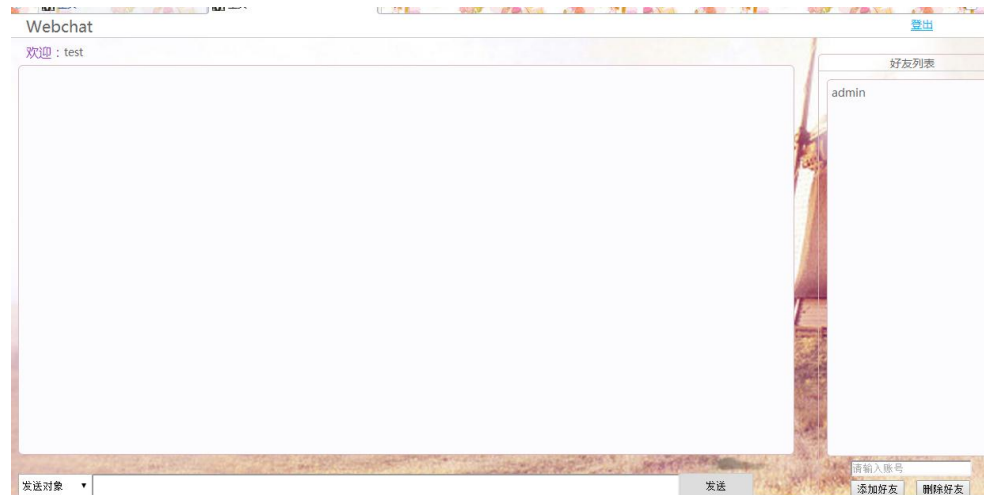
e) 登录后的界面如下图



f) 进行好友添加，在 **admin** 账户上添加 **test** 为好友



对应的 test 界面



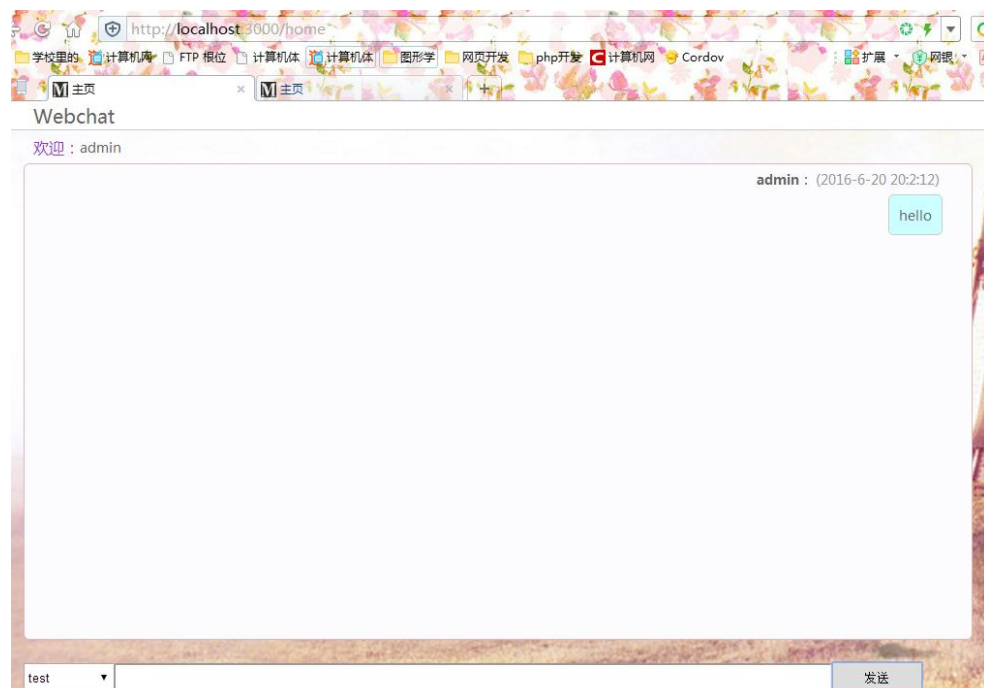
g) Friendship 数据库信息

```
mysql> select * from firendship;
+-----+-----+
| own_username | other_username |
+-----+-----+
| admin        | test           |
| test         | admin          |
+-----+-----+
2 rows in set (0.00 sec)
```

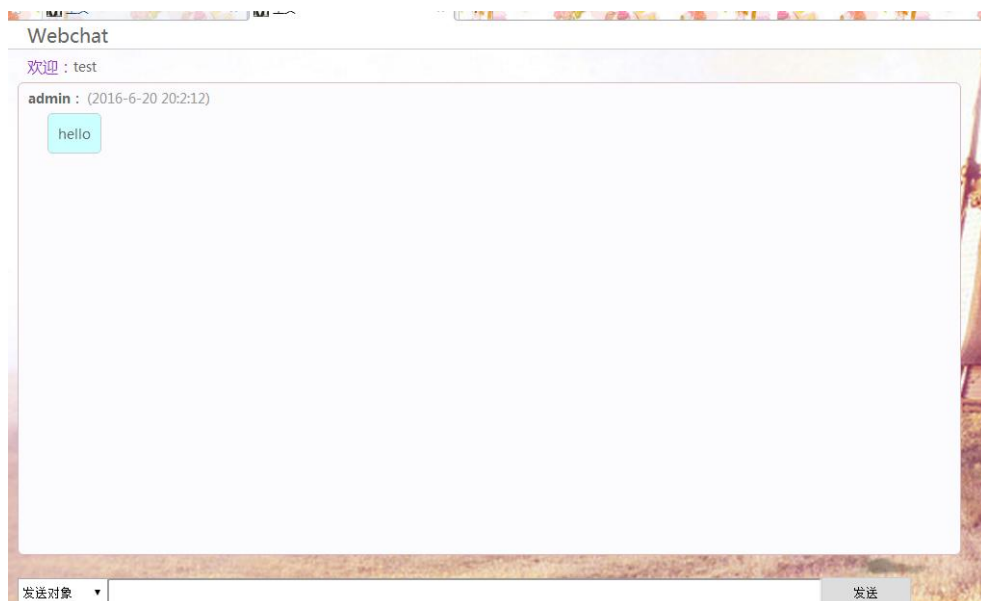
两个用户互为好友

h) 在线消息发送，admin 发送 hello 给 test

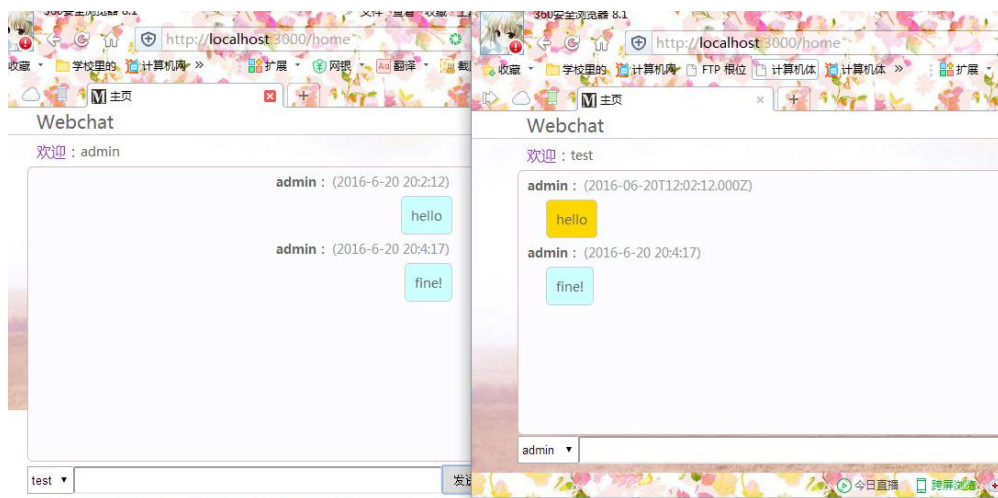
在好友列表中选择要发送的对象



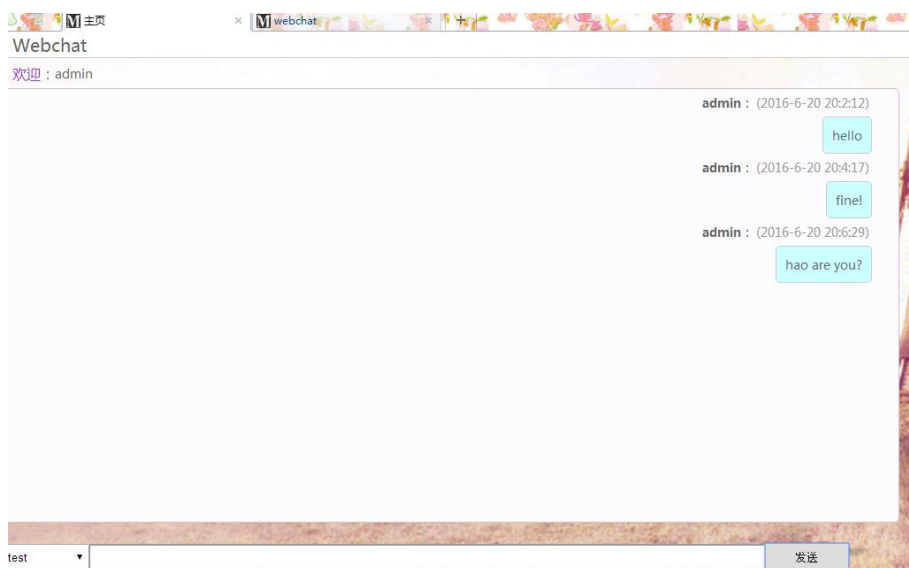
Test 用户收到的信息



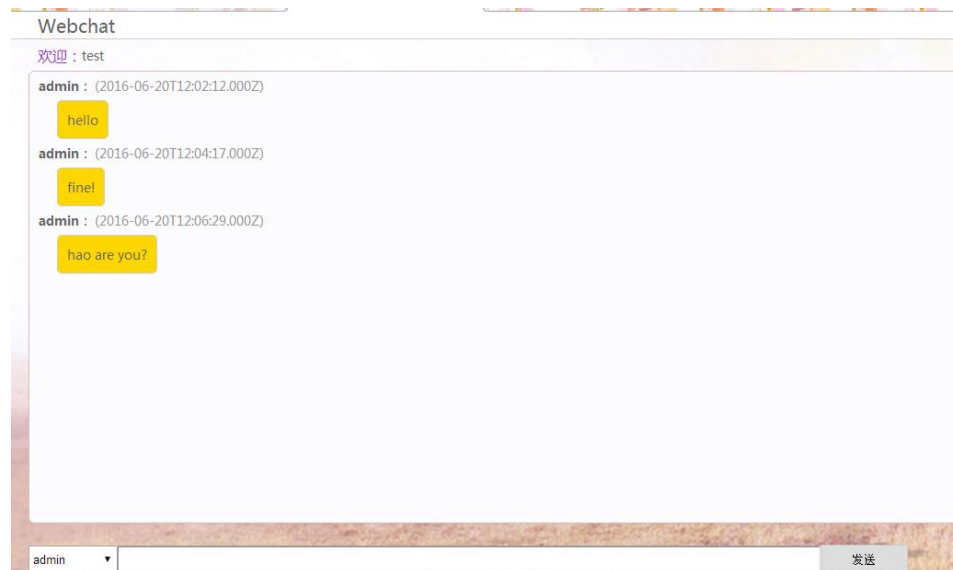
多发几条信息



i) 发送离线消息，若 test 不在线，使用 admin 发送信息



在 test 用户上线后，会看到 admin 发送的聊天信息



j) Message 表中的数据

```
mysql> select * from message;
+-----+-----+-----+-----+-----+-----+
| messageid | sender | receiver | send_time | ifread | content |
+-----+-----+-----+-----+-----+-----+
| 7 | admin | test | 2016-06-20 20:02:12 | 0 | hello |
| 8 | admin | test | 2016-06-20 20:04:17 | 0 | fine! |
| 9 | admin | test | 2016-06-20 20:06:29 | 0 | hao are you? |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

已保存了刚才发送的聊天记录。

四、 开发体会

整个大程用到的技术较多，自己一开始也走了很多弯路，使用了很多种方法，但都不尽人意，最后选择了 node.js 作为后端，使用 express4 框架开发的技术，相对来说，前端的 html5 以及 css3 都是比较简单的，而 javascript 和 jquery 则相对复杂，其中有很多的用法以及函数，现在也只是掌握了皮毛而已。而后端则是一个大难题，相对前端来说，后端逻辑比较混乱。

现在开发的产品只是一个简单的应用，实现了需求报告中的要求，而更多的功能，在未来一段时间内会逐步完善，希望能做成一个功能齐全的应用。