

```
1
2 package acme.features.manager.assignment;
3
4 import java.util.Collection;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8
9 import acme.client.data.models.Dataset;
10 import acme.client.services.AbstractService;
11 import acme.client.views.SelectChoices;
12 import acme.entities.project.Assignment;
13 import acme.entities.project.UserStory;
14 import acme.features.manager.project.ManagerProjectRepository;
15 import acme.roles.Manager;
16
17 @Service
18 public class ManagerAssignmentDeleteService extends AbstractService<Manager,
19 Assignment> {
20     // Internal state
21     -----
22     @Autowired
23     private ManagerProjectRepository repository;
24
25     // AbstractService interface
26     -----
27
28     @Override
29     public void authorise() {
30         boolean status;
31         int id = super.getRequest().getData("id", int.class);
32         Manager manager = this.repository.findManagerProjectByAssignmentId
33 (id);
34         status = super.getRequest().getPrincipal().getActiveRoleId() ==
35 manager.getId();
36         super.getResponse().setAuthorised(status);
37     }
38
39     @Override
40     public void load() {
41         Assignment assignment;
42         int id;
43         id = super.getRequest().getData("id", int.class);
44         assignment = this.repository.findAssignmentById(id);
45 }
```

```
46     super.getBuffer().addData(assignment);
47 }
48
49 @Override
50 public void bind(final Assignment assignment) {
51     assert assignment != null;
52
53     super.bind(assignment, "project.title", "userStory");
54 }
55
56 @Override
57 public void validate(final Assignment assignment) {
58     boolean updateable =
59         this.repository.findProjectOfAnAssignmentByAssignmentId(assignment.getId())
60             .isDraft();
61     if (!super.getBuffer().getErrors().hasErrors("project"))
62         super.state(updateable, "*", "manager.project.form.updateable");
63 }
64
65 @Override
66 public void perform(final Assignment assignment) {
67     assert assignment != null;
68
69     this.repository.delete(assignment);
70 }
71
72 @Override
73 public void unbind(final Assignment assignment) {
74     assert assignment != null;
75
76     Dataset dataset;
77
78     int id = super.getRequest().getPrincipal().getActiveRoleId();
79     SelectChoices userStoriesChoices;
80     Collection<UserStory> userStories =
81         this.repository.findAllUserStoriesOfAManagerById(id);
82     userStoriesChoices = SelectChoices.from(userStories, "title",
83         assignment.getUserStory());
84
85     dataset = super.unbind(assignment, "project.title", "userStory");
86     dataset.put("masterId", assignment.getProject().getId());
87     dataset.put("userStories", userStoriesChoices);
88
89     super.getResponse().addData(dataset);
90 }
91
```

ManagerAssignmentDeleteService.java

domingo, 7 de julio de 2024 9:53

```
92 }  
93
```