



restrooms




Installation

- 1.) Copy it to your htdocs directory and modify and call localhost/phpmyadmin to setup the database. For example, create the following database:

Databases

Create database 

Collation 

Database	Master replication	
<input type="checkbox"/> information_schema	✓ Replicated	 Check Privileges
<input type="checkbox"/> mysql	✓ Replicated	 Check Privileges
<input type="checkbox"/> performance_schema	✓ Replicated	 Check Privileges

- 2.) Import the file backend.sql:



The screenshot shows the phpMyAdmin interface for the 'localhost' server. The 'Import' tab is selected. The main area is titled 'Importing into the current server'. Under 'File to Import:', there is a note that files can be compressed (gzip) or uncompressed, and a compressed file's name must end in .[format].[compression], with an example of .sql.zip. Below this, there is a 'Browse your computer:' section with a 'Choose File' button, indicating 'No file chosen' and a '(Max: 50MiB)' limit. The 'Character set of the file:' is set to 'utf8'. Under 'Partial Import:', there is a checked checkbox for 'Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be good for transactions.)' and a 'Number of rows to skip, starting from the first row:' field set to '0'.

After the import you have the following tables:

☐ friends

☐ friend_invitations

☐ roommembers

☐ rooms

☐ room_invitations

☐ users

3.) Modify connect_db_settings.php:

```
define('DB_SERVER', "localhost");
```

```
define('DB_PASSWORD', "");
```

```
define('DB_NAME', "backend");
```

```
define('DB_USER', "root");
```

If you use “backend” as database please enter “backend” here as well.

Call <http://localhost/restrooms> - you will land on the login.php page:

localhost/restrooms/login.php

Admin username: admin Password: admin email address: password:

Password and username are already given: admin, admin

After logging in, you will get shown the default view, on the left side:

Action	id	email	title	first name	last name	md5 password	sex
<div>Edit</div> <div>Delete</div>	198610	oliverbach@gmail.com	Dr.	Oliver	Bachmann	b6d767d2f8ed5d21a44b0e5886680cb9	1
<div>Edit</div> <div>Delete</div>	198618	22	33	44	55	b6d767d2f8ed5d21a44b0e5886680cb9	0
<div>Edit</div> <div>Delete</div>	198619	33	44	55	66	b6d767d2f8ed5d21a44b0e5886680cb9	0
<div>Edit</div> <div>Delete</div>	198615	11	22	33	44	b6d767d2f8ed5d21a44b0e5886680cb9	0

And on the right side, for every entry, you have the following buttons:

friend list		joined rooms
<div>Show friends</div> <div>Show friend requests</div> <div>Send friend request</div> <div>Search users demo</div> <div>Add friends</div> <div>Remove friends</div> 11	<div>Create room</div> <div>Join (all) rooms</div>	<div>Show only joined rooms</div> <div>User room list</div> <div>User and friends room list</div> <div>Invite other users to rooms</div> <div>Show invitations to room</div>
<div>Show friends</div> <div>Show friend requests</div> <div>Send friend request</div> <div>Search users demo</div> <div>Add friends</div> <div>Remove friends</div>	<div>Create room</div> <div>Join (all) rooms</div>	<div>Show only joined rooms</div> <div>User room list</div> <div>User and friends room list</div> <div>Invite other users to rooms</div> <div>Show invitations to room</div> test - 2
<div>Show friends</div> <div>Show friend requests</div> <div>Send friend request</div> <div>Search users demo</div> <div>Add friends</div> <div>Remove friends</div>	<div>Create room</div> <div>Join (all) rooms</div>	<div>Show only joined rooms</div> <div>User room list</div> <div>User and friends room list</div> <div>Invite other users to rooms</div> <div>Show invitations to room</div> test - 7
<div>Show friends</div> <div>Show friend requests</div> <div>Send friend request</div> <div>Search users demo</div> <div>Add friends</div> <div>Remove friends</div> oliverbach@gmail.com	<div>Create room</div> <div>Join (all) rooms</div>	<div>Show only joined rooms</div> <div>User room list</div> <div>User and friends room list</div> <div>Invite other users to rooms</div> <div>Show invitations to room</div> test - 7, room not found (198653)

1. EDIT

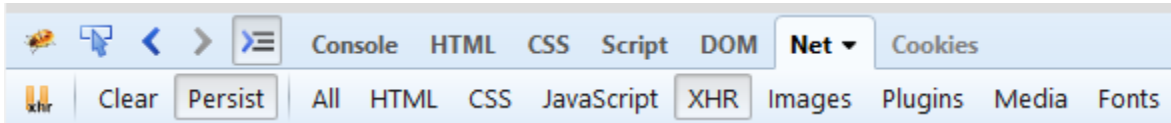
When you click on edit, the following input fields will appear at the bottom of the page:

avatar_url: sex:

This is just an example. Here you can modify the field avatar_url and sex.

What happens when you click on Save?

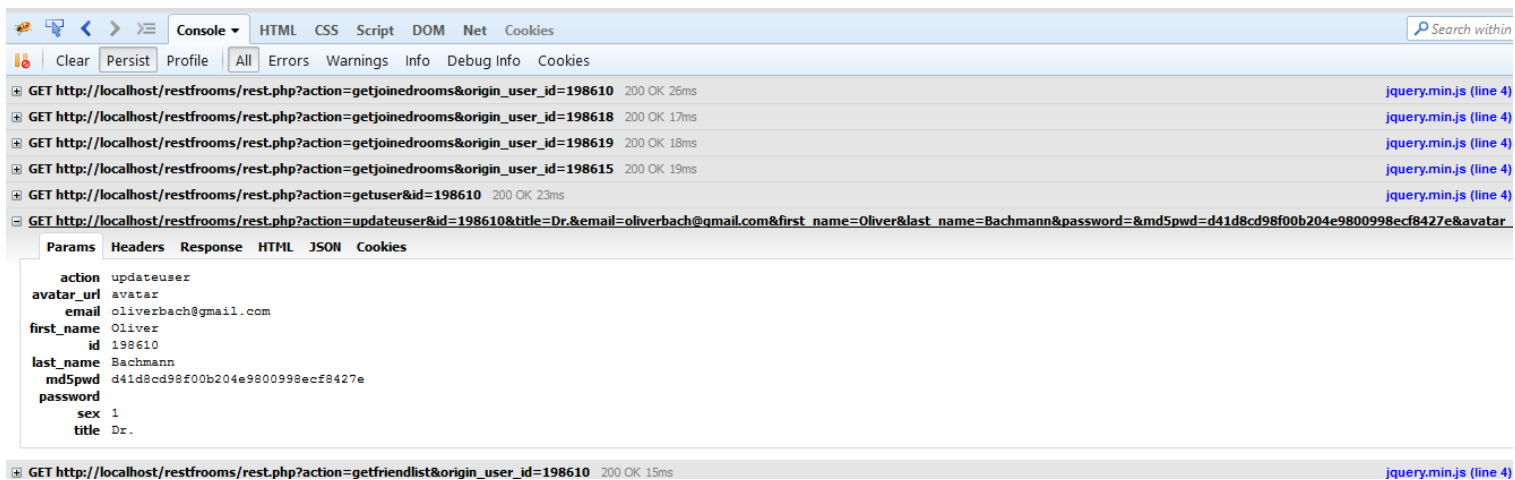
Let's open Firebug, the Firefox debugger. Alternatively you can of course also use the Chrome Inspector or the Safari Developer Tool. To see the following, make sure that "Persist" in Firebug is turned on:



Then, when you click on Save:

avatar_url: sex:

...you will see the following request:

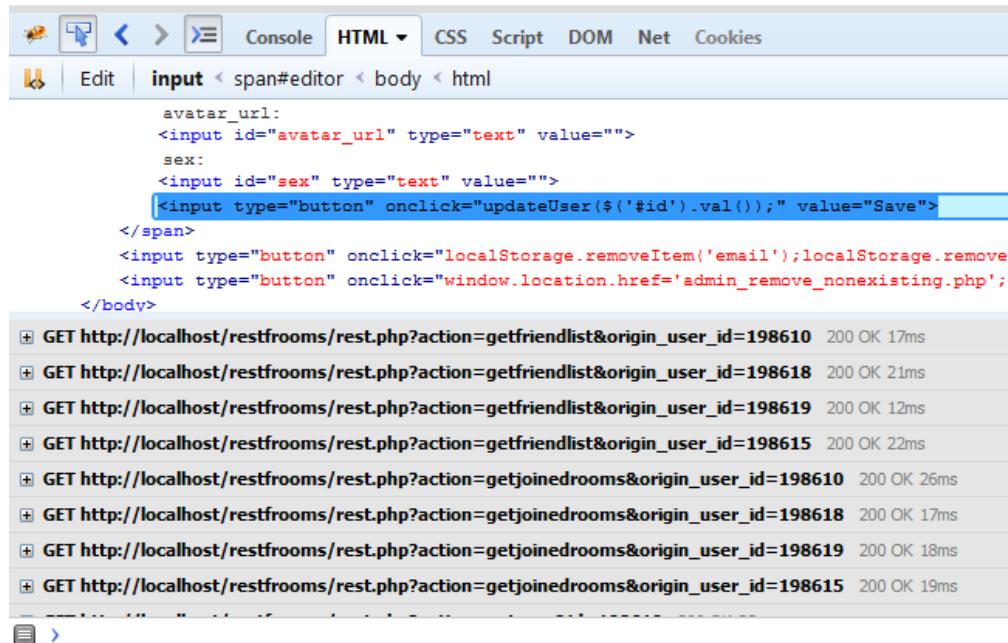


The fields:

action = updateuser, avatar_url = avatar, email = oliverbach@gmail.com, first_name=Oliver, id=198610, last_name = Bachmann, md5pwd = d41d8cd98f00b204e9800998ecf8427e, password = (empty), sex = 1, title = Dr.

Use Firebug to look at the button save

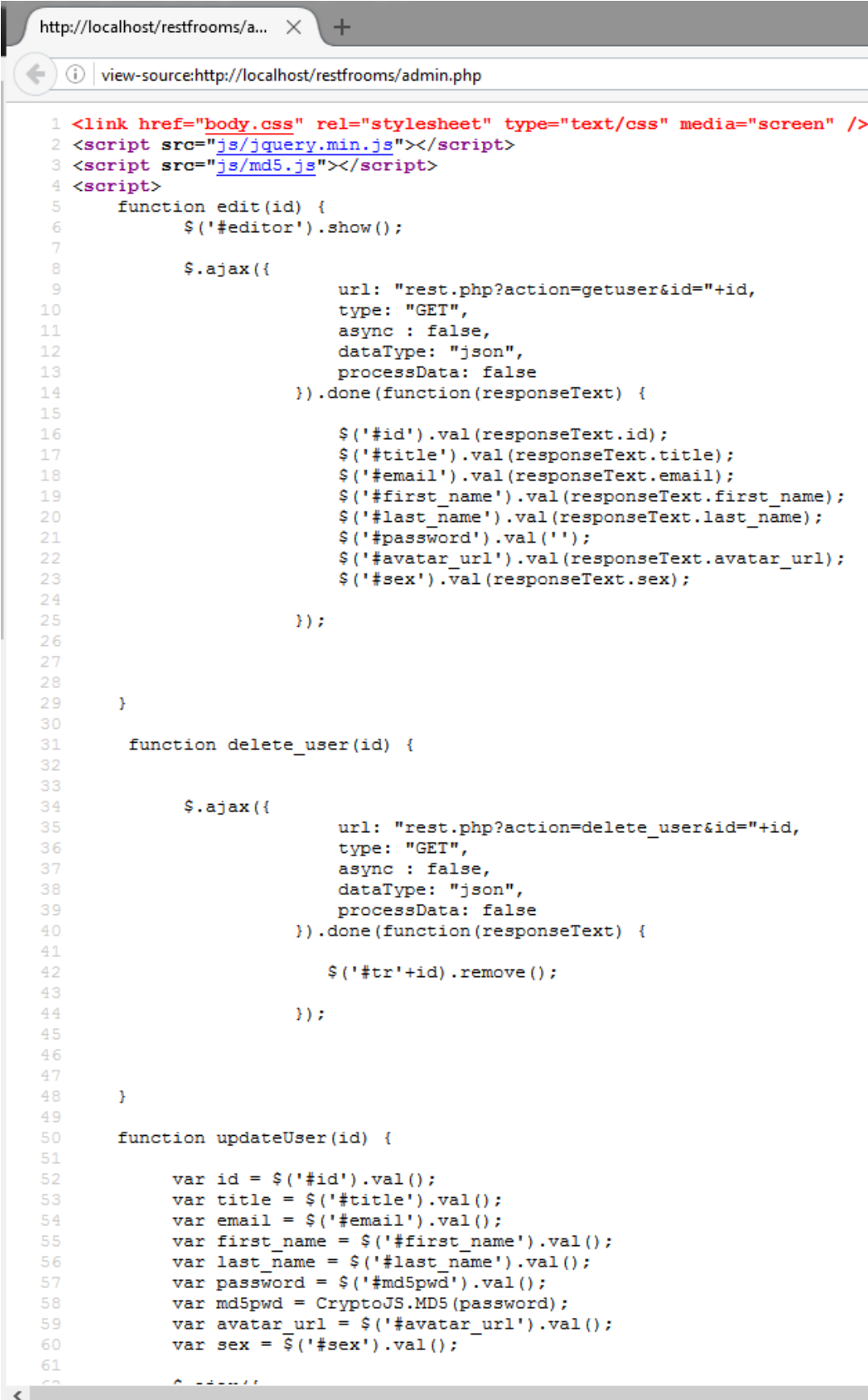
avatar_url: sex:



Now we know what is happening here: updateUser() gets called.

We can simply click Ctrl-U or on View Source in the menu, to look at the Sourcecode.

First look at the sourcecode



The screenshot shows a web browser window with the address bar displaying `http://localhost/restrooms/a...`. The page title is `view-source:http://localhost/restrooms/admin.php`. The source code is displayed in a monospaced font with syntax highlighting. It includes a link to `body.css`, and scripts for `jquery.min.js` and `md5.js`. The main code consists of three functions: `edit(id)`, `delete_user(id)`, and `updateUser(id)`. The `edit` function uses `$.ajax` to fetch user data from `rest.php?action=getuser&id="+id` and updates the form fields. The `delete_user` function uses `$.ajax` to delete a user from `rest.php?action=delete_user&id="+id` and removes the corresponding row from the table. The `updateUser` function collects form data, hashes the password using `CryptoJS.MD5`, and prepares data for an update request.

```
1 <link href="body.css" rel="stylesheet" type="text/css" media="screen" />
2 <script src="js/jquery.min.js"></script>
3 <script src="js/md5.js"></script>
4 <script>
5     function edit(id) {
6         $('#editor').show();
7
8         $.ajax({
9             url: "rest.php?action=getuser&id="+id,
10            type: "GET",
11            async : false,
12            dataType: "json",
13            processData: false
14        }).done(function(responseText) {
15
16            $('#id').val(responseText.id);
17            $('#title').val(responseText.title);
18            $('#email').val(responseText.email);
19            $('#first_name').val(responseText.first_name);
20            $('#last_name').val(responseText.last_name);
21            $('#password').val('');
22            $('#avatar_url').val(responseText.avatar_url);
23            $('#sex').val(responseText.sex);
24
25        });
26
27
28    }
29
30    function delete_user(id) {
31
32
33        $.ajax({
34            url: "rest.php?action=delete_user&id="+id,
35            type: "GET",
36            async : false,
37            dataType: "json",
38            processData: false
39        }).done(function(responseText) {
40
41            $('#tr'+id).remove();
42
43        });
44
45
46    }
47
48    function updateUser(id) {
49
50        var id = $('#id').val();
51        var title = $('#title').val();
52        var email = $('#email').val();
53        var first_name = $('#first_name').val();
54        var last_name = $('#last_name').val();
55        var password = $('#md5pwd').val();
56        var md5pwd = CryptoJS.MD5(password);
57        var avatar_url = $('#avatar_url').val();
58        var sex = $('#sex').val();
59
60        // Prepare data for update request
61    }
```

So it's time to look at rest.php

rest.php

```
66 include("connect_db.php");
67 $action="";
68 if (isset($_GET['action'])) $action=$_GET['action'];
69 if (strcmp($action, "adduser")==0) {
70     $email="";
71     $title="";
72     $first_name="";
73     $last_name="";
74     $md5pwd="";
75     $sex="";
76     if (isset($_GET['email'])) $email=$_GET['email'];
77     if (isset($_GET['title'])) $title=$_GET['title'];
78     if (isset($_GET['first_name'])) $first_name=$_GET['first_name'];
79     if (isset($_GET['last_name'])) $last_name=$_GET['last_name'];
80     if (isset($_GET['md5pwd'])) $md5pwd=$_GET['md5pwd'];
81     if (isset($_GET['sex'])) $sex=$_GET['sex'];
82     if (strlen($email)==0) die(json_encode(array("status" => 0, "msg" => "email not set")));
83     if (strlen($title)==0) die(json_encode(array("status" => 0, "msg" => "title not set")));
84     if (strlen($first_name)==0) die(json_encode(array("status" => 0, "msg" => "first_name not set")));
85     if (strlen($last_name)==0) die(json_encode(array("status" => 0, "msg" => "first_name not set")));
86     if (strlen($md5pwd)==0) die(json_encode(array("status" => 0, "msg" => "first_name not set")));
87     if (strlen($sex)==0) die(json_encode(array("status" => 0, "msg" => "first_name not set")));
88     $sql="SELECT * FROM users WHERE (email='".$email."')";
89     $query=DBi::$conn->query($sql) or die(DBi::$conn->error." ".__FILE__." line ".__LINE__.$sql);
90     if ($row=$query->fetch_assoc()) {
91         echo json_encode(array("status" => 0, "msg" => "email already exists".$sql));
92     } else {
93         $sql = "INSERT INTO users (email, title, first_name, last_name, md5pwd, sex) VALUES ('".$email."', '".$title."', '".$first_name."', '".$last_name."', '".$md5pwd."', '".$sex."')";
94         $query = DBi::$conn->query($sql) or die(json_encode(array("status" => 0, "error" => DBi::$conn->error." ".__FILE__." line ".__LINE__)));
95         echo json_encode(array("status" => 1, "msg" => "success"));
96     }
97 } else
98 if (strcmp($action, "addfriend")==0) {
99     $origin_user_id="";
100     if (isset($_GET['origin_user_id'])) $origin_user_id=$_GET['origin_user_id'];
101     if (strlen($origin_user_id)==0) die(json_encode(array("status" => 0, "msg" => "origin_user_id not set")));
102     $destination_user_id="";
103     if (isset($_GET['destination_user_id'])) $destination_user_id=$_GET['destination_user_id'];
104     if (strlen($destination_user_id)==0) die(json_encode(array("status" => 0, "msg" => "destination_user_id not set")));
105     $sql="SELECT * FROM friends WHERE (destination_user_id='".$destination_user_id."' AND (origin_user_id='".$origin_user_id."'))";
106     $query=DBi::$conn->query($sql) or die(DBi::$conn->error." ".__FILE__." line ".__LINE__.$sql);
107     if ($row=$query->fetch_assoc()) {
108         echo json_encode(array("status" => 0, "msg" => "Users are already friends"));
109     } else {
110         $sql="INSERT INTO friends (origin_user_id, destination_user_id) VALUES ('".$origin_user_id."', '".$destination_user_id."')";
111         $query=DBi::$conn->query($sql) or die(DBi::$conn->error." ".__FILE__." line ".__LINE__.$sql);
112         echo json_encode(array("status" => 1, "msg" => "Users are now friends"));
113     }
114 }
```

In rest.php we mainly have it to do with evaluating GET parameters, and mainly depending on the action parameter do different actions, which are updating fields using SQL statements.

You will steadily encounter strcmp (string comparison) functions.

When fields get returned, sometimes the fields get JOINED with a different table, for example the user who owns the group or similar things:

```
if (strcmp($action, "getroomlist")==0) {

    $owner_user_id="";
    if (isset($_GET['owner_user_id'])) {
        $owner_user_id=$_GET['owner_user_id'];
    }

    $joined_only="";
    if (isset($_GET['joined_only'])) {
        $joined_only=$_GET['joined_only'];
    }

    $getinvitationinfo=false;

    $invitation_by="";
    if (isset($_GET['invitation_by'])) {
        $invitation_by=$_GET['invitation_by'];
        $getinvitationinfo=true;
    }

    $friends="";
    if (isset($_GET['friends'])) {
        $friends=$_GET['friends'];
    }

    if ($getinvitationinfo==true) { // Check if all parameters are set
        if ( (strlen($invitation_by)==0) || (strlen($owner_user_id)==0) ) {
            die(json_encode(array("status" => 0, "msg" => "invitation_by or owner_user_id not set")));
        }
    }

    $counter=0;
    $roomlist=array();
    $sql="SELECT r.id as r_id, r.*,u.id as u_id, u.* FROM rooms r LEFT JOIN users u ON (r.owner_user_id=u.id)";
    if (strcmp($joined_only,"1")==0) {

        $sql.=" WHERE (owner_user_id=".$owner_user_id.")";
        $friendids=getFriendIDS($owner_user_id);
        for ($i=0;$i<sizeof($friendids);$i++) {
            $sql.=" OR (owner_user_id=".$friendids[$i].")";
        }
    }
} else if (strcmp($friends,"1")==0) {
    if (strlen($owner_user_id)>0) {
        $sql.=" WHERE ( (owner_user_id=".$owner_user_id.")";
    } else
    {
        $sql.="WHERE ( (false)";
    }
    $friendids=getFriendIDS($owner_user_id);
    for ($i=0;$i<sizeof($friendids);$i++) {
        $sql.=" OR (owner_user_id=".$friendids[$i].")";
    }
    $sql.=")";
} else {
    if (strlen($owner_user_id)>0) {
        $sql.=" WHERE (owner_user_id=".$owner_user_id.")";
    }
}
}
```


What functionality does the app have to offer?

1. Edit and delete

Action	id	email	title	first name	last name	md5 password
<div>Edit</div> <div>Delete</div>	198610	oliverbach@gmail.com	Dr.	Oliver	Bachmann	d41d8cd98f00b204e9800998ec

Console HTML CSS Script DOM Net Cookies

Edit input < td < tr#tr198610 < tbody < table < body < html

<body onload="getFriendList();getRoomList();">

<table>

<tbody>

<tr>

<tr id="tr198610">

<td>

<input type="button" onclick="edit(198610);" value="Edit">

<input type="button" onclick="delete_user(198610);" value="Delete">

</td>

When editing and deleting users the javascript calls edit and delete_user exist.

The functionality itself should be clear.

When a user gets deleted the following ajax request gets made:

+ GET http://localhost/restrooms/rest.php?action=getfriendlist&origin_user_id=198610 200 OK 16ms

+ GET http://localhost/restrooms/rest.php?action=getfriendlist&origin_user_id=198618 200 OK 21ms

+ GET http://localhost/restrooms/rest.php?action=getfriendlist&origin_user_id=198619 200 OK 24ms

+ GET http://localhost/restrooms/rest.php?action=getjoinedrooms&origin_user_id=198610 200 OK 16ms

+ GET http://localhost/restrooms/rest.php?action=getjoinedrooms&origin_user_id=198618 200 OK 13ms

+ GET http://localhost/restrooms/rest.php?action=getjoinedrooms&origin_user_id=198619 200 OK 15ms

- GET http://localhost/restrooms/rest.php?action=delete_user&id=198619 200 OK 17ms

Params Headers Response HTML JSON Cookies

{ "status": 1, "msg": "OK" }

As you can see, it's all ajax requests. When the status is 1, then also javascript will remove the field from the table, without reloading the table, just removing the <tr> that belongs to the removed entry.

Adding a new user

When a new user gets added, this also happens with an ajax request, the form gets loaded and is on register.php.

email	title	first name	last name	password	password repetition	sex
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	female ▾
<input type="button" value="Back"/>		<input type="button" value="save"/>				

Let's have a look at what happens when pressing save.



onclick is:

```
submit( $('#email').val(), $('#title').val(), $('#first_name').val(), $('#last_name').val(), $('#md5pwd').val(),  
$('#md5pwd2').val(), $('#sex').val() );
```

The function submit in register.php looks like this:

```
1 <link href="body.css" rel="stylesheet" type="text/css" media="screen" />  
2 <script src="js/jquery.min.js"></script>  
3 <script src="js/md5.js"></script>  
4  
5  
6 <script>  
7   function submit(email,title,first_name,last_name,md5pwd,md5pwd2,sex) {  
8     if (md5pwd!=md5pwd2) {  
9       $('#info').html('Passwords do not match');  
10      return;  
11     } else {  
12       $('#info').html('');  
13     }  
14  
15     md5pwd = CryptoJS.MD5(md5pwd);  
16  
17     $.ajax({  
18       url: "rest.php?action=adduser&email="+email+"&title="+title+"&first_name="+first_name+"&last_name="+last_name+"&md5pwd="+md5pwd+"&md5pwd2="+md5pwd2+"&sex="+sex,  
19       type: "GET",  
20       dataType: "json",  
21       processData: false  
22     }).done(function(responseText) {  
23  
24       if (responseText.status == 1) {  
25         window.location.href="index.php";  
26       } else {  
27         alert(responseText.msg);  
28       }  
29     });  
30  
31  
32   }
```

So, here we have it to do with the action adduser in register.php, because the action is "adduser".

Now how ajax calls work should be clear. Besides editing and deleting the provided functionality is also about friends:

Show friends	Show friend requests	Send friend request
Search users demo	Add friends	Remove friends

user
not found (198615)

Show friends – shows all your friends

Send friend request – Ask others to become your friends

Show friend request – if others have sent you a friend request, you will be able to see who sent you a friend invitation

Search users demo – experimental, will be explained later

Remove friends – Of course you can also remove friends from your friend list

There is also functionality for creating and joining rooms:

Create room
Join (all) rooms

Imagine we were Skype. You have got rooms when you talk with people. For example there is the room between you and your girlfriend, and the room between you and somebody you work with, and another room between you and somebody you don't know yet. All this are by definition rooms.

More room functionality:

Show only joined rooms	User room list
User and friends room list	Invite other users to rooms
Show invitations to room	

Show only joined rooms – does exactly that. Only if you have joined rooms.

User room list - When you have created rooms, this will show all the rooms which you have created

User and friend room list – will get explained later

Invite other users to rooms – You can have even rooms with more than 2 people. You could for example invite another person to the room with you and your girlfriend. That's what this is about.

Show invitations to room – If other people, who possibly are your friends, invite you to their rooms, the rooms will show up here.

Show friends

Note: Before clicking on anything make sure you are working with a new database install without any deleted users. In case you have deleted any users before, you will have to re-install the database so really the same users will show up here.

Let's click on the second user – id 198618, email: 22

This will show up, because he has no friends:

ID	email	Title	first name	last name
----	-------	-------	------------	-----------

Sorry, this user has no friends

[Back](#)

If you click "Show friends" on the first user – user id 198610, email: oliverbach@gmail.com – this will show up:

ID	email	Title	first name	last name
198615	11	22	33	44

[Back](#)

This is because oliverbach@gmail.com and the last user with the email "11" are friends.

To verify this, we can also show on the friends of the last user:

Edit										Show friends
Delete	198615	11		22	33	44	b6d767d2f8ed5d21a44b0e5886680cb9	0		Send friend
										Add friends

And who shows up?

ID	email	Title	first name	last name
198610	oliverbach@gmail.com	Dr.	Oliver	Bachmann

[Back](#)

Now it's clear 11 and oliverbach@gmail.com show up each other as friends.

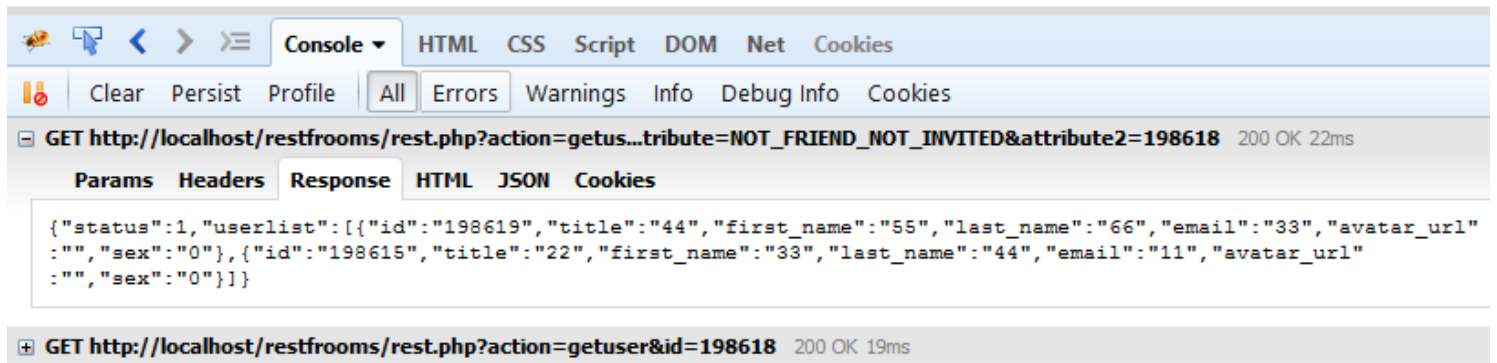
<div> <div>Edit</div> <div>Delete</div> </div>	198618 22	33 44	55	b6d767d2f8ed5d21a44b0e5886680cb9 0	<div> <div>Show friends</div> <div>Show friend requests</div> <div>Send friend request</div> <div>Search users demo</div> <div>Remove friends</div> </div>	Add friend
--	-----------	-------	----	------------------------------------	--	------------

All users that are not friends of 22, and also which are NOT invited already.

Back

Action	id	email	title	first name	last name	avatar url	sex
send friend request	198619	33	44	55	66		0
send friend request	198615	11	22	33	44		0

[Back](#)



When we click now on “send friend request” of both users we first will get a textarea which we can use to give the other persons details about the request:

All users that are not friends of 22, and also which are NOT invited already.

Action	id	email	title	first name	last name	avatar url	sex
send friend request	198619	33	44	55	66		0
send friend request	198615	11	22	33	44		0

Send

Back

When we click on Send the following happens:

```
POST http://localhost/restrooms/rest.php?action=sendf...origin_user_id=198618&destination_user_id=198619 200 OK 20ms
Params Headers Post Response HTML JSON Cookies
{"status":1,"msg":"Created an invitation by the owner with the ID 198618 for 198619"}
```

Again, a rest call, with a success message. We do this for both users!

All users that are not friends of 22, and also which are NOT invited already.

Action	id	email	title	first name	last name	avatar url	sex
	198619	33	44	55	66		0
send friend request	198615	11	22	33	44		0

Back

So-to-say we invite the user with the email “11” too.

```
POST http://localhost/restrooms/rest.php?action=sendf...origin_user_id=198618&destination_user_id=198615 200 OK 34ms
Params Headers Post Response HTML JSON Cookies
{"status":1,"msg":"Created an invitation by the owner with the ID 198618 for 198615"}
```

Success!

Now we can take a look at the received friend requests. 22 has invited the users with the email “33” and “11”. Let’s use the perspective of “33” and “11” now:

Action	id	email	title	first name	last name	md5 password	sex	friend list
Edit Delete	198610	oliverbach@gmail.com	Dr.	Oliver	Bachmann	b6d767d2f8ed5d21a44b0e5886680cb9	1	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends 11
Edit Delete	198618	22	33	44	55	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends
Edit Delete	198619	33	44	55	66	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends
Edit Delete	198615	11	22	33	44	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends oliverbach@gmail.com

The results, for 33:

reason	direction	ID	email	title	first name	last name	Action
ICH BIN NEU	outgoing	19872	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirmed
	incoming	19874	22	33	44	44	confirm

[Back](#)

The result for 11:

reason	direction	ID	email	title	first name	last name	Action
ADADADDA	incoming	19873	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirmed
Hello	incoming	19875	22	33	44	44	confirm

[Back](#)

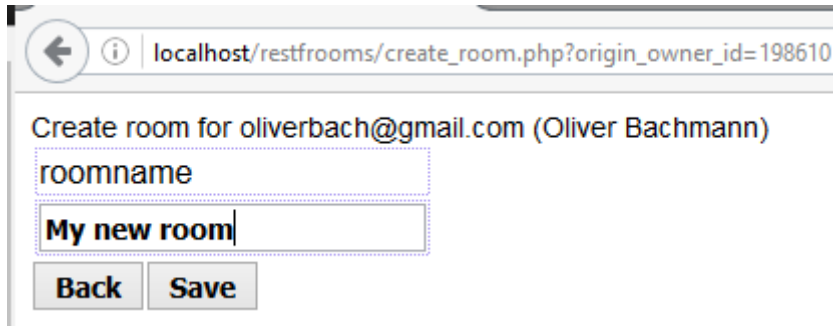
The text on the very left side should be the text you inserted as friend request reason inside the textarea.
 When the users 11 and 33 have confirmed the friend requests of 22, 33 and 11 should show up in the listing on the main page as friends of user 11:

Edit Delete	198618	22	33	44	55	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends 11, 33
Edit Delete	198619	33	44	55	66	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends 22
Edit Delete	198615	11	22	33	44	b6d767d2f8ed5d21a44b0e5886680cb9	0	Show friends Show friend requests Send friend request Search users demo Add friends Remove friends oliverbach@gmail.com, 22

Also, 33 and 11 should have 22 as friend.

Creating a room

Let's create a room for oliverbach@gmail.com



localhost/restrooms/create_room.php?origin_owner_id=198610

Create room for oliverbach@gmail.com (Oliver Bachmann)

roomname

My new room

Back Save

After creating a room the view changes to the view which appears when pressing "Show rooms" on a user on the main page.

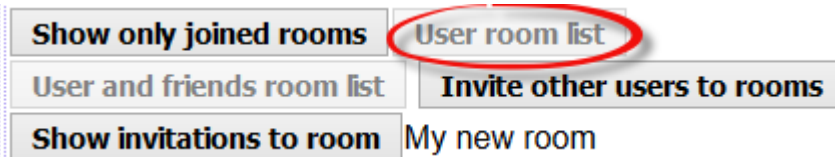
The new room will show up:

ID	Name	Members	Owner		
198621	test - 1		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198622	test - 2	22	oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198624	test - 3		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198627	test - 4		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198628	test - 5		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198629	test - 6		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198630	test - 7	33,11	oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198631	test - 8		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198632	test - 9		oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198633	My new room	oliverbach@gmail.com	oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room

Back

Editing and deleting the room is possible. All happens by making ajax calls to the rest.php file.

You can reach this page by clicking here on the main page:

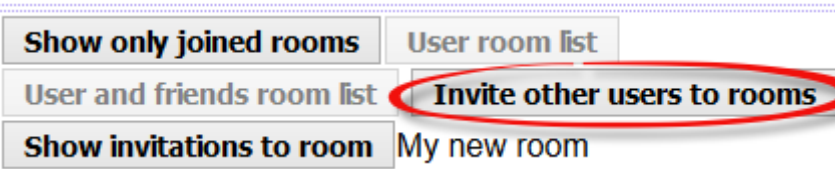


Show only joined rooms User room list

User and friends room list Invite other users to rooms

Show invitations to room My new room

The only member inside the newly created room is oliverbach@gmail.com. If we want other people to join in, we have to click here:



Show only joined rooms User room list

User and friends room list Invite other users to rooms

Show invitations to room My new room


Invite other users to room

First you have to choose to which room you would like to invite somebody

Invitation by: oliverbach@gmail.com (Oliver Bachmann)

ID	Name	Members	Owner	
198621	test - 1		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 1
198622	test - 2	22	oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 2
198624	test - 3		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 3
198627	test - 4		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 4
198628	test - 5		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 5
198629	test - 6		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 6
198630	test - 7	33,11	oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 7
198631	test - 8		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 8
198632	test - 9		oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room test - 9
198633	My new room	oliverbach@gmail.com	oliverbach@gmail.com (Oliver Bachmann)	Invite users on the following page to room My new room

Back

 [localhost/restfrooms/invite_to_room2.php?byuserid=198610&toroomid=198621](#)

Invitation by: oliverbach@gmail.com (Oliver Bachmann)
Invitation to this room: test - 1

ID	email	Title	first name	last name	
198615	11	22	33	44	invite

Back

Only the friends of [oliverbach@gmail.com](#) should show up here.

How does this list get created? In effect, it are three ajax calls:

```
window.onload = function() {  
    getUserInfo(198610, function(responseText) {  
        $('#email').html(responseText.email);  
        $('#first_name').html(responseText.first_name);  
        $('#last_name').html(responseText.last_name);  
    });  
    getRoomInfo(198621, function(responseText) {  
        $('#roomname').html(responseText.name);  
    });  
    getFriendList(198610, "INVITE", 198610, 198621);  
}
```

The first function `getUserInfo` in `users.js`:

```
function getUserInfo(id, callback) {

    $.ajax({
        url: "rest.php?action=getuser&id="+id,
        type: "GET",
        async : false,
        dataType: "json",
        processData: false
    }).done(function(responseText) {

        callback(responseText);

    });

}
```

The function callback is also specified here:

```
<script>
    window.onload = function() {
        getUserInfo(198610, function(responseText) {
            $('#email').html(responseText.email);
            $('#first_name').html(responseText.first_name);
            $('#last_name').html(responseText.last_name);
        });
        getRoomInfo(198621, function(responseText) {
            $('#roomname').html(responseText.name);
        });
        getFriendList(198610, "INVITE", 198610, 198621);
    }
</script>
```

What it does here, that is populating the divs with content from `responseText`.

We make 3 asynchronous ajax calls. We also want to show the room name, so we call the function `getRoomInfo` from `rooms.js`:

```
function getRoomInfo(id, callback) {

    $.ajax({
        url: "rest.php?action=get_room&roomid="+id,
        type: "GET",
        async : false,
        dataType: "json",
        processData: false
    }).done(function(responseText) {

        callback(responseText);

    });

}
```

Again, the information from responseText is used to populate the fields in the table. Mainly showing the room name:

Invitation by: oliverbach@gmail.com (Oliver Bachmann)

Invitation to this room: test - 1

ID	email	Title	first name	last name	
198615	11	22	33	44	invite

Back

When we click on “invite” the following ajax call gets made:

Clear Persist Profile All Errors Warnings Info Debug Info Cookies

GET http://localhost/restrooms/rest.php?action=getuser&id=198610 200 OK 16ms

Params Headers Response HTML JSON Cookies

{ "id": "198610", "status": 1, "title": "Dr.", "first_name": "Oliver", "last_name": "Bachmann", "email": "oliverbach@gmail.com", "md5pwd": "b6d767d2f8ed5d21a44b0e5886680cb9", "avatar_url": "avatar", "sex": "1" }

GET http://localhost/restrooms/rest.php?action=get_room&roomid=198621 200 OK 13ms

GET http://localhost/restrooms/rest.php?action=getfr...rray&origin_user_id=198610&invite_to_room=198621 200 OK 28ms

GET http://localhost/restrooms/rest.php?action=inviteuser&invitation_by=198610&invitation_to=198615&roomid=198621 200 OK 13ms

Params Headers Response HTML JSON Cookies

{ "status": 1, "msg": "OK" }

rest.php?action=inviteuser&invitation_by=198610&invitation_to=198615&roomid=198621

The screen will change to the following:

Invitation by: oliverbach@gmail.com (Oliver Bachmann)

Invitation to this room: test - 1

ID	email	Title	first name	last name	
198615	11	22	33	44	INVITED

Back

Now we have invited the user with the email “11” to the room test – 1 – and the initiator is [oliverbach@gmail.com](#) – now let’s have a look if we can see the invitation somewhere.

Edit Delete

198615	11	22	33	44	b6d767d2f8ed5d21a44b0e5886680cb9
--------	----	----	----	----	----------------------------------

Show friends Show friend requests

Send friend request Search users demo

Add friends Remove friends oliverbach@gmail.com, 22

Create room

Join (all) rooms

Show only joined rooms User and friends room list Invite other users to rooms

Show invitations to room

Room not found (198653)

When we click on “Show invitations to rooms” we indeed get shown the invitation we just initiated:

Room name	direction	ID	going to/coming from email	going to/coming from Title	going to/coming from First name	going to/coming from Last name	Action
test - 2	incoming	198653	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirm Delete invitation
test - 1	incoming	198654	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirm Delete invitation

Back

Don't ask me why there is also an invitation to room test – 2 ... important here is that the invitation to test – 1 exists. We can now confirm the invitation to the room test – 1:

Room name	direction	ID	going to/coming from email	going to/coming from Title	going to/coming from First name	going to/coming from Last name	Action	
test - 2	incoming	198653	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirm	Delete invitation
test - 1	incoming	198654	oliverbach@gmail.com	Dr.	Oliver	Oliver	Confirm	Delete invitation

Back

Now the following ajax call will get made:

GET http://localhost/restrooms/rest.php?action=joinroom&origin_user_id=198615&roomid=198621 200 OK 31ms

ParamsHeadersResponseHTMLJSONCookies

```
{ "status": 1, "msg": "OK" }
```

We should now take a look at the room that we just invited to.

EditDelete

198615 11

22 33 44

b6d767d2f8ed5d21a44b0e5886680cb9 0

Show friends

Show friend requests

Send friend request

Search users demo

Add friends

Remove friends

oliverbach@gmail.com, 22

Create room

Join (all) rooms

Show only joined rooms

User room list

User and friends room list

Invite other users to rooms

Show invitations to room

test - 7, room not found (198653), test - 1

Remember: This is user 11. We are now part of test -1. Let's click on this, and you will get to see the joined rooms for user 11:

ID	Name	Members	Owner		
198621	test - 1	11	oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room
198630	test - 7	33,11	oliverbach@gmail.com (Oliver Bachmann)	Delete room	Edit room

Back

“Show only joined rooms” is the killer feature, which means that with it, you can check if all invitations were properly made.

Search users demo

You will get to see this screen:

Items per page: 8
Current page:1

PAGE UP

PAGE DOWN

email	title	first_name	last_name	avatar_url	Action
-------	-------	------------	-----------	------------	--------

Whatever you type in will get searched. The character “e” will be the most successful string to search for. When we type that we get to see these users:

Items per page: 8
Current page:1

PAGE UP

e

PAGE DOWN

email	title	first_name	last_name	avatar_url	Action
oliverbach@gmail.com	Dr.	Oliver	Bachmann	avatar	<code>inviteUser(198610);</code>

Since there is by default only one user with the character “e” in it it would be a good time to test another button on the main page, which adds more dummy users to the database.

Click on this button on the main page:

Add a new user

Login

Create 10 dummy users

Logout

Remove nonexistent room owners and friends

And all of a sudden we have more users, very dummy ones indeed:

<div>EditDelete</div>	198621	USER13	Herr	first_nameecAfEabDeAaafCB	last_namebabfdAFDAaCeaEB	c51ce410c124a10e0db5e4b97fc2af39	0	<div>Show friendsShow friend requestsSend friend requestSearch users demoAdd friendsRemove friends</div>	<div>Create roomJoin (all) rooms</div>	<div>Show only joined roomsUser room listUser and friends room listInvite other users to roomsShow invitations to room</div>
<div>EditDelete</div>	198622	USER14	Herr	first_nameeDbbAFcebBaaEEc	last_namebEDCcEADfcaBeAB	aab3238922bcc25a6f606eb525fdcf56	0	<div>Show friendsShow friend requestsSend friend requestSearch users demoAdd friendsRemove friends</div>	<div>Create roomJoin (all) rooms</div>	<div>Show only joined roomsUser room listUser and friends room listInvite other users to roomsShow invitations to room</div>
<div>EditDelete</div>	198623	USER15	Herr	first_nameBcabCefEeaaDBEf	last_namebEEFDcBdaFdBEbf	9bf31c7f062936a96d3c8bd1f82ff3	0	<div>Show friendsShow friend requestsSend friend requestSearch users demoAdd friendsRemove friends</div>	<div>Create roomJoin (all) rooms</div>	<div>Show only joined roomsUser room listUser and friends room listInvite other users to roomsShow invitations to room</div>
<div>EditDelete</div>	198624	USER16	Herr	first_nameBfcbBDbadbfbEda	last_namebFCfAfCbFAaFADb	c74d97b01eae257e44aa9d5bade97baf	0	<div>Show friendsShow friend requestsSend friend requestSearch users demoAdd friendsRemove friends</div>	<div>Create roomJoin (all) rooms</div>	<div>Show only joined roomsUser room listUser and friends room listInvite other users to roomsShow invitations to room</div>
<div>EditDelete</div>	198625	USER17	Herr	first_namefEADCfaEfaFDeAf	last_nameceCeCEAEFEBBDb	70efd2ec9b086079795c442636b55fb	0	<div>Show friendsShow friend requestsSend friend requestSearch users demoAdd friendsRemove friends</div>	<div>Create roomJoin (all) rooms</div>	<div>Show only joined roomsUser room listUser and friends room listInvite other users to roomsShow invitations to room</div>
<div>Edit</div>								<div>Show friendsShow friend requests</div>	<div>Create room</div>	<div>Show only joined roomsUser room listUser and friends room list</div>

When you look for “e” now you will get a bunch of matching pairs:

Items per page: 8
Current page: 1

<div>PAGE UP</div>	<div>e</div>	<div>PAGE DOWN</div>					
email	title	first_name	last_name	avatar_url	Action		
oliverbach@gmail.com	Dr.	Oliver	Bachmann	avatar	<div>inviteUser(198610);</div>		
USER12	Herr	first_nameeeefbFeBfcAABcEc	last_namefDAbDcedfaEaeAD	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198620);</div>		
USER13	Herr	first_nameecAfEabDeAaafCB	last_namebabfdAFDAaCeaEB	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198621);</div>		
USER14	Herr	first_nameeDbbAFcebBaaEEc	last_namebEDCcEADfcaBeAB	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198622);</div>		
USER15	Herr	first_nameBcabCefEeaaDBEf	last_namebEEFDcBdaFdBEbf	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198623);</div>		
USER16	Herr	first_nameBfcbBDbadbfbEda	last_namebFCfAfCbFAaFADb	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198624);</div>		
USER17	Herr	first_namefEADCfaEfaFDeAf	last_nameceCeCEAEFEBBDb	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198625);</div>		
USER18	Herr	first_nameECCBefaceeCeffDc	last_nameeaAACBabfBaacBa	https://www.google.de/logos/doodles/2015/evidence-of-water-found-on-mars-5652760466817024.2-hp.gif	<div>inviteUser(198626);</div>		

Now you should know

- 1. How to create new rooms
- 2. How to invite others to new rooms
- 3. How to confirm invitations

The same is true for friend invitations / requests.

Outstanding are the features “show joined rooms” and “show friends”, with them you can always check if the other buttons were successful.

Also, greyed out buttons do not mean that they do not work, they only mean that they are not really important. You can get to understand most of the features by clicking on the black buttons.