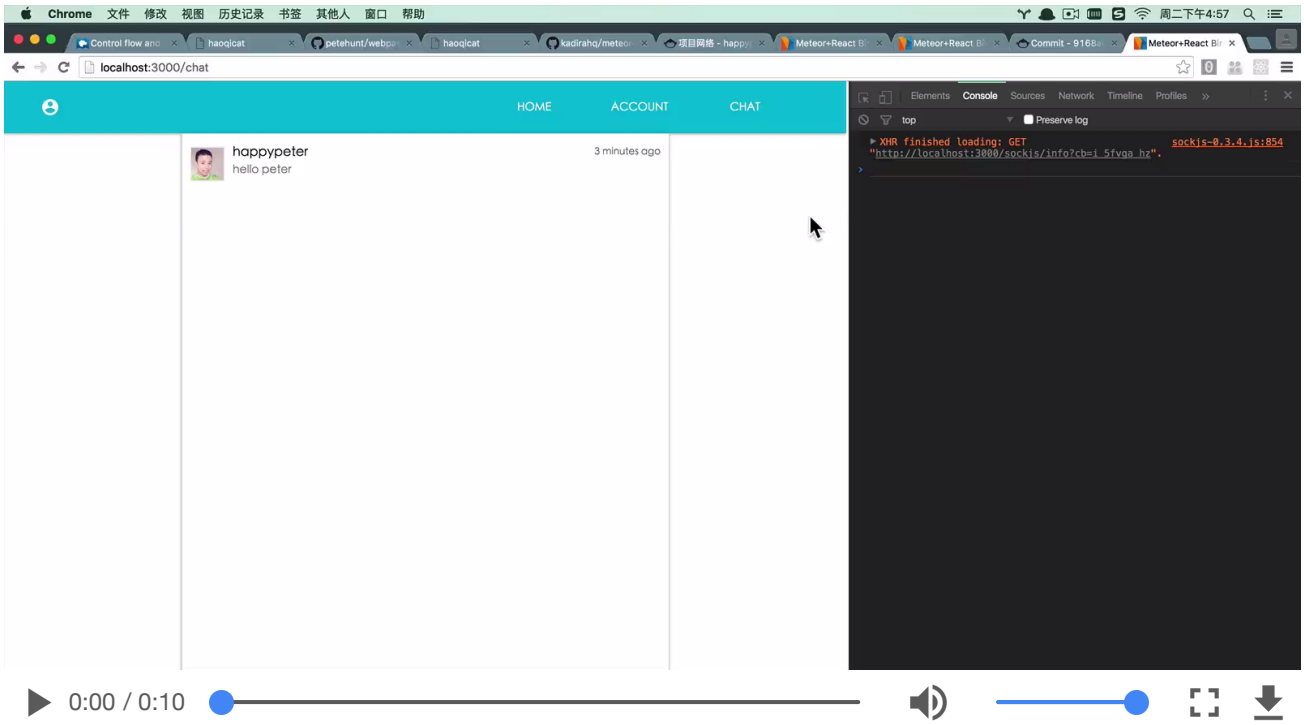

用户登录和登出

第六章：基本数据读写操作

三条线路上的数据读写

第七章：聊天室

提交聊天信息的表单

显示聊天信息

本节制作聊天记录的列表，当发送聊天信息之后，信息会在聊天表单的上方立即显示出来。同时聊天室中的其他人也可以看到这条信息。打开 Safari 浏览器，用另一个用户登录，也发送一条聊天信息，如果用户没有保存头像链接，就把用户名的首字母大写格式当做头像。另外，聊天信息发布日期的显示，我们用著名的 Moment.js 库处理。

创建信息列表 MessageList 组件

首先，我们在聊天室页面的空白处添加一个留言列表组件，所以到聊天室的 Chat 组件的 `render` 方法中，使用 MessageList 组件：

```
<MessageList messages={this.props.messages} />
```

并把聊天信息传递给 MessageList 组件的 `messages` 属性，当然这个聊天信息是动态的，这就涉及到前面介绍的 Meteor 的 Publish 和 Subscribe 机制了。

```
export default createContainer(() => {
  Meteor.subscribe("messages");
  return {
    messages: Messages.find({}, {sort: {createdAt: 1}}).fetch()
  };
}, Radium(Chat));
```

我们先在 `createContainer` 容器内订阅服务器端发布的数据集 `messages`，订阅之后，服务器端就会把数据集发送到客户端，然后通过下面代码：

```
Messages.find({}, {sort: {createdAt: 1}}).fetch()
```

获取传送到客户端的数据集合 `messages` 中的所有数据，并且这些数据按照创建时间排序。上述语句会返回一个数组。另外，还需要在 `Chat.jsx` 文件，添加一行代码：

```
import { Messages } from '../api/messages.js';
```

这样才能在 Chat 组件中访问 `Messages` 对象。

接下来，修改 `messages.js` 文件：

```
if (Meteor.isServer) {
  Meteor.publish('messages', function() {
    return Messages.find();
```

```
    });
  }
```

这里稍微提一下，Messages.find 接口若不添加任何参数，就把 `messages` 集合中的所有数据都导出来了，也可以通过添加参数，规定特定的记录之中哪个字段要导出，哪个字段不要导出。

查看更改：从数据库获取聊天信息

下面我们就要定义 MessageList 组件了，把它放到 `messages/MessageList.jsx` 文件中，显示所有的聊天信息。代码中，使用了 Lodash.js 库提供的 map 接口，把每条信息的内容传递给 Message 组件，并返回整个信息列表。其中 Message 组件的 `key` 属性有点儿特殊，这是 React 规范要求的，每个 Message 组件有了一个独一无二的标识符之后呢，能提高 React 的执行效率，请参考文档 React Keys。

在这再感慨一下，React 组件一级一级嵌套的形式，每一级通过组件属性传递信息，写出的代码确实是非常简单，容易阅读。

查看更改：创建 MessageList 组件

创建 Message 组件

现在，只需要创建 Message 组件，就能显示聊天信息了。在 `imports/ui/message/` 目录中，新建 `Message.jsx` 文件，添加一些代码。

查看更改：创建 Message 组件

首先显示一个头像，然后显示信息发送者，接下来显示信息创建日期，日期不是直接显示出来，而是用到了 Momment.js 库提供的 fromNow 接口对日期显示格式做了一下修改。最后是显示发送的聊天信息了。

对于信息发送者的头像，我们稍微做了一下处理，就是如果用户因为各种原因，他的头像链接在数据库中并没有存储，那就取他用户名的首字母大写格式作为头像。另外，还使用了一个新的 Radium 包提供的特性：

```
<div style={[styles.img, styles.avatar]}>
```

通过给组件的 `style` 属性传递一个样式对象数组，数组中的样式对象会自动合并，以此达到修改组件样式的目的。参考 Radium 的官方文档 **Modifiers** 部分的内容。

最后，我们需要安装 Moment.js 提供的 node 包 moment，安装命令如下：

```
npm install --save moment
```

所有代码修改完成之后，到浏览器聊天室页面测试一下，在输入框中输入 `hello bb`，这条信息马上显示出来了，信息发布日期的格式也正确，说明 Moment.js 库生效了。本节要实现的效果就完成了。