

```
1
2 package acme.features.manager.project;
3
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Service;
6
7 import acme.client.services.AbstractService;
8 import acme.entities.project.Project;
9 import acme.roles.Manager;
10
11 @Service
12 public class ManagerProjectDeleteService extends AbstractService<Manager, Project> {
13
14     // Internal state -----
15
16     @Autowired
17     private ManagerProjectRepository repository;
18
19     // AbstractService interface -----
20
21     @Override
22     public void authorise() {
23         boolean status;
24         int masterId;
25         Project project;
26
27         masterId = super.getRequest().getData("id", int.class);
28         project = this.repository.findProjectById(masterId);
29         status = project != null && project.isDraft() && super.getRequest().getPrincipal
30             ().hasRole(Manager.class) && project.getManager().getId() == super.getRequest().getPrincipal
31             ().getActiveRoleId();
32         super.getResponse().setAuthorised(status);
33     }
34
35     @Override
36     public void load() {
37         Project project;
38         int id;
39
40         id = super.getRequest().getData("id", int.class);
41         project = this.repository.findProjectById(id);
42
43         super.getBuffer().addData(project);
44     }
45
46     @Override
47     public void bind(final Project project) {
48         assert project != null;
49
50         super.bind(project, "code", "title", "abstractText", "hasFatalErrors", "cost",
51             "link", "isDraft");
52     }
53
54     @Override
55     public void validate(final Project project) {
56         assert project != null;
57     }
58
59     @Override
60     public void perform(final Project project) {
```

```
60     assert project != null;
61
62     this.repository.deleteAll(this.repository.findAllAssignmentsOfAProjectById
(project.getId()));
63     this.repository.deleteAll(this.repository.findAllProgressLogsByProjectId
(project.getId()));
64     this.repository.deleteAll(this.repository.findAllContractOfAProjectById
(project.getId()));
65
66     this.repository.deleteAll(this.repository.findAllAuditRecordsOfAProjectById
(project.getId()));
67     this.repository.deleteAll(this.repository.findAllCodeAuditsOfAProjectById
(project.getId()));
68
69     this.repository.deleteAll(this.repository.findAllSponsorShipOfAProjectById
(project.getId()));
70     this.repository.deleteAll(this.repository.findAllObjectivesOfAProjectById
(project.getId()));
71
72     this.repository.deleteAll(this.repository.findAllTrainingSessionsOfAProjectById
(project.getId()));
73     this.repository.deleteAll(this.repository.findAllTrainingModuleOfAProjectById
(project.getId()));
74     this.repository.delete(project);
75 }
76
77 }
78
```