

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

```
60 }
61
62 @Override
63 public void perform(final UserStory userStory) {
64     assert userStory != null;
65
66     this.uSRepository.save(userStory);
67 }
68
69 @Override
70 public void unbind(final UserStory userStory) {
71     assert userStory != null;
72     SelectChoices priorities = SelectChoices.from(Priority.class, userStory.getPriority
73 ());
74     Dataset dataset;
75
76     dataset = super.unbind(userStory, "title", "description", "estimatedCost",
77 "priority", "acceptanceCriteria", "link", "isDraft");
78     dataset.put("priorities", priorities);
79     dataset.put("isDraft", userStory.isDraft());
80     super.getResponse().addData(dataset);
81 }
82 }
```