

```
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     @Override
25     public void authorise() {
26         boolean status;
27         int masterId;
28         UserStory userStory;
29
30         masterId = super.getRequest().getData("id", int.class);
31         userStory = this.uSRepository.findUserStoryById(masterId);
32         status = userStory != null && userStory.isDraft() && super.getRequest().getPrincipal
33             ().hasRole(Manager.class) && userStory.getManager().getId() == super.getRequest
34             ().getPrincipal().getActiveRoleId();
35         super.getResponse().setAuthorised(status);
36     }
37
38     @Override
39     public void load() {
40         UserStory userStory;
41         int id;
42
43         id = super.getRequest().getData("id", int.class);
44         userStory = this.uSRepository.findUserStoryById(id);
45
46         super.getBuffer().addData(userStory);
47     }
48
49     @Override
50     public void bind(final UserStory userStory) {
51         assert userStory != null;
52
53         super.bind(userStory, "title", "description", "estimatedCost", "priority",
54             "acceptanceCriteria", "link", "isDraft");
55     }
56
57     @Override
58     public void validate(final UserStory userStory) {
59         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 }
```