



## Managing Users

John Fitzpatrick

## Table of Contents

Lab Overview and Requirements .....	3
Overview .....	3
Lab Environment .....	3
Requirements .....	3
Section 1. User Management.....	4
1.1 Create a Session & Set Alias .....	4
1.2 Users and Accounts .....	4
1.2.1 List Users in the Account.....	4
1.2.2 Add a User & Set Observer Privileges .....	6
1.2.3 Verify Account.....	8
1.2.4 Add/Delete Further Permissions.....	9
1.2.5 Remove Privileges .....	11

## Lab Overview and Requirements

### Overview

This lab introduces some of the RightScale Governance Controls via the API. The tasks you will perform include

- Reporting
- User & Account Management
- Permissions
- Listing Deployments
- Listing Servers within a Deployment

### Lab Environment

You will launch an 'API Sender' server in AWS EC2 cloud, and SSH into to from your desktop. All API commands will then be issued from this server.

### Requirements

- A currently supported Browser. See <http://tinyurl.com/cl8p4mh>
- Java must be enabled in the browser.
- Unrestricted Internet Access. In particular, you must be able to browse to [rightscales.com](http://rightscales.com) on TCP port 80 (HTTP).
- You must have TCP port 22 (SSH) open so you can SSH into remote servers
- RightScale account with valid cloud credentials, and the following user role privileges - '**designer**', '**actor**', '**library**', '**server\_login**', '**security\_manager**'
- You must also be familiar with [vi](#) editor

## Section 1. User Management

This lab introduces some User Management functions. The tasks you will perform include

- User & Account Management
- Permissions

### 1.1 Create a Session & Set Alias

Before you start, you need to create a Session.

1. Run the script `auth-bash.sh` and supply your password when prompted

```
[api]# /opt/api/3tier/auth-bash.sh
```

2. Now set up an alias to make it easier to invoke an API command from the command line.

```
[api]# alias mycurl='curl -s -H X-API-Version:1.5 -b ~/mycookie '
```

### 1.2 Users and Accounts

See: <http://reference.rightscale.com/api1.5/resources/ResourceUsers.html>

#### 1.2.1 List Users in the Account

3. Run the following command to list the users who have access to the Training Account you're using

```
[api]# mycurl -X GET https://us-3.rightscale.com/api/users.xml
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user>
    <company>RightScale</company>
    <actions></actions>
    <created_at>2010/12/06 22:07:49 +0000</created_at>
    <phone></phone>
    <updated_at>2013/03/06 00:28:40 +0000</updated_at>
    <email>student04@rightscale.com</email>
    <links>
      <link href="/api/users/32746" rel="self"/>
    </links>
    <first_name>RightScale Student</first_name>
    <last_name>03</last_name>
  </user>
  <user>
    <company>RightScale Inc.</company>
    <actions></actions>
    <created_at>2010/12/28 22:54:17 +0000</created_at>
    <phone></phone>
    <updated_at>2013/03/06 00:29:04 +0000</updated_at>
    <email>trainer2@rightscale.com</email>
    <links>
      <link href="/api/users/33483" rel="self"/>
    </links>
    <first_name>RS Trainer</first_name>
    <last_name>2</last_name>
  </user>
</users>
```

Note each user has a unique **'self'** href.

```
<link href="/api/users/33483" rel="self"/>
```

4. Now use the **show** method on the **users** resource to see more detail on a specific user, i.e. do an **HTTP Get** on the **self** href

```
# mycurl -X GET https://us-3.rightscale.com/api/users/33483.xml
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <company>RightScale Inc.</company>
  <actions></actions>
  <created_at>2010/12/28 22:54:17 +0000</created_at>
  <phone></phone>
  <updated_at>2013/03/06 00:29:04 +0000</updated_at>
  <email>trainer2@rightscale.com</email>
  <links>
    <link href="/api/users/33483" rel="self"/>
  </links>
  <first_name>RS Trainer</first_name>
  <last_name>2</last_name>
</user>
```

### 1.2.2 Add a User & Set Observer Privileges

5. Please take a few minutes to read the important information in the API Reference docs on creating users

<http://reference.rightscale.com/api1.5/resources/ResourceUsers.html#create>

6. Edit the bash script `User-and-Permission-Add.sh`

This script will invoke the API Request to add a user to the account, and then use the User ID returned to assign `observer` permissions on the account.

**Note:** The Observer permission **MUST** be added first before any other role can be set. Further, when deleting a user, it must be the last role that is deleted.

```
[api]# cd /opt/api/users
[api]# vim 01_User-and-Permission-Add.sh
```

Populate the following variables with dummy values (do not use your own account details)

- `COMPANYNAME`
- `EMAIL`
- `FIRSTNAME`
- `LASTNAME`

- **PHONE**
- **PASSWORD**

The script should look like the following

```
#!/bin/bash -e

COMPANYNAME="MyCompanyName"
EMAIL="jdoe@example.com"
FIRSTNAME="Myfirstname"
LASTNAME="Mylastname"
PHONE="0123456789"
PASSWORD="P@ssword123"

curl -i -H X-API-Version:1.5 -b ~/mycookie -X POST \
-d user[company]=$COMPANYNAME \
-d user[email]=$EMAIL \
-d user[first_name]=$FIRSTNAME \
-d user[last_name]=$LASTNAME \
-d user[password]=$PASSWORD \
-d user[phone]=$PHONE \
https://us-3.rightscale.com/api/users \
| tee output/User-and-Permission-Add.out

#Set the USERID to that of the user just created
USERID=`grep Location output/User-and-Permission-Add.out |cut -c 22-|tr -d '\r'`

#Set the permissions for user
curl -i -H X-API-Version:1.5 -b ~/mycookie -X POST \
-d permission[role_title]=observer \
-d permission[user_href]=/api/users/$USERID \
https://us-3.rightscale.com/api/permissions \
| tee -a output/User-and-Permission-Add.out
```

## 7. Run the script

```
[api]# ./01_User-and-Permission-Add.sh
HTTP/1.1 201 Created
Server: nginx/1.0.14
Date: Thu, 21 Mar 2013 19:43:51 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Status: 201 Created
Location: /api/users/67485
X-Runtime: 656
X-Request-Uid: 365d242406ad4b46a404ab09ccc01e92
Set-Cookie:
Cache-Control: no-cache

HTTP/1.1 201 Created
Server: nginx/1.0.14
Date: Thu, 21 Mar 2013 19:43:51 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Status: 201 Created
Location: /api/permissions/837548
X-Runtime: 236
X-Request-Uid: 6c9e345c6f94419a9d73f250bdabd887
Set-Cookie:
Cache-Control: no-cache
```

### 1.2.3 Verify Account

8. Run the following script (replacing the User ID) to check the user



```
[api]# mycurl -X GET https://us-3.rightscale.com/api/users/67485.xml
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <company>MyCompanyName</company>
  <actions></actions>
  <created_at>2013/03/20 17:45:42 +0000</created_at>
  <phone>0123456789</phone>
  <updated_at>2013/03/20 17:45:42 +0000</updated_at>
  <email>name@example.com</email>
  <links>
    <link href="/api/users/67485" rel="self"/>
  </links>
  <first_name>Myfirstname</first_name>
  <last_name>Mylastname</last_name>
</user>
```

## 9. Now browse to 'Settings > Account Settings > Users' the Dashboard

Info	Plan	Account Groups	Preferences	Users	Invitations	Clouds	Administered Clouds	Server Defaults	Access Controls	API Credentials
Authorization Logging										
<input type="button" value="edit"/>										
<b>Account users and their permissions</b>										
[redacted]		admin, actor, observer, publisher, designer, billing, server_login, library, security_manager							<input type="button" value="x"/>	
[redacted]m		admin, actor, observer, publisher, designer, billing, server_login, library, security_manager							<input type="button" value="x"/>	
name@example.com		observer							<input type="button" value="x"/>	
student04@rightscale.com		admin, actor, observer, designer, billing, server_login, library, security_manager							<input type="button" value="x"/>	
trainer2@rightscale.com		admin, actor, observer, publisher, designer, server_login, library							<input type="button" value="x"/>	
trainer@rightscale.com		admin, actor, observer, publisher, designer, billing, server_login, library, security_manager							<input type="button" value="x"/>	
<input type="button" value="edit"/>										

### 1.2.4 Add/Delete Further Permissions

The script you ran earlier added **observer** privileges. Now you will add the User **admin** and **designer** privileges.

## 10. Edit the script `Permissions-AdminDesigner-Add.sh`

```
[api]# vim 02_Permissions-AdminDesigner-Add.sh
```

This script will add the two permissions to the user

## 11. Update the script to add the ID for the User you just created.

The script should look like the following

```
#!/bin/bash -e

USERID="67485"

curl -i -H X-API-Version:1.5 -b ~/mycookie -X POST \
-d permission[role_title]=admin \
-d permission[user_href]=/api/users/$USERID \
https://us-3.rightscale.com/api/permissions \
| tee output/Permissions-AdminDesigner-Add.out

curl -i -H X-API-Version:1.5 -b ~/mycookie -X POST \
-d permission[role_title]=designer \
-d permission[user_href]=/api/users/$USERID \
https://us-3.rightscale.com/api/permissions \
| tee output/Permissions-AdminDesigner-Add.out
```

12. Now run the script

```
[api]# ./02_Permissions-AdminDesigner-Add.sh
HTTP/1.1 201 Created
Server: nginx/1.0.14
Date: Fri, 22 Mar 2013 10:30:39 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Status: 201 Created
Location: /api/permissions/837760
X-Runtime: 285
X-Request-Uuid: 2eaaa71c322b48098af6ba657c81243b
Set-Cookie:
Cache-Control: no-cache

HTTP/1.1 201 Created
Server: nginx/1.0.14
Date: Fri, 22 Mar 2013 10:30:39 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Status: 201 Created
Location: /api/permissions/837761
X-Runtime: 214
X-Request-Uuid: ce3d91a8014a471399289cf40734a419
Set-Cookie:
Cache-Control: no-cache
```

13. Now browse to 'Settings > Account Settings > Users' the Dashboard again to verify the permissions.
14. Make a note of the permission numbers in the **Location** fields of the API Response, as you will need these in the following section

### 1.2.5 Remove Privileges

See: <http://reference.rightscale.com/api1.5/resources/ResourcePermissions.html#destroy>

Now you will remove the **admin** and **designer** privileges.

15. Edit the script **03\_Permissions-AdminDesigner-Delete.sh**

```
[api]# vim 03_Permissions-AdminDesigner-Delete.sh
```

The script should look like the following

```
#!/bin/bash -e

PERMISSION1="837760"
PERMISSION2="837761"

curl -i -H X-API-Version:1.5 -b ~/mycookie -X DELETE \
https://us-3.rightscale.com//api/permissions/$PERMISSION1 \
| tee output/Permissions-AdminDesigner-Delete.out

curl -i -H X-API-Version:1.5 -b ~/mycookie -X DELETE \
https://us-3.rightscale.com//api/permissions/$PERMISSION2 \
| tee -a output/Permissions-AdminDesigner-Delete.out
```

16. Now run the script

```
[api]# ./03_Permissions-AdminDesigner-Delete.sh
...
Status: 204 No Content
...

...
Status: 204 No Content
...
```

17. Now browse to 'Settings > Account Settings > Users' the Dashboard again to verify the permissions have been removed.