

Rocket-Chat-Sooty Project Documentation

Task #1: Create System Roles

default roles for system

Admin

EjabAdmin

KBAdmin

HRAdmin

NeedAdmin

TrainingAdmin

|| These permissions needs to be migrated as the server runs.

|| These roles can't be archived like other roles, because the system need them to operate.

Status: (done)

I have created a custom function in Rocket-Chat authorization module. It creates system roles when called from front-end. It adds system defined default role which cannot be deleted. This functions also serves for the purpose editing / modifying a role. It can be called from front-end console as following:

Syntax:

```
Meteor.call("addSystemRole", "nameOfRole", "description", "scope", function(error, result){
  console.log(result);
});
```

Example -

```
Meteor.call("addSystemRole", "NewTestAdmin", "new-test", "global",
function(error, result){
  console.log(result);
});
```

Note: The parameters in the above function are mandatory and must to be passed by keeping in a programmer mindset convention. I.e, please follow the standard naming convention. Avoid use of special characters.

Main files handling the request:

- addPermissionToRole.js
(/Rocket.Chat/packages/rocketchat-authorization/server/methods/addPermissionToRole.js)
- Role.js
(/Rocket.Chat/packages/rocketchat-authorization/server/methods/Role.js)

Code snippet:

addPermissionToRole.js

```
'addSystemRole'(role, desc, scope) {  
    const someValue = RocketChat.models.Roles.addSystemRole(role,  
desc, scope);  
    return 'result = ' + someValue;  
},
```

Role.js

```
// Add Role  
  
addSystemRole(newRole, description, scope) {  
    if ((newRole || description || scope) !== null) {  
        newRole = newRole.trim();  
        description = description.trim();  
        scope = scope.trim();  
        const roleData = {  
            description: description,  
            scope: scope,  
            protected: true,  
            mandatory2fa: true,  
        };  
        this.upsert({ _id: newRole }, { $set: roleData });  
        return 'success: Added Role ' + newRole;  
    } else {  
        return "Error: can't insert null values.";  
    }  
}
```

```
}

```

OUTPUT Screenshots:

Permissions									
	admin	user	bot	EjabAdmin	KBAAdmin	HRAdmin	NeedAdmin	TrainingAdmin	
Access Mailer Screen [access-mailer]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Access Permissions Screen [access-permissions]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Add Oauth Service [add-oauth-service]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Add User to Any Public Channel [add-user-to-any-c-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Add User to Any Private Channel [add-user-to-any-p-room]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Add User to Any Joined Channel [add-user-to-joined-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Archive Room [archive-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Assign Admin Role [assign-admin-role]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Auto Translate [auto-translate]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
badges [badges]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Database entry-result:

The screenshot displays the NoSQLBooster application interface. On the left is a sidebar with various application icons. The main window is titled "NoSQLBooster for MongoDB DB" and contains a menu bar (File, Edit, Options, View, Window, Help) and a toolbar with icons for Connect, Run, Stop, Import, Export, Monitoring, Test Data, and Schema. Below the toolbar is a "Connection Tree" showing a list of databases and collections. The "meteor:rocketchat_permissions" collection is selected. The main panel shows a MongoDB query: `db.rocketchat_roles.find({}).projection({}).sort({_id:-1}).limit(100)`. Below the query, a table displays the results of the query, showing 11 documents from the "rocketchat_roles" collection. The table has columns: _id, _updatedAt, description, mandatory2fa, protected, and scope. The data includes roles like user, owner, guest, bot, admin, TrainingAdmin, NewTestAdmin, NeedAdmin, KBAdmin, HRAdmin, and EjabiAdmin.

_id	_updatedAt	description	mandatory2fa	protected	scope
1	25/01/2019, 12:10:51		false	true	Users
2	25/01/2019, 12:10:51	Owner	false	true	Subscriptions
3	25/01/2019, 12:10:51		false	true	Users
4	25/01/2019, 12:10:51		false	true	Users
5	25/01/2019, 12:10:51	Admin	false	true	Users
6	23/01/2019, 16:46:35	TrainingAdmin	false	true	Users
7	24/01/2019, 20:53:04	new-test	true	true	global
8	23/01/2019, 16:46:14	NeedAdmin	false	true	Users
9	23/01/2019, 16:40:21	KBAdmin	false	true	Users
10	23/01/2019, 16:41:29	HRAdmin	false	true	Users
11	23/01/2019, 16:16:04	EjabiAdmin	false	true	Users

Copyright © nosqlbooster.com Version 5.0.3 Free Edition

Task #2: Add permissions to our rocket.chat app

You need to add our default permissions to sooty rocket.chat.

eg.

- Employees leaderboard
- Lite Dashboard
- Ejabi Questions
- Ejabi Network Graph
- Ejabi ML
- Badges

And as a developer, I can add new permissions due to business needs.

|| Add our permissions to our system.

Status: (done)

I have created a custom function in Rocket-Chat authorization module. It creates permissions to the rocket-chat when called from front-end. It adds user- defined permissions to the rocket-chat system. This functions also serves for the purpose editing / modifying a permission. It's a generic function which can be created in two ways to serve two different purposes:

1. **When called with a single parameter:**
it creates a permission in rocket-chat system.
2. **When called with a two parameters:**
it creates a permission and assigns the roles to it. First parameter is the name of permission and second parameter is an array which may contain a single or multiple roles.

It can be called from front-end console as following:

Syntax:

```
Meteor.call("addPermission", "name-of-permission", ['role1', 'role2', 'role3'],  
function(error, result){  
  
  console.log(result);  
  
});
```

Example -

```
Meteor.call("addPermission", "new-test-permission1",
["admin", "owner", "KBAAdmin", "EjabAdmin"], function(error,
result) {

console.log(result);

});
```

Note: The parameters in the above function are mandatory and must to be passed by keeping in a programmer mindset convention. I.e, please follow the standard naming convention. Avoid use of special characters.

Main files handling the request:

- addPermissionToRole.js
(/Rocket.Chat/packages/rocketchat-authorization/server/methods/addPermissionToRole.js)
- Permission.js
(/Rocket.Chat/packages/rocketchat-authorization/server/methods/Permission.js)

Code snippet:

addPermissionToRole.js

```
'addPermission'(permission, roles) {
    const someValue =
RocketChat.models.Permissions.addPermission(permission, roles);
    return 'result = ' + someValue;
},
```

Permission.js

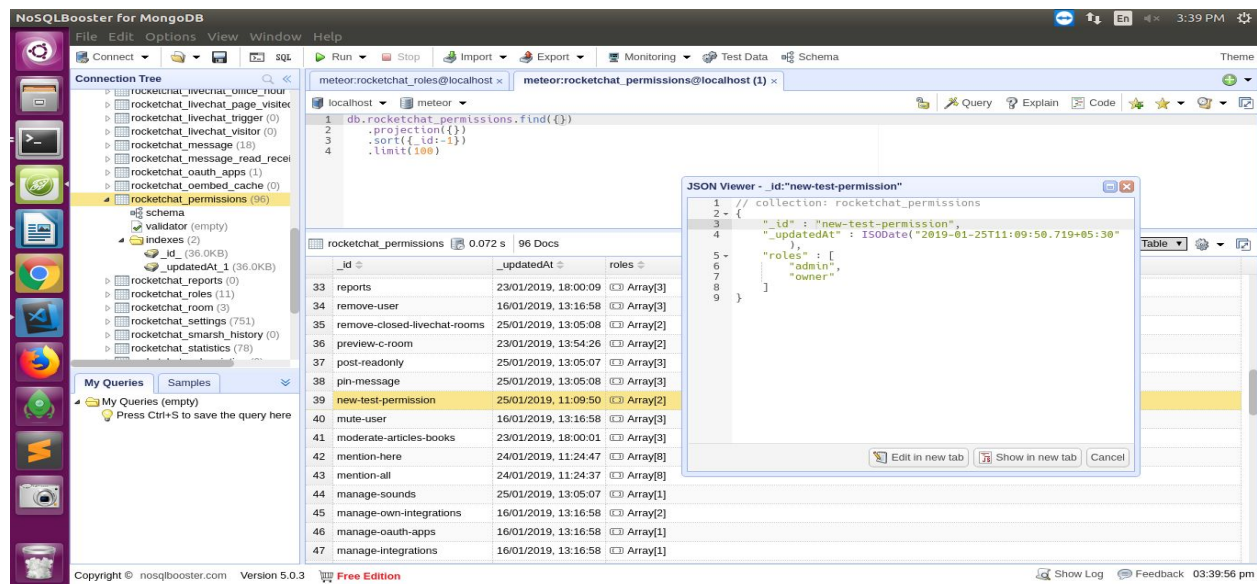
```
// Add Permissions
addPermission(permission, role = '') {
    if (permission != null) {
        this.upsert({ _id: permission }, { $set: { roles: role }
});
        return 'success: Added permission: ' + permission;
    } else {
```

```
        return "Error: can't insert null values.";
    }
}
```

OUTPUT Screenshots:

[illegible]

Database Entry -result:



Task #3: Create Permissions for KBAdmin

Will has these permissions

- Employees leaderboard
- Moderate Articles/Books
- Reports
- Knowledge Graph
- KB ML
- Badges

Status: (done)

I have called our custom add permission API method from front-end console to enter these permissions after creating KBAdmin Role. First I called my another custom method with appropriate parameters to create a system Role as follows:

Calling Custom method-

```
Meteor.call("addSystemRole", "KBAdmin", "kb-Admin", "global",
function(error, result){

console.log(result);

});
```

OUTPUT Screenshots:

The top screenshot shows the RocketChat Admin Panel at `localhost:3000/admin/permissions`. The 'Permissions' tab is active, displaying a table of permissions and their assigned roles.

	admin	user	bot	EjabAdmin	KBAAdmin	HRAAdmin	NeedAdmin	TrainingAdmin
Access Mailer Screen [access-mailer]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Access Permissions Screen [access-permissions]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add OAuth Service [add-oauth-service]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add User to Any Public Channel [add-user-to-any-c-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add User to Any Private Channel [add-user-to-any-p-room]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add User to Any Joined Channel [add-user-to-joined-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Archive Room [archive-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign Admin Role [assign-admin-role]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto Translate [auto-translate]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
badges [badges]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ban User [ban-user]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bulk Create Channels	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The bottom screenshot shows the MongoDB interface. A query is run on the `rocketchat_roles` collection: `db.rocketchat_roles.find({}).projection({}).sort({_id:-1}).limit(100)`. The results show the role 'EjabAdmin' with the following details:

```
{
  "_id": "EjabAdmin",
  "_updatedAt": ISODate("2019-01-23T16:16:04.924+05:30"),
  "description": "EjabAdmin",
  "mandatory2fa": false,
  "protected": true,
  "scope": "Users"
}
```

Task #4: Create Permissions for EjabAdmin

Will has these permissions

- Employees leaderboard

- Lite Dashboard
- Ejabi Questions
- Ejabi Network Graph
- Ejabi ML
- Badges






Status: (done)

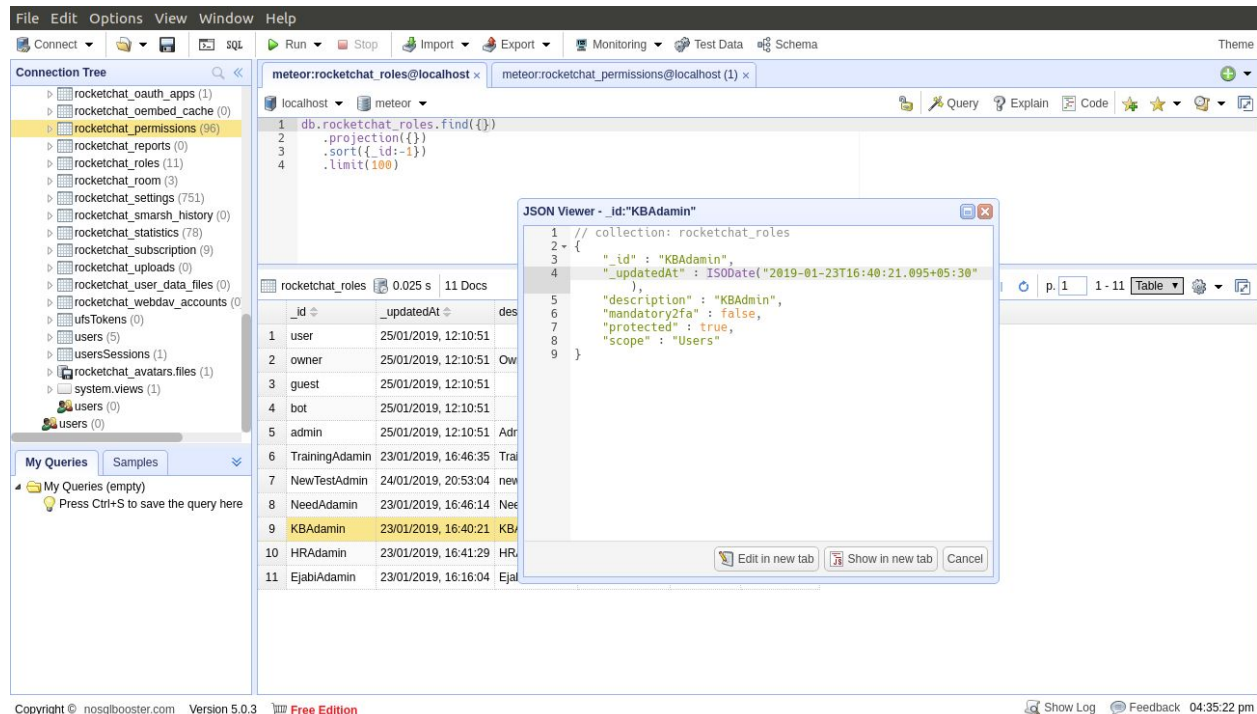
I have called our custom add permission API method from front-end console to enter these permissions after creating EjabiAdmin Role. First I called my another custom method with appropriate parameters to create a system Role as follows:

Calling Custom method-

```
Meteor.call("addSystemRole", "EjabiAdmin", "Ejabi-Admin", "global",
function(error, result){
console.log(result);
});
```

OUTPUT Screenshots:

	 admin	 user	 bot	 EjabiAdmin	 KBAdmin
Access Mailer Screen [access-mailer]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Access Permissions Screen [access-permissions]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add Oauth Service [add-oauth-service]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Add User to Any Public Channel [add-user-to-any-c-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Add User to Any Private Channel [add-user-to-any-p-room]	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add User to Any Joined Channel [add-user-to-joined-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Archive Room [archive-room]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign Admin Role [assign-admin-role]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto Translate [auto-translate]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
badges [badges]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



Task #5: Send message to others

As an employee/manager, I can send a message to other employees in the organization, So that I can communicate with others.

Status: (done)

I have provided all the admin such as - EjabiAdmin, KbAdmin, Admin, HRAdmin, NeedAdmin, TrainingAdmin etc. to the global scope and other users to “user” scope. We are providing anybody with a user scope has permission to send messages to each other. While Admins have permission to access the user profiles and send direct messages to them.

Although, A user can send message to others, an Admin has rights to remove this permission. He can grant or take away the permissions from users.

Task #6: Create Channel

As an employee/manager, I can create a channel in sootyai, So that I can communicate with other employees in the organization..

Status: (done)

I have provided all the admin such as - EjabiAdmin, KbAdmin, Admin, HRAdmin, NeedAdmin, TrainingAdmin etc. to the global scope and other users to “user” scope. We are providing anybody with a user scope has permission to create a new channel. While Admins have permission to create a private channel and invite users to it. Users can join a channel through invite codes only if it is set to private. Other channels like public channel, can be created by any

user and can be added users to it. However the owner of a group / channel to grant rights to other user. In general case, an admin has supreme access. A public channel can be deleted by the owner only. Anybody in a public channel can text to each other. Delete and edit his messages but can't delete others messages.

Thus, A user can create his own channels and set its attributes, add users, remove users.