chamilo / **chamilo-lms** Public

Code | Issues 399 | Pull requests 13 | Discussions | Actions | Projects

# Commit c9c30cd

**AngelFQC** committed 3 weeks ago · ✕ 2 / 6 · Verified

```
Feature: Add custom state provider for CourseRelUser collection to handle role-based access and filtering

See advisory GHSA-x373-8j9j-g5pj
```

master · v2.0.0 v2.0.0-RC.3

1 parent e167bdc commit c9c30cd

## 2 files changed +112 -2 lines changed

Top

Filter files…

- src/CoreBundle
  - Entity
    - CourseRelUser.php
  - State
    - CourseRelUserCollectionStateProvider.php

## 2 files changed +112 -2 lines changed

Search within code

src/CoreBundle/Entity/CourseRelUser.php

```
@@ -16,6 +16,7 @@
16 16  use ApiPlatform\Metadata\Post;
17 17  use Chamilo\CoreBundle\Filter\PartialSearchOrFilter;
18 18  use Chamilo\CoreBundle\Repository\CourseRelUserRepository;
   19 + use Chamilo\CoreBundle\State\CourseRelUserCollectionStateProvider;
19 20  use Chamilo\CoreBundle\State\CourseRelUserStateProcessor;
20 21  use Chamilo\CoreBundle\State\UserCourseSubscriptionsStateProvider;
21 22  use Chamilo\CoreBundle\Traits\UserTrait;
```

```
@@ -30,8 +31,12 @@
 */
#[ApiResource(
    operations: [
-        new Get(security: "is_granted('ROLE_ADMIN') or is_granted('ROLE_TEACHER') or is_granted('ROLE_SESSION_MANAGER') or object.user == user"),
-        new GetCollection(security: "is_granted('ROLE_ADMIN') or is_granted('VIEW', object.course)"),
+        new Get(
+            security: "is_granted('ROLE_ADMIN') or is_granted('ROLE_TEACHER') or is_granted('ROLE_SESSION_MANAGER') or object.getUser() == user"
+        ),
+        new GetCollection(
+            provider: CourseRelUserCollectionStateProvider::class
+        ),
         new Post(
             security: "is_granted('ROLE_USER')",
             securityPostDenormalize: 'object.getUser() == user',
```

...te/CourseRelUserCollectionStateProvider.php

```
@@ -0,0 +1,105 @@
+ <?php
+
+ /* For licensing terms, see /license.txt */
+
+ declare(strict_types=1);
+
+ namespace Chamilo\CoreBundle\State;
+
+ use ApiPlatform\Doctrine\Orm\Extension\FilterExtension;
+ use ApiPlatform\Doctrine\Orm\Extension\OrderExtension;
+ use ApiPlatform\Doctrine\Orm\Extension\PaginationExtension;
+ use ApiPlatform\Doctrine\Orm\Extension\QueryResultCollectionExtensionInterface;
+ use ApiPlatform\Doctrine\Orm\State\CollectionProvider;
+ use ApiPlatform\Doctrine\Orm\Util\QueryNameGenerator;
+ use ApiPlatform\Metadata\GetCollection;
+ use ApiPlatform\Metadata\Operation;
+ use ApiPlatform\State\ProviderInterface;
```

```
18 + use Chamilo\CoreBundle\Entity\CourseRelUser;
19 + use Chamilo\CoreBundle\Entity\User;
20 + use Chamilo\CoreBundle\Helpers\UserHelper;
21 + use Chamilo\CoreBundle\Repository\CourseRelUserRepository;
22 + use Chamilo\CoreBundle\Repository\Node\CourseRepository;
23 + use Chamilo\CoreBundle\Security\Authorization\Voter\CourseVoter;
24 + use Symfony\Bundle\SecurityBundle\Security;
25 + use Symfony\Component\Security\Core\Exception\AccessDeniedException;
26 +
27 + /**
28 +  * @template-implements ProviderInterface<CourseRelUser>
29 +  */
30 + final class CourseRelUserCollectionStateProvider implements ProviderInterface
31 + {
32 +     private array $extensions;
33 +
34 +     public function __construct(
35 +         private readonly CollectionProvider $collectionProvider,
36 +         private readonly CourseRelUserRepository $courseRelUserRepository,
37 +         private readonly UserHelper $userHelper,
38 +         private readonly Security $security,
39 +         private readonly CourseRepository $courseRepo,
40 +         FilterExtension $filterExtension,
41 +         OrderExtension $orderExtension,
42 +         PaginationExtension $paginationExtension,
43 +     ) {
44 +         $this->extensions = [$filterExtension, $orderExtension,
    $paginationExtension];
45 +     }
46 +
47 +     public function provide(Operation $operation, array $uriVariables = [],
    array $context = []): array|object|null
48 +     {
49 +         if (!$operation instanceof GetCollection) {
50 +             return $this->collectionProvider->provide($operation,
    $uriVariables, $context);
51 +         }
52 +
53 +         // Privileged roles: return the full collection with all API Platform
    filters applied.
```

```
54  +         if (
55  +             $this->security->isGranted('ROLE_ADMIN')
56  +             || $this->security->isGranted('ROLE_SUPER_ADMIN')
57  +             || $this->security->isGranted('ROLE_GLOBAL_ADMIN')
58  +         ) {
59  +             return $this->collectionProvider->provide($operation,
      $uriVariables, $context);
60  +         }
61  +
62  +         // Students and other authenticated users: restrict to their own
      subscriptions.
63  +         $currentUser = $this->userHelper->getCurrent();
64  +
65  +         if (!$currentUser instanceof User) {
66  +             throw new AccessDeniedException('User not authenticated.');
67  +         }
68  +
69  +         if ($context['filters']['course'] ?? null) {
70  +             $course = $this->courseRepo->find($context['filters']['course']);
71  +
72  +             if (!$this->security->isGranted(CourseVoter::VIEW, $course)) {
73  +                 throw new AccessDeniedException();
74  +             }
75  +         }
76  +
77  +         $qb = $this->courseRelUserRepository->createQueryBuilder('cru');
78  +         $qb
79  +             ->andWhere(
80  +                 $qb->expr()->exists(
81  +                     'SELECT 1 FROM '.CourseRelUser::class.' my_cru
82  +                      WHERE my_cru.course = cru.course
83  +                        AND my_cru.user = :currentUser'
84  +                 )
85  +             )
86  +             ->setParameter('currentUser', $currentUser)
87  +         ;
88  +
89  +         $queryNameGenerator = new QueryNameGenerator();
90  +         $items = [];
91  +
```

```
92  +         foreach ($this->extensions as $extension) {
93  +             $extension->applyToCollection($qb, $queryNameGenerator,
    CourseRelUser::class, $operation, $context);
94  +
95  +             if (
96  +                 $extension instanceof QueryResultCollectionExtensionInterface
97  +                 && $extension->supportsResult(CourseRelUser::class, $operation,
    $context)
98  +             ) {
99  +                 $items = $extension->getResult($qb, CourseRelUser::class,
    $operation, $context);
100 +             }
101 +         }
102 +
103 +         return [] !== $items ? $items : $qb->getQuery()->getResult();
104 +     }
105 + }
```

## Comments 0

Please sign in to comment.