

```
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 ManagerAssignmentUpdateService extends AbstractService<Manager, Assignment> {
19
20     @Autowired
21     private ManagerProjectRepository repository;
22
23
24     @Override
25     public void authorise() {
26         boolean status;
27         int assignmentId = super.getRequest().getData("id", int.class);
28         Manager manager = this.repository.findManagerProjectByAssignmentId(assignmentId);
29         status = manager != null && super.getRequest().getPrincipal().hasRole
30             (Manager.class) && manager.getId() == super.getRequest().getPrincipal().getActiveRoleId();
31         super.getResponse().setAuthorised(status);
32     }
33
34     @Override
35     public void load() {
36         int id = super.getRequest().getData("id", int.class);
37         Assignment assignment = this.repository.findAssignmentById(id);
38         super.getBuffer().addData(assignment);
39     }
40
41
42
43     @Override
44     public void bind(final Assignment assignment) {
45         assert assignment != null;
46
47         super.bind(assignment, "project.title", "userStory");
48     }
49
50
51     @Override
52     public void validate(final Assignment assignment) {
53         assert assignment != null;
54         boolean updateable = this.repository.findProjectOfAnAssignmentByAssignmentId
55             (assignment.getId()).isDraft();
56         if (!super.getBuffer().getErrors().hasErrors("userStory")) {
57             boolean condition = assignment.getProject().getManager().getId() ==
58                 assignment.getUserStory().getManager().getId();
59             super.state(condition, "*", "manager.project.form.error.ownership");
60         }
61     }
62 }
```

```
60
61     if (!super.getBuffer().getErrors().hasErrors("project")) {
62         super.state(updateable, "*", "manager.project.form.updateable");
63         super.state(assignment.getProject().isDraft() == true, "*",
"manager.project.form.create-denied");
64     }
65
66     if (!super.getBuffer().getErrors().hasErrors("userStory")) {
67         int projectId = assignment.getProject().getId();
68         int userStoryId = assignment.getUserStory().getId();
69
70         Collection<Assignment> existingAssignments =
this.repository.findAssignmentProjectUserStory(projectId, userStoryId);
71
72         boolean isListSizeValid = existingAssignments.size() <= 1;
73
74         boolean isEmptyCollection = existingAssignments.isEmpty();
75
76         boolean isSameAssignment = existingAssignments.size() == 1 &&
existingAssignments.iterator().next().getId() == assignment.getId();
77
78         super.state(isListSizeValid && (isEmptyCollection || isSameAssignment),
"userStory", "manager.project.form.usDuplicated");
79     }
80
81 }
82
83 @Override
84 public void perform(final Assignment assignment) {
85     assert assignment != null;
86
87     this.repository.save(assignment);
88 }
89
90 @Override
91 public void unbind(final Assignment assignment) {
92     assert assignment != null;
93
94     Dataset dataset;
95
96     int id = super.getRequest().getPrincipal().getActiveRoleId();
97     SelectChoices userStoriesChoices;
98     Collection<UserStory> userStories =
this.repository.findAllUserStoriesOfAManagerById(id);
99
100     userStoriesChoices = SelectChoices.from(userStories, "title",
assignment.getUserStory());
101
102     dataset = super.unbind(assignment, "project.title", "userStory");
103
104     dataset.put("masterId", assignment.getProject().getId());
105     dataset.put("userStories", userStoriesChoices);
106
107     super.getResponse().addData(dataset);
108 }
109
110 }
111
```