

```

    },
9    onmessage : function(message) {
        var messagesNode = document.getElementById(message
            .data.split(":")[0].concat(" messages"));
11    var messageNode = document.createElement('div');
        messageNode.appendChild(document.createTextNode(
            message.data));
13    messagesNode.appendChild(messageNode);
        processMessage(message.data.split(":")[0], message
            .data.split(":")[1]);
15
    },
17    onerror : function(error) {},
        onclose : function() {leave();}
19 });

```

The code above takes the received message, splits the text and extracts the sender address and message, and then adds the message to the appropriate chat box and sends the message to be processed via the processMessage function which calls a servlet via AJAX.

To token is passed to the page as a parameter when the page is loaded from an CreateChannelServlet servlet, which is called when the user decided to go from the homescreen to the chat screen. This servlet uses the Google user service (provided as part of the App Engine API) to get the email address of the logged in user, and use this to generate the token for the channel. This is added to the request as a parameter and sent to the chat.jsp page where it is used as above to open the channel.

```

1  UserService userService = UserServiceFactory.getUserService();
   User user = userService.getCurrentUser();
3  ChannelService channelService = ChannelServiceFactory
       .getChannelService();
5  String token = channelService.createChannel(user.getEmail().trim()
       );

```

Messages can then be sent to this user simply by using the email address as the destination when creating a ChannelMessage object (the Channels API equivalent of a UDP packet):

```

1  ChannelService channelService = ChannelServiceFactory
       .getChannelService();
3
   ChannelMessage toSend = new ChannelMessage(destEmail, userEmail
5       + ":" + output);
   channelService.sendMessage(toSend);

```