

```
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.Project;
14 import acme.entities.project.UserStory;
15 import acme.features.manager.project.ManagerProjectRepository;
16 import acme.roles.Manager;
17
18 @Service
19 public class ManagerAssignmentCreateService extends AbstractService<Manager, Assignment> {
20
21     @Autowired
22     private ManagerProjectRepository repository;
23
24
25     @Override
26     public void authorise() {
27         boolean status = super.getRequest().getPrincipal().hasRole(Manager.class);
28
29         super.getResponse().setAuthorised(status);
30     }
31
32     @Override
33     public void load() {
34
35         Assignment assignment = new Assignment();
36         super.getBuffer().addData(assignment);
37
38     }
39
40     @Override
41     public void bind(final Assignment assignment) {
42         assert assignment != null;
43
44         super.bind(assignment, "project", "userStory");
45     }
46
47     @Override
48     public void validate(final Assignment assignment) {
49         assert assignment != null;
50
51         if (!super.getBuffer().getErrors().hasErrors("project")) {
52             super.state(!assignment.getProject().isHasFatalErrors(), "project",
53 "manager.project.form.error.fatal-errors");
54             super.state(assignment.getProject().isDraft() == true, "*",
55 "manager.project.form.create-denied");
56         }
57
58     }
59
60     @Override
61     public void perform(final Assignment assignment) {
62         assert assignment != null;
```

```
61
62     this.repository.save(assignment);
63 }
64
65 @Override
66 public void unbind(final Assignment assignment) {
67     assert assignment != null;
68
69     Dataset dataset;
70
71     int id = super.getRequest().getPrincipal().getActiveRoleId();
72     SelectChoices projectChoices;
73     SelectChoices userStoriesChoices;
74
75     Collection<Project> projects = this.repository.findAllProjectsByManagerId(id);
76     Collection<UserStory> userStories = this.repository.findAllUserStoriesOfAManagerById
77 (id);
78     projectChoices = SelectChoices.from(projects, "title", assignment.getProject());
79     userStoriesChoices = SelectChoices.from(userStories, "title", assignment.getUserStory
80 ());
81     dataset = super.unbind(assignment, "project", "userStory");
82
83     dataset.put("projects", projectChoices);
84     dataset.put("userStories", userStoriesChoices);
85
86     super.getResponse().addData(dataset);
87 }
88
89 }
90
```