For Meteor 1.0.5, use

```
Template.<yourTemplate>.onCreated(function() { ... })
```

instead of

```
Template.<yourTemplate>.created = ...
```

in step 4 of Meteor #20: Verify an Email with Meteor Accounts.

The demo below works fine under Meteor 1.0.5.

verifyEmail.html:

```
<head>
  <title>Verify Email</title>
</head>

<body>
  <h1>Sending Email Demo</h1>
  {{> loginButtons}}
</body>
```

verifyEmail.js: (replace user, password and server value for your case)

```
if (Meteor.isClient) {
  Template.SendAnyEmail.onCreated(function() {
    if (Accounts._verifyEmailToken) {
      Accounts.verifyEmail(Accounts._verifyEmailToken, function(err) {
        if (err != null) {
          if (err.message = 'Verify email link expired [403]') {
            console.log('Sorry this verification link has expired.')
          }
        } else {
          console.log('Thank you! Your email address has been confirmed.')
        }
      });
    }
  });
}

if (Meteor.isServer) {
  Meteor.startup(function () {
    smtp = {
      username: 'user@example.com',
      password: 'password',
      server: 'mail.example.com',
      port: 465
    };
```

```
    process.env.MAIL_URL = 'smtp://' + encodeURIComponent(smtp.username) + ':' +
encodeURIComponent(smtp.password) + '@' + encodeURIComponent(smtp.server) + ':' +
smtp.port;

    Accounts.emailTemplates = {
      from: 'Administrator <user@example.com>',
      siteName: 'YourSite',
      verifyEmail: {
        subject: function(user) {
          return 'Verification email from Example.com';
        },
        text: function(user, url) {
          return 'Hi,\n' +
            'Please open the link below to verify your account on Example.com:\n' + url;
        }
      }
    };
  });

  Accounts.onCreateUser(function(options, user) {
    Meteor.setTimeout(function() {
      Accounts.sendVerificationEmail(user._id);
    }, 2 * 1000);
    return user;
  });
}
```