\* @param array $params

\*

\* @return bool \*/

public function execute($user, $item, $params)

{

return isset($params['post']) ? $params['post']->createdBy == $user : false;

}

}

2. Create a console command, command/RbacController. php, to init the RBAC rules command:

<?php

namespace app\commands;

use app\models\User; use Yii;

use yii\console\Controller;

/\*\*

\* Class RbacController.

\* @package app\commands \*/

class RbacController extends Controller {

public function actionInit()

{

$auth = Yii::$app->authManager;

$createPost = $auth->createPermission('createPost');

$createPost->description = 'Create a post';

$updatePost = $auth->createPermission('updatePost');

$updatePost->description = 'Update a post';

$updatePost = $auth->createPermission('updatePost');

$updatePost->description = 'Update a post';

$deletePost = $auth->createPermission('deletePost');

$deletePost->description = 'Delete a post';

$readPost = $auth->createPermission('readPost');

$readPost->description = 'Read a post';

$authorRule = new \app\rbac\AuthorRule();

// add permissions $auth->add($createPost);

$auth->add($updatePost);

$auth->add($deletePost);

$auth->add($readPost);

$auth->add($authorRule);

// add the "updateOwnPost" permission and associate the rule with it. $updateOwnPost = $auth->createPermission('updateOwnPost'); $updateOwnPost->description = 'Update own post';

$updateOwnPost->ruleName = $authorRule->name;

$auth->add($updateOwnPost);

$auth->addChild($updateOwnPost, $updatePost);

// create Author role

$author = $auth->createRole('author');

$auth->add($author);