

Alexander Andonov,
Ariel Potolski Eilat,
Alan Kuznicki,
Bart Coster,
Kasper van Duijne,
Vladimir Pavlov

Assignment 3 - Group 11b

Task 1: Automated Mutation Testing

1. Class 1: RequestAllocationService

Location: Request Microservice -> services

Why this class?

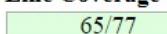
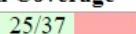
The client's requirements can only be adequately satisfied when the integral part of the system works, where a user can submit requests in order to use part of Delft Blue's supercomputer. That is why this service, which handles the incoming requests' allocation from the controller and should be fully tested.

Coverage Before:

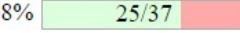
Pit Test Coverage Report

Package Summary

`nl.tudelft.sem.template.example.services`

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---|---|
| 1 | 84%  | 68%  |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|--|---|--|
| <code>RequestAllocationService.java</code> | 84%  | 68%  |

Report generated by [PIT](#) 1.5.1

```

58 /**
59  * This method is responsible for getting the associated faculties with the user who made the request.
60  * It receives that info through an endpoint from the user service
61  *
62  * @param token the token
63  * @return the faculty user faculties
64  */
65 public List<String> getFacultyUserFaculties(String token) {
66
67     try {
68         String url = "http://localhost:8081/getUserFaculties";
69         HttpHeaders headers = new HttpHeaders();
70         headers.setContentType(MediaType.APPLICATION_JSON);
71         headers.setBearerAuth(token);
72
73         HttpEntity<TokenRequestModel> entity = new HttpEntity<>(new TokenRequestModel(token), headers);
74         ResponseEntity<FacultiesResponseModel> result = restTemplate
75             .postForEntity(url, entity, FacultiesResponseModel.class);
76
77         String string = result.getBody().getFaculties()
78             .replace("[", "").replace("]", "");
79
80         assert string != null;
81         if (string.equals("")) {
82             return new ArrayList<>();
83         }
84     }
85
86     // extract
87
88     return Arrays.stream(string.split(", ")).collect(Collectors.toList());
89 } catch (Exception e) {
90     System.out.println("error with post:" + e);
91 }
92
93 return new ArrayList<>();
94
95 }
96
97 /**
98  * This method is responsible for getting the available resources for a particular faculty up until a given date.
99  * If a job can be executed prior to that date, it will be done earlier.
100 *
101 * @param facultyName the faculty name
102 * @param preferredDate the preferred date
103 * @return the list
104 */
105 public List<Resource> getReservedResource(String facultyName, LocalDate preferredDate, String token) {
106     try {
107         String url = "http://localhost:8082/resources/availableUntil/"
108             + preferredDate.format(DateTimeFormatter.ofPattern("yyyy-MM-dd"))
109             + "/" + facultyName;
110
111         HttpHeaders headers = new HttpHeaders();
112         headers.setContentType(MediaType.APPLICATION_JSON);
113         headers.setBearerAuth(token);
114
115         HttpEntity<TokenRequestModel> entity = new HttpEntity<>(new TokenRequestModel(token), headers);
116         var result = restTemplate.exchange(url, HttpMethod.GET, entity, ResourceResponseModel[].class);
117         var listOfResources = Stream.of(Objects.requireNonNull(result.getBody()))
118             .map(x -> new Resource(x.getFacultyName(), x.getResourceCpu(),
119                 x.getResourceGpu(), x.getResourceMemory())).collect(Collectors.toList());
120
121         return listOfResources;
122     } catch (Exception e) {
123         System.err.println("Error getting reserved resources: " + e.getMessage());
124         return new ArrayList<>();
125     }
126 }
127
128 /**
129  * Checks if there are enough computational resources for a given job to be executed.
130  *
131  * @param request the request
132  * @return the boolean
133  */
134 public boolean enoughResourcesForJob(Request request, String token) {
135
136     List<Resource> resources = getReservedResource(request.getFaculty(), request.getPreferredDate(), token);
137
138     for (int i = 0; i < resources.size(); i++) {
139         Resource currentResource = resources.get(i);
140
141         if (currentResource.getResourceCpu() >= request.getCpu()
142             && currentResource.getResourceGpu() >= request.getGpu()
143             && currentResource.getResourceMemory() >= request.getMemory()) {
144             return true;
145         }
146     }
147
148     return false;
149 }
150
151 }
152
153 }
154

```

```

156 /**
157 * If the user is verified and there are enough resources available, the job request is forwarded to the cluster service.
158 *
159 * @param request the request
160 * @throws JsonProcessingException the json processing exception
161 */
162 public boolean sendRequestToCluster(Request request, String token) throws JsonProcessingException {
163
164     try {
165         String url = "http://localhost:8082/request";
166         HttpHeaders headers = new HttpHeaders();
167         headers.setContentType(MediaType.APPLICATION_JSON);
168         headers.setBearerAuth(token);
169
170         HttpEntity<JobRequestRequestModel> entity =
171             new HttpEntity<>(JobRequestRequestModel.convertToRequestModel(request), headers);
172         ResponseEntity<String> result = restTemplate.postForEntity(url, entity, String.class);
173         if (result.getBody().equals("ok")) {
174             return true;
175         }
176     } catch (Exception e) {
177         System.out.println("error with post: " + e);
178         return false;
179     }
180     return true;
181
182
183 }
184
185 /**
186 * This method is responsible for notifying the user if their job request is declined for some reason.
187 * It sends the job notification to the user service.
188 *
189 * @param request the request
190 */
191 public boolean sendDeclinedRequestToUserService(Request request, String token) {
192
193     try {
194
195         var model = new NotificationRequestModel(
196             request.getPreferredDate().format(DateTimeFormatter.ofPattern("yyyy-MM-dd")),
197             "REQUEST", "REJECTED",
198             request.getDescription(), request.getNetId());
199
200         HttpHeaders headers = new HttpHeaders();
201         headers.setContentType(MediaType.APPLICATION_JSON);
202         headers.setBearerAuth(token);
203
204         String url = "http://localhost:8081/notification";
205
206         HttpEntity<NotificationRequestModel> entity = new HttpEntity<>(model, headers);
207         ResponseEntity<String> result = restTemplate.postForEntity(url, entity, String.class);
208         if (result.getBody().equals("ok")) {
209             return true;
210         }
211     } catch (Exception e) {
212         System.out.println("error with post" + e);
213         return false;
214     }
215
216     return true;
217
218
219 }
220
221 }

```

Mutations

```

21 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
22 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth + KILLED
23 1. negated conditional → KILLED
24 1. replaced return value with Collections.emptyList for nl/tudelft/semp/template/example/services/RequestAllocationService::getFacultyUserFaculties → NO_COVERAGE
25 1. replaced return value with Collections.emptyList for nl/tudelft/semp/template/example/services/RequestAllocationService::getFacultyUserFaculties → KILLED
26 1. removed call to java/io/PrintStream::println → SURVIVED
27 1. replaced return value with Collections.emptyList for nl/tudelft/semp/template/example/services/RequestAllocationService::getFacultyUserFaculties → SURVIVED
28 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
29 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth + KILLED
30 1. replaced return value with null for nl/tudelft/semp/template/example/services/RequestAllocationService::lambda$getReservedResource$0 → KILLED
31 1. replaced return value with Collections.emptyList for nl/tudelft/semp/template/example/services/RequestAllocationService::getReservedResource → KILLED
32 1. removed call to java/io/PrintStream::println → NO_COVERAGE
33 1. replaced return value with Collections.emptyList for nl/tudelft/semp/template/example/services/RequestAllocationService::getReservedResource → NO_COVERAGE
34 1. changed conditional boundary → KILLED
35 2. negated conditional → KILLED
36 1. changed conditional boundary → KILLED
37 2. negated conditional → KILLED
38 1. changed conditional boundary → KILLED
39 2. negated conditional → KILLED
40 1. changed conditional boundary → KILLED
41 2. negated conditional → KILLED
42 1. replaced boolean return with false for nl/tudelft/semp/template/example/services/RequestAllocationService::enoughResourcesForJob → KILLED
43 1. replaced boolean return with true for nl/tudelft/semp/template/example/services/RequestAllocationService::enoughResourcesForJob → KILLED
44 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
45 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth + KILLED
46 1. negated conditional → KILLED
47 1. replaced boolean return with false for nl/tudelft/semp/template/example/services/RequestAllocationService::sendRequestToCluster → KILLED
48 1. removed call to java/io/PrintStream::println → NO_COVERAGE
49 1. replaced boolean return with true for nl/tudelft/semp/template/example/services/RequestAllocationService::sendRequestToCluster → NO_COVERAGE
50 1. replaced boolean return with false for nl/tudelft/semp/template/example/services/RequestAllocationService::sendRequestToCluster → KILLED
51 1. removed call to org/springframework/http/HttpHeaders::setContentType → SURVIVED
52 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth + SURVIVED
53 1. negated conditional → SURVIVED
54 1. replaced boolean return with false for nl/tudelft/semp/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → KILLED
55 1. removed call to java/io/PrintStream::println → NO_COVERAGE
56 1. replaced boolean return with true for nl/tudelft/semp/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → NO_COVERAGE
57 1. replaced boolean return with false for nl/tudelft/semp/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → KILLED

```

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- NEGATE_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

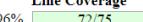
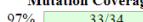
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestControllerTest] [method:sendRequestExactly@1Test()] (853 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequestForTransitTest()] (898 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:sendDeclinedRequestToUserService2@1] (17 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:getFacultyUserFacultiesNoneTest()] (21 ms)
- nl.tudelft.sem.template.example.authentication.AuthManagerTests [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.authentication.AuthManagerTests] [method:getNetIdTest()] (14 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:getFacultyUserFacultiesTest()] (15 ms)
- nl.tudelft.sem.template.example.integration.ApprovalRequestServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.ApprovalRequestServiceTest] [method:sendRequestForTodayDateTest()] (1132 ms)
- nl.tudelft.sem.template.example.integration.BonusAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.BonusAllocationServiceTest] [method:sendEnoughMemoryResourceForJobTest()] (17 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequest501MinBeforeNextDayTest@1] (1098 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:sendRequestToClusterTest2@1] (32 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:notEnoughCPUResourceForJobTest()] (20 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:getReservedResourceTest()] (20 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:sendRequest501MinBeforeNextDayTest@1] (1209 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequest6001hToMidnightTest@1] (561 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequestWaitingApprovalTest@1] (692 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequestLessThan5MinBeforeNextDayTest@1] (734 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequestTestNotInFaculty@1] (733 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:sendDeclineRequestToUserService1@1] (20 ms)
- nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.RequestAllocationServiceTest] [method:notEnoughMemoryResourceForJobTest@1] (31 ms)
- nl.tudelft.sem.template.example.integration.SendingRequestControllerTest [engine:junit-jupiter] [class:nl.tudelft.sem.template.example.integration.SendingRequestsControllerTest] [method:sendRequest601hTomorrowNotPreferredTest@1] (785 ms)

Coverage After:

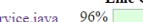
Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.example.services

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|--|--|
| 1 | 96%  72/75 | 97%  33/34 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|-------------------------------|---|---|
| RequestAllocationService.java | 96%  72/75 | 97%  33/34 |

Report generated by [PIT](#) 1.5.1

```

59 /**
60  * This method is responsible for getting the associated faculties with the user who made the request.
61  * It receives that info through an endpoint from the user service
62  *
63  * @param token the token
64  * @return the faculty user faculties
65  */
66 public List<String> getFacultyUserFaculties(String token) {
67
68     try {
69         String url = "http://localhost:8081/getUserFaculties";
70         HttpHeaders headers = new HttpHeaders();
71         1 headers.setContentType(MediaType.APPLICATION_JSON);
72         1 headers.setBearerAuth(token);
73
74         HttpEntity<TokenRequestModel> entity = new HttpEntity<>(new TokenRequestModel(token), headers);
75         ResponseEntity<FacultiesResponseModel> result = restTemplate
76             .postForEntity(url, entity, FacultiesResponseModel.class);
77
78         String string = result.getBody().getFaculties()
79             .replace("[", "").replace("]", "");
80
81         if (string.equals("")) {
82             1 return Collections.emptyList();
83         }
84
85         1 return Arrays.stream(string.split(", ")).collect(Collectors.toList());
86
87     } catch (Exception e) {
88         1 System.out.println("error with post:" + e);
89     }
90     return Collections.emptyList();
91
92     }

```

```

94 /**
95  * This method is responsible for getting the available resources for a particular faculty up until a given date.
96  * If a job can be executed prior to that date, it will be done earlier.
97  *
98  * @param facultyName the faculty name
99  * @param preferredDate the preferred date
100 * @return the list
101 */
102 public List<Resource> getReservedResource(String facultyName, LocalDate preferredDate, String token) {
103     try {
104         String url = "http://localhost:8082/resources/availableUntil/"
105             + preferredDate.format(DateTimeFormatter.ofPattern("yyyy-MM-dd"))
106             + "/" + facultyName;
107
108         HttpHeaders headers = new HttpHeaders();
109         headers.setContentType(MediaType.APPLICATION_JSON);
110         headers.setBearerAuth(token);
111
112         HttpEntity<TokenRequestModel> entity = new HttpEntity<>(new TokenRequestModel(token), headers);
113         var result = restTemplate.exchange(url, HttpMethod.GET, entity, ResourceResponseModel[].class);
114         var listOfResources = Stream.of(Objects.requireNonNull(result.getBody()))
115             .map(x -> new Resource(x.getFacultyName(), x.getResourceCpu(),
116                                     x.getResourceGpu(), x.getResourceMemory())).collect(Collectors.toList());
117
118         return listOfResources;
119     } catch (Exception e) {
120         System.out.println("Error getting reserved resources: " + e.getMessage());
121         return Collections.emptyList();
122     }
123 }
124
125 /**
126  * Checks if there are enough computational resources for a given job to be executed.
127  *
128  * @param request the request
129  * @return the boolean
130  */
131 public boolean enoughResourcesForJob(Request request, String token) {
132
133     List<Resource> resources = getReservedResource(request.getFaculty(), request.getPreferredDate(), token);
134
135     for (int i = 0; i < resources.size(); i++) {
136         Resource currentResource = resources.get(i);
137
138         if (currentResource.getResourceCpu() >= request.getCpu()
139             && currentResource.getResourceGpu() >= request.getGpu()
140             && currentResource.getResourceMemory() >= request.getMemory()) {
141             return true;
142         }
143     }
144     return false;
145 }
146
147 }
148
149 }
150 }
151
152 /**
153  * If the user is verified and there are enough resources available, the job request is forwarded to the cluster service.
154  *
155  * @param request the request
156  */
157 public boolean sendRequestToCluster(Request request, String token) {
158
159     try {
160         String url = "http://localhost:8082/request";
161         HttpHeaders headers = new HttpHeaders();
162         headers.setContentType(MediaType.APPLICATION_JSON);
163         headers.setBearerAuth(token);
164
165         HttpEntity<JobRequestRequestModel> entity =
166             new HttpEntity<>(JobRequestRequestModel.convertToRequestModel(request), headers);
167         ResponseEntity<String> result = restTemplate.postForEntity(url, entity, String.class);
168
169         if (result.getBody().equals("ok")) {
170             return true;
171         }
172     } catch (Exception e) {
173         System.out.println("error with post: " + e);
174         return false;
175     }
176     return true;
177
178 }
179
180 }
```

```

181 /**
182  * This method is responsible for notifying the user if their job request is declined for some reason.
183  * It sends the job notification to the user service.
184  *
185  * @param request the request
186  */
187 public boolean sendDeclinedRequestToUserService(Request request, String token) {
188
189     try {
190
191         var model = new NotificationRequestModel(
192             request.getPreferredDate().format(DateTimeFormatter.ofPattern("yyyy-MM-dd")),
193             "REQUEST", "REJECTED",
194             request.getDescription(), request.getNetId());
195
196
197         HttpHeaders headers = new HttpHeaders();
198         headers.setContentType(MediaType.APPLICATION_JSON);
199         headers.setBearerAuth(token);
200
201         String url = "http://localhost:8081/notification";
202
203         HttpEntity<NotificationRequestModel> entity = new HttpEntity<>(model, headers);
204         ResponseEntity<String> result = restTemplate.postForEntity(url, entity, String.class);
205         if (result.getBody().equals("ok")) {
206             return true;
207         }
208     } catch (Exception e) {
209         System.out.println("error with post: " + e);
210         return false;
211     }
212     return false;
213
214 }
215
216 }
217 }
```

Mutations

```

71 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
72 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth → KILLED
81 1. negated conditional → KILLED
85 1. replaced return value with Collections.emptyList for nl/tudelft/semt/template/example/services/RequestAllocationService::getFacultyUserFaculties → KILLED
88 1. removed call to java/io/PrintStream::println → KILLED
109 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
110 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth → KILLED
115 1. replaced return value with null for nl/tudelft/semt/template/example/services/RequestAllocationService::lambda$getReservedResource$0 → KILLED
118 1. replaced return value with Collections.emptyList for nl/tudelft/semt/template/example/services/RequestAllocationService::getReservedResource → KILLED
120 1. removed call to java/io/PrintStream::println → KILLED
136 1. changed conditional boundary → KILLED
2. negated conditional → KILLED
139 1. changed conditional boundary → KILLED
2. negated conditional → KILLED
141 1. changed conditional boundary → KILLED
2. negated conditional → KILLED
143 1. changed conditional boundary → KILLED
2. negated conditional → KILLED
144 1. replaced boolean return with false for nl/tudelft/semt/template/example/services/RequestAllocationService::enoughResourcesForJob → KILLED
148 1. replaced boolean return with true for nl/tudelft/semt/template/example/services/RequestAllocationService::enoughResourcesForJob → KILLED
163 1. removed call to org/springframework/http/HttpHeaders::setContentType → KILLED
164 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth → KILLED
169 1. negated conditional → KILLED
170 1. replaced boolean return with false for nl/tudelft/semt/template/example/services/RequestAllocationService::sendRequestToCluster → KILLED
172 1. removed call to java/io/PrintStream::println → KILLED
174 1. replaced boolean return with true for nl/tudelft/semt/template/example/services/RequestAllocationService::sendRequestToCluster → KILLED
176 1. replaced boolean return with false for nl/tudelft/semt/template/example/services/RequestAllocationService::sendRequestToCluster → KILLED
198 1. removed call to org/springframework/http/HttpHeaders::setContentType → SURVIVED
199 1. removed call to org/springframework/http/HttpHeaders::setBearerAuth → KILLED
205 1. negated conditional → KILLED
206 1. replaced boolean return with false for nl/tudelft/semt/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → KILLED
209 1. removed call to java/io/PrintStream::println → KILLED
210 1. replaced boolean return with true for nl/tudelft/semt/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → KILLED
212 1. replaced boolean return with true for nl/tudelft/semt/template/example/services/RequestAllocationService::sendDeclinedRequestToUserService → KILLED
```

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- BYPASS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

```

• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequestToClusterTestException()](13 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequestExactly6hTest()](779 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequestLessThan6hTest()](743 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendDeclinedRequestToUserService2()](13 ms)
• nl.tudelft.semt.example.authentication.AuthManagerTests.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.authentication.AuthManagerTests].[method:getNetIdTest()](7 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getFacultyUserFacultiesNoneTest()](16 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getFacultyUserFacultiesTest()](16 ms)
• nl.tudelft.semt.example.integration.ApprovingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.ApprovingRequestsControllerTest].[method:sendRequestWaitingApprovalTest()](959 ms)
• nl.tudelft.semt.example.integration.ApprovingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.ApprovingRequestsControllerTest].[method:sendRequestWaitingApprovalTest()](12 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest50MinBeforeNextDayTest()](991 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequestToClusterTest2()](17 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:notEnoughpuResourceForJobTest()](20 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getReservedResourceTest()](14 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest50MinBeforeNextDayTest()](693 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest72hTest()](732 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequestWaitingApprovalTest()](668 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest75MinBeforeNextDayTest()](765 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendDeclinedRequestToUserService1()](20 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:notEnoughMemoryResourceForJobTest()](17 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequestEst75MinBeforeNextDayTest()](765 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getFacultyUserFacultiesRandomResult()](14 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest5MinBeforeNextDayTest()](837 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest55MinBeforeNextDayTest()](656 ms)
• nl.tudelft.semt.example.integration.SendingRequestsControllerTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest].[method:sendRequest70hTest()](740 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getReservedResourceExceptionTest()](14 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequestForTodayDate()](825 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequestToClusterTest1()](15 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:sendRequests60hToMidnightTest()](687 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getFacultyUserFacultiesErrorTest()](26 ms)
• nl.tudelft.semt.example.integration.RequestAllocationServiceTest.[engine:junit-jupiter].[class:nl.tudelft.semt.template.example.integration.RequestAllocationServiceTest].[method:getFacultyUserFacultiesExceptionThrown()](15 ms)
```

Report:

The mutation coverage went from a 68% coverage to a 97% coverage. This is a successful testing-refactoring as the mutation coverage has increased by more than 20%, with the positive side effect that the line coverage has also gone up to 96%.

2. Class 2: NodeContributionService

Location: Cluster Microservice

Coverage Before Adding Tests

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.cluster.domain.services

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---------------|-------------------|
| 7 | 78% 234/300 | 79% 119/151 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|--|---------------|-------------------|
| JobReschedulingService.java | 95% 54/57 | 100% 31/31 |
| JobSchedulingService.java | 60% 28/47 | 68% 13/19 |
| NodeContributionService.java | 58% 19/33 | 44% 4/9 |
| NodeDataProcessingService.java | 100% 15/15 | 92% 11/12 |
| NotificationManagerService.java | 69% 18/26 | 63% 5/8 |
| PrivilegeVerificationService.java | 43% 16/37 | 15% 3/20 |
| SchedulingDataProcessingService.java | 99% 84/85 | 100% 52/52 |

Report generated by [PIT](#) 1.5.1

Why this class?

This class is vital to our system since it handles the Node contribution and removal functionalities of our service. It follows that it should be one of our priorities to increase its reliability by improving its coverage against mutations.

Coverage After Adding Tests

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.cluster.domain.services

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---------------|-------------------|
| 7 | 79% 238/302 | 81% 123/151 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|--|---------------|-------------------|
| JobReschedulingService.java | 95% 54/57 | 100% 31/31 |
| JobSchedulingService.java | 60% 28/47 | 68% 13/19 |
| NodeContributionService.java | 66% 23/35 | 78% 7/9 |
| NodeDataProcessingService.java | 100% 15/15 | 100% 12/12 |
| NotificationManagerService.java | 69% 18/26 | 63% 5/8 |
| PrivilegeVerificationService.java | 43% 16/37 | 15% 3/20 |
| SchedulingDataProcessingService.java | 99% 84/85 | 100% 52/52 |

Report generated by [PIT](#) 1.5.1

Mutations list after adding tests

Mutations

| | |
|-----|--|
| 64 | 1. replaced return value with Collections.emptyList for nl/tudelft/sem/template/cluster/domain/services/NodeContributionService::getNodesToRemove → KILLED |
| 73 | 1. replaced int return with 0 for nl/tudelft/sem/template/cluster/domain/services/NodeContributionService::numberOfNodesToRemove → KILLED |
| 106 | 1. removed call to nl/tudelft/sem/template/cluster/domain/services/NodeDataProcessingService::save → KILLED |
| 115 | 1. removed call to nl/tudelft/sem/template/cluster/domain/services/NodeDataProcessingService::assignNodeToFaculty → KILLED |
| 125 | 1. removed call to nl/tudelft/sem/template/cluster/domain/services/NodeContributionService::assignNodeToFaculty → KILLED |
| 128 | 1. removed call to nl/tudelft/sem/template/cluster/domain/services/NodeDataProcessingService::save → KILLED |
| 138 | 1. replaced boolean return with false for nl/tudelft/sem/template/cluster/domain/services/NodeContributionService::addNodeToBeRemoved → KILLED |
| 151 | 1. removed call to nl/tudelft/sem/template/cluster/domain/services/NodeDataProcessingService::deleteNode → NO_COVERAGE |
| 155 | 1. removed call to org/springframework/context/ApplicationEventPublisher::publishEvent → NO_COVERAGE |

3. Class 3: ApprovingRequestsController

Why is this class critical?

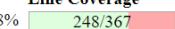
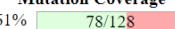
The class contains the methods that manage the pending requests and sends approvals, if they meet the criteria to be committed to the cluster. It acts as a functional bridge between the class where new requests are created and the cluster microservice, where requests are handled. Therefore, it's of critical importance that this class is thoroughly tested, otherwise sent requests may have an issue being approved and they might never reach the cluster.

What was the mutation coverage beforehand?

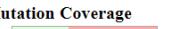
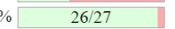
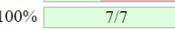
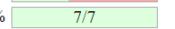
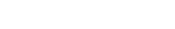
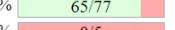
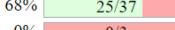
We ran the Pitest tool to automatically test against mutants and the report showed initial coverage of only 10% for the ApprovingRequestsController. We also examined the line specific mutation causes, in order to identify which part of the functionality of the methods in the class need to be tested specifically. The methods that lacked mutation resilience were the handleRequests and the sendApprovals methods.

Pit Test Coverage Report

Project Summary

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---|--|
| 18 | 68%  248/367 | 61%  78/128 |

Breakdown by Package

| Name | Number of Classes | Line Coverage | Mutation Coverage |
|--|-------------------|---|---|
| nl.tudelft.sem.template.example.authentication | 4 | 35%  15/43 | 39%  9/23 |
| nl.tudelft.sem.template.example.config | 2 | 96%  26/27 | 100%  7/7 |
| nl.tudelft.sem.template.example.controllers | 2 | 73%  52/71 | 67%  22/33 |
| nl.tudelft.sem.template.example.domain | 8 | 63%  90/144 | 60%  15/25 |
| nl.tudelft.sem.template.example.services | 1 | 84%  65/77 | 68%  25/37 |
| nl.tudelft.sem.template.example.util | 1 | 0%  0/5 | 0%  0/3 |

Report generated by [PIT](#) 1.5.1

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.example.controllers

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---------------|-------------------|
| 2 | 73% 52/71 | 67% 22/33 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|----------------------------------|---------------|-------------------|
| ApprovingRequestsController.java | 27% 7/26 | 10% 1/10 |
| SendingRequestsController.java | 100% 45/45 | 91% 21/23 |

Report generated by [PIT](#) 1.5.1

```
27  /**
28   * Instantiates a new controller.
29   *
30   * @param authManager      Spring Security component used to authenticate and authorize the user
31   * @param requestAllocationService the request allocation service
32   * @param requestRepository    the request repository
33   */
34   @Autowired
35   public ApprovingRequestsController(AuthManager authManager, RequestAllocationService requestAllocationService,
36                                     RequestRepository requestRepository) {
37     this.authManager = authManager;
38     this.requestAllocationService = requestAllocationService;
39     this.requestRepository = requestRepository;
40   }
41
42 /**
43  * This endpoint is responsible for broadcasting all pending requests,
44  * so a faculty-privileged account can approve/decline them.
45  *
46  * @return the response entity
47  */
48 @GetMapping("pendingRequests")
49 @PreAuthorize("hasRole('FACULTY')")
50 public ResponseEntity<List<Request>> publishRequest() {
51   List<Request> requests = requestRepository.findAllByApprovedIs(false);
52   return ResponseEntity.status(HttpStatus.OK).body(requests);
53 }
54
55 /**
56  * This endpoint is responsible for accepting the ids of approved requests,
57  * and see if the sender is legitimate and from the respective faculty.
58  * Afterwards, the requests are sent to the cluster.
59  *
60  * @param headers          the headers
61  * @param approvalInformation the approval information
62  * @return                  the response entity
63  * @throws JsonProcessingException the json processing exception
64  */
65
66 /**
67  * This endpoint is responsible for accepting the ids of approved requests,
68  * and see if the sender is legitimate and from the respective faculty.
69  * Afterwards, the requests are sent to the cluster.
70  *
71  * @param headers          the headers
72  * @param approvalInformation the approval information
73  * @return                  the response entity
74  * @throws JsonProcessingException the json processing exception
75  */
76 @PostMapping("sendApprovals")
77 @PreAuthorize("hasRole('FACULTY')")
78 @SuppressWarnings("PMD.DataflowAnalysis")
79 public ResponseEntity<List<Request>> sendApprovals(@RequestHeader HttpHeaders headers,
80                                                       @RequestBody ApprovalInformation approvalInformation)
81   throws JsonProcessingException {
82   //I require a file with the ids of all approved requests, check if the sender is with a faculty profile
83
84   String token = headers.get("Authorization").get(0).replace("Bearer ", "");
85   List<String> facultiesOfFacultyUser = requestAllocationService
86     .getFacultyUserFaculties(token);
87
88   List<Request> requests = requestRepository.findAll().stream()
89     .filter(x -> Utils.idsContained(approvalInformation.getIds(), x.getId()))
90     .filter(x -> facultiesOfFacultyUser.contains(x.getFaculty()))
91     .collect(Collectors.toList());
92
93   handleRequests(requests, token);
94   return ResponseEntity.ok().body(requests);
95 }
```

```

96     }
97
98
99     /**
100      * Synthesizes and extracts all the request logic from the sendApprovals method.
101      * It is done for better code quality and readability.
102      *
103      * @param requests list of requests
104      * @param token token of the current user
105      * @throws JsonProcessingException the json processing exception
106      */
107     public void handleRequests(List<Request> requests, String token)
108         throws JsonProcessingException {
109
110         for (Request request : requests) {
111             request.setApproved(true);
112         }
113         //Implementation of changing the status of respective requests to approved
114         for (Request request : requests) {
115             if (requestAllocationService.enoughResourcesForJob(request, token)) {
116                 requestAllocationService.sendRequestToCluster(request, token);
117             } else {
118                 // Users team must open an endpoint for post requests, notifying for declined requests
119                 requestAllocationService.sendDeclinedRequestToUserService(request, token);
120             }
121         }
122         // Deleting approved and sent entities
123         requestRepository.deleteAll(requests);
124     }
125
126 }

```

Mutations

```

63 1. replaced return value with null for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::publishRequest → KILLED
64 1. replaced boolean return with false for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$0 → NO_COVERAGE
65 1. replaced boolean return with true for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$0 → NO_COVERAGE
66 1. replaced boolean return with false for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$1 → NO_COVERAGE
67 2. replaced boolean return with true for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$1 → NO_COVERAGE
68 1. removed call to nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::handleRequests → NO_COVERAGE
69 1. replaced return value with null for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::sendApprovals → NO_COVERAGE
70 1. removed call to nl/tudelft/semt/template/example/domain/Request::setApproved → NO_COVERAGE
71 1. negated conditional → NO_COVERAGE
72 1. removed call to nl/tudelft/semt/template/example/domain/RequestRepository::deleteAll → NO_COVERAGE

```

Active mutators

- BOOLEAN_FALSE_MUTATOR
- BOOLEAN_TRUE_MUTATOR
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INJECT_NIL_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequestLessThan6hNoResourceTest()] (853 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:pendingRequestTest()] (1299 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequestWaitingApprovalTest()] (821 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequestExactly6hTest()] (812 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequestMoreThan6hTest()] (808 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequestLessThan6hTomorrowNotPreferredTest()] (908 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequest501MinBeforeNextDayTest()] (849 ms)
- nl.tudelft.semt.example.authentication.AuthManagerTests [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.authentication.AuthManagerTests] [method:getNetIdTest()] (11 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequest601hToMidnightTest()] (862 ms)
- nl.tudelft.semt.example.integration.ApprovingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.ApprovingRequestsControllerTest] [method:pendingRequestTest()] (1426 ms)
- nl.tudelft.semt.example.integration.SendingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.SendingRequestsControllerTest] [method:sendRequest539ToMidnightTest()] (721 ms)
- nl.tudelft.semt.example.integration.ApprovingRequestsControllerTest [engine:junit-jupiter] [class:nl.tudelft.semt.template.example.integration.ApprovingRequestsControllerTest] [method:sendRequestWaitingApprovalTest()] (1239 ms)

What changed after we made corrections in our tests?

After we added additional tests for correctly retrieving pending requests and approving and retrieving approved requests the mutation coverage jumped to 100% with 96% line coverage. More specifically, we added test cases that include the interaction between the ApprovingRequestsController and the RequestAllocationService class that checks if there are enough resources for the completion of a given request and sends or declines the requests to the cluster. We mocked that interaction and sent pre-made requests that still don't have the approved status, then we check if they are pending and mock the sendApprovals method to retrieve them afterwards. Through these added tests we cover the potential points of mutation highlighted in the initial Pitest report.

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.example.controllers

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|--|--|
| 2 | 99% <div style="width: 99%;">70/71</div> | 94% <div style="width: 94%;">31/33</div> |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|--|--|--|
| ApprovingRequestsController.java | 96% <div style="width: 96%;">25/26</div> | 100% <div style="width: 100%;">10/10</div> |
| SendingRequestsController.java | 100% <div style="width: 100%;">45/45</div> | 91% <div style="width: 91%;">21/23</div> |

Report generated by [PIT](#) 1.5.1

```
36
37  /**
38   * Instantiates a new controller.
39   *
40   * @param authManager      Spring Security component used to authenticate and authorize the user
41   * @param requestAllocationService the request allocation service
42   * @param requestRepository    the request repository
43   */
44  @Autowired
45  public ApprovingRequestsController(AuthManager authManager, RequestAllocationService requestAllocationService,
46                                     RequestRepository requestRepository) {
47      this.authManager = authManager;
48      this.requestAllocationService = requestAllocationService;
49      this.requestRepository = requestRepository;
50  }
51
52 /**
53  * This endpoint is responsible for broadcasting all pending requests,
54  * so a faculty-privileged account can approve/decline them.
55  *
56  * @return the response entity
57  */
58 @GetMapping("pendingRequests")
59 @PreAuthorize("hasRole('FACULTY')")
60 public ResponseEntity<List<Request>> publishRequest() {
61     List<Request> requests = requestRepository.findAllByApprovedIs(false);
62     return ResponseEntity.status(HttpStatus.OK).body(requests);
63 }
64
65 /**
66  * This endpoint is responsible for accepting the ids of approved requests,
67  * and see if the sender is legitimate and from the respective faculty.
68  * Afterwards, the requests are sent to the cluster.
69  *
70  * @param headers      the headers
71  * @param approvalInformation the approval information
72  * @return            the response entity
73  * @throws JsonProcessingException the json processing exception
74  */
75 @PostMapping("sendApprovals")
76 @PreAuthorize("hasRole('FACULTY')")
77 @SuppressWarnings("PMD.DataflowAnomalyAnalysis")
78 public ResponseEntity<List<Request>> sendApprovals(@RequestHeader HttpHeaders headers,
79                                         @RequestBody ApprovalInformation approvalInformation)
80     throws JsonProcessingException {
81     //I require file with the ids of all approved requests, check if the sender is with a faculty profile
82
83     String token = headers.get("Authorization").get(0).replace("Bearer ", "");
84     List<String> facultiesOffFacultyUser = requestAllocationService
85         .getFacultyUserFaculties(token);
86
87     List<Request> requests = requestRepository.findAll().stream()
88         .filter(x -> Utils.idsContained(approvalInformation.getIds(), x.getId()))
89         .filter(x -> facultiesOffFacultyUser.contains(x.getFaculty()))
90         .collect(Collectors.toList());
91
92     handleRequests(requests, token, requestAllocationService, requestRepository);
93     return ResponseEntity.status(HttpStatus.OK).body(requests);
94 }
95 }
96 }
```

```

99
100 /**
101 * Synthesizes and extracts all the request logic from the sendApprovals method.
102 * It is done for better code quality and readability.
103 *
104 * @param requests list of requests
105 * @param token token of the current user
106 * @throws JsonProcessingException the json processing exception
107 */
108 public void handleRequests(List<Request> requests, String token,
109     RequestAllocationService requestAllocationService, RequestRepository requestRepository)
110     throws JsonProcessingException {
111
112     for (Request request : requests) {
113         request.setApproved(true);
114     }
115     //Implementation of changing the status of respective requests to approved
116     for (Request request : requests) {
117         if (requestAllocationService.enoughResourcesForJob(request, token)) {
118             requestAllocationService.sendRequestToCluster(request, token);
119         } else {
120             // Users team must open an endpoint for post requests, notifying for declined requests
121             requestAllocationService.sendDeclinedRequestToUserService(request, token);
122         }
123     }
124     // Deleting approved and sent entities
125     requestRepository.deleteAll(requests);
126 }
127
128
129
130
131
132 }

Mutations

63 1. replaced return value with null for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::publishRequest - KILLED
64 1. replaced boolean return with false for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$0 - KILLED
65 2. replaced boolean return with true for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$0 - KILLED
66 1. replaced boolean return with false for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$1 - KILLED
67 2. replaced boolean return with true for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::lambda$sendApprovals$1 - KILLED
68 1. removed call to nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::handleRequests - KILLED
69 1. replaced return value with null for nl/tudelft/semt/template/example/controllers/ApprovingRequestsController::sendApprovals - KILLED
70 1. removed call to nl/tudelft/semt/template/example/domain/Request::setApproved - KILLED
71 1. negated conditional - KILLED
72 1. removed call to nl/tudelft/semt/template/example/domain/RequestRepository::deleteAll - KILLED

```

Active mutators

- BOOLEAN_FALSE_MUTATOR
- BOOLEAN_TRUE_MUTATOR
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERSE_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

```

nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestLaterThan6h50ResourceTest()] (815 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method pendingRequestsTest()] (939 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestWaitingApprovalTest()] (693 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestExactly6hTest()] (614 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestLessThan6hTest()] (866 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestLessThan6hTomorrowNotPreferredTest()] (765 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequestLessThan6hTodayNotPreferredTest()] (765 ms)
nl.tudelft.semt.example.integration.ApprovingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.ApprovingRequestControllerTest] [method pendingRequestTest()] (1414 ms)
nl.tudelft.semt.example.integration.ApprovingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.ApprovingRequestControllerTest] [method sendRequest5597hMidnightTest()] (794 ms)
nl.tudelft.semt.example.integration.ApprovingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.ApprovingRequestControllerTest] [method pendingRequestWithTest()] (1223 ms)
nl.tudelft.semt.example.integration.ApprovingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.ApprovingRequestControllerTest] [method sendRequestWaitingApprovalTest()] (1357 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequest10MinBeforeNextDayTest()] (845 ms)
nl.tudelft.semt.example.integration.SendingRequestControllerTest [engine.junit-jupiter] [class nl.tudelft.semt.example.integration.SendingRequestControllerTest] [method sendRequest6h1h7midnightTest()] (760 ms)

```

4. Class 4: VerificationDto

Why did we choose this class?

While very small and relatively unimportant, we've noticed that this class is one of the only ones with no test coverage at all. While for almost all other classes the remaining mutations were benign or harmless, this class serves an important role in the request microservice, and none of its mutations were killed. While smaller in size than the others we have improved in this report, we nevertheless believe it to be important enough to warrant adding some test cases.

Mutation coverage improvement:

Because of the size of this class, we managed to take the mutation coverage score from 0 all the way to 100%.

Mutations before:

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.example.domain

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---------------|-------------------|
| 8 | 63% 90/144 | 64% 16/25 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|-----------------------------|---------------|-------------------|
| ApprovalInformation.java | 0% 0/8 | 0% 0/1 |
| AvailableResources.java | 0% 0/6 | 0% 0/1 |
| ClockUser.java | 64% 7/11 | 67% 2/3 |
| FacultiesResponseModel.java | 67% 4/6 | 100% 1/1 |
| JobRequestRequestModel.java | 87% 13/15 | 0% 0/1 |
| Request.java | 85% 55/65 | 91% 10/11 |
| Resource.java | 58% 11/19 | 75% 3/4 |
| VerificationDto.java | 0% 0/14 | 0% 0/3 |

Report generated by [PIT](#) 1.5.1

Mutations after:

Pit Test Coverage Report

Package Summary

nl.tudelft.sem.template.example.domain

| Number of Classes | Line Coverage | Mutation Coverage |
|-------------------|---------------|-------------------|
| 8 | 68% 98/144 | 72% 18/25 |

Breakdown by Class

| Name | Line Coverage | Mutation Coverage |
|-----------------------------|---------------|-------------------|
| ApprovalInformation.java | 0% 0/8 | 0% 0/1 |
| AvailableResources.java | 0% 0/6 | 0% 0/1 |
| ClockUser.java | 64% 7/11 | 67% 2/3 |
| FacultiesResponseModel.java | 67% 4/6 | 100% 1/1 |
| JobRequestRequestModel.java | 87% 13/15 | 0% 0/1 |
| Request.java | 85% 55/65 | 82% 9/11 |
| Resource.java | 58% 11/19 | 75% 3/4 |
| VerificationDto.java | 57% 8/14 | 100% 3/3 |

Task 2:

1. SchedulingDataProcessingService

Why is this class critical?

The class provides access to the repositories of the Cluster Microservice and is integral to the microservice as it can't work without this data access.

Link to merge request:

https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM11b/-/merge_requests/66

Original method:

```
4 usages  ▾ Arne Potoolski Edit +1
public FacultyTotalResources getAssignedResourcesForGivenFaculty(String facultyId)
    throws IllegalArgumentException {

    if (this.getAllFaculties().contains(facultyId)) {
        return getAssignedResourcesPerFaculty().stream()
            .filter(x -> x.getFaculty_Id().equals(facultyId))
            .collect(Collectors.toList()).get(0);
    } else {
        throw new IllegalArgumentException("Faculty does not exist.");
    }
}
```

New method:

```
public FacultyTotalResources getAssignedResourcesForGivenFaculty(String facultyId)
    throws IllegalArgumentException {

    if (this.getAllFaculties().contains(facultyId)) {
        return getAssignedResourcesPerFaculty().stream()
            .filter(x -> x.getFaculty_Id().equals(facultyId))
            .collect(Collectors.toList()).get(0);
    } else {

        return getAssignedResourcesPerFaculty().get(0);
    }
}
```

Why did we choose this mutant?

This method retrieves the resources that a given faculty has been assigned. The method is called in a controller class so admins can see how many resources a given faculty has, and this method should not result in bad information. Now once a request for the cluster contains an incorrect faculty, it may be accepted and returned as scheduled, while this would not be possible at all. Also if administrators who want to check if a faculty has resources, and sends a faculty who is not registered or misspells it, the method could return a number as if the faculty does have resources, while it may have none or does not exist.

Test that fixed this mutant:

```
@Test
public void getAssignedResourcesForFacultyIdException() {
    this.nodeRepository.save(this.node1);
    this.nodeRepository.save(this.node2);
    this.nodeRepository.save(this.node3);
    Exception exception = assertThrows(IllegalArgumentException.class, ()
        -> this.schedulingDataProcessingService.getAssignedResourcesForGivenFaculty("DWI"));
    assertThat(exception.getMessage()).isEqualTo("Faculty does not exist.");
}
```

2. LeastBusyDayStrategy

The method is part of our scheduling functionality in the Cluster microservice. It's imperative that it works properly, otherwise a core function, mainly being able to schedule a job in the earliest possible time slot, wouldn't work properly and that issue might cascade into other methods of the cluster. In the method, as it is in the code, each resource required for the job is checked to verify there are equal or more of it available. If one of the comparisons is changed to require the exact same amount of the given resource, the method still passes as functional, but there is obviously a mutant. To combat that an additional test case is implemented that checks whether or not the memory required for the job isn't too much for a request past the preferred date.

Method before:

```
public LocalDate scheduleJobFor(List<AvailableResourcesForDate> availableResourcesForDates, Job job) {
    // index of date
    int index = availableResourcesForDates.stream()
        .map(AvailableResourcesForDate::getDate).collect(Collectors.toList())
        .indexOf(job.getPreferredCompletionDate());

    var beforePreferredDate : List<AvailableResourcesForDate> = availableResourcesForDates.subList(0, index + 1);
    beforePreferredDate.sort(Comparator.comparingDouble((AvailableResourcesForDate x) ->
        x.getAvailableCpu() + x.getAvailableGpu() + x.getAvailableMemory()));
    for (AvailableResourcesForDate availableResources : beforePreferredDate) {
        if (job.getRequiredCpu() <= availableResources.getAvailableCpu()
            && job.getRequiredGpu() <= availableResources.getAvailableGpu()
            && job.getRequiredMemory() <= availableResources.getAvailableMemory()) {
            return availableResources.getDate();
        }
    }

    // no match - default to the earliest possible
    for (int j = index; j < availableResourcesForDates.size(); ++j) {
        var availableResources : AvailableResourcesForDate = availableResourcesForDates.get(j);
        if (job.getRequiredCpu() <= availableResources.getAvailableCpu()
            && job.getRequiredGpu() <= availableResources.getAvailableGpu()
            && job.getRequiredMemory() <= availableResources.getAvailableMemory()) {
            return availableResources.getDate();
        }
    }

    // should never be reached
    return availableResourcesForDates.get(availableResourcesForDates.size() - 1).getDate();
}
```

Method after:

```
public LocalDate scheduleJobFor(List<AvailableResourcesForDate> availableResourcesForDates, Job job) {
    // index of date
    int index = availableResourcesForDates.stream()
        .map(AvailableResourcesForDate::getDate).collect(Collectors.toList())
        .indexOf(job.getPreferredCompletionDate());

    var beforePreferredDate : List<AvailableResourcesForDate> = availableResourcesForDates.subList(0, index + 1);
    beforePreferredDate.sort(Comparator.comparingDouble((AvailableResourcesForDate x) ->
        x.getAvailableCpu() + x.getAvailableGpu() + x.getAvailableMemory()));
    for (AvailableResourcesForDate availableResources : beforePreferredDate) {
        if (job.getRequiredCpu() <= availableResources.getAvailableCpu()
            && job.getRequiredGpu() <= availableResources.getAvailableGpu()
            && job.getRequiredMemory() <= availableResources.getAvailableMemory()) {
            return availableResources.getDate();
        }
    }

    // no match - default to the earliest possible
    for (int j = index; j < availableResourcesForDates.size(); ++j) {
        var availableResources : AvailableResourcesForDate = availableResourcesForDates.get(j);
        if (job.getRequiredCpu() <= availableResources.getAvailableCpu()
            && job.getRequiredGpu() <= availableResources.getAvailableGpu()
            && job.getRequiredMemory() == availableResources.getAvailableMemory()) {
            return availableResources.getDate();
        }
    }

    // should never be reached
    return availableResourcesForDates.get(availableResourcesForDates.size() - 1).getDate();
}
```

Additional test that catches the mutation:

```
@Test  
public void tooLittleMemoryPastPreferredTest1() {  
    job.setRequiredMemory(1.0);  
    assertThat(strat.scheduleJobFor(resources, job)).isEqualTo(dateProvider.getTomorrow().plusDays(4));  
}
```

3. JobSchedulingService

Why is this class critical?

This class is a key component of the system in general, and the cluster microservice in particular, because of its role in scheduling jobs (as the name suggests.) If it were not to work, the entire system would cease to operate, as erroneously scheduling jobs essentially makes it impossible to do anything in the system.

Our choice of method to mutate

We chose the method “scheduleJob” to manually mutate. This is because it is the most important in this already key class; if something goes wrong here, the entire system is now broken. We have noticed that our tests failed to take into account the date on which a job is scheduled, as we chose to test that elsewhere. Therefore, we mutated the method by changing the date generated by the strategy to always be tomorrow. This did not fail any tests, even though it would completely break the microservice.

Method before:

```
public LocalDate scheduleJob(Job job) {
    // available resources from tomorrow to day after last scheduled job, inclusive
    // this way, since a check whether this job can be scheduled has been passed, the job can always be fit into
    // the schedule
    var maxDateInSchedule :LocalDate = this.schedulingDataService.findLatestDateWithReservedResources();
    if (job.getPreferredCompletionDate().isAfter(maxDateInSchedule)) {
        maxDateInSchedule = job.getPreferredCompletionDate();
    }
    var availableResourcesPerDay :List<AvailableResourcesForDate> = this.schedulingDataService
        .getAvailableResourcesForGivenFacultyUntilDay(job.getFacultyId(),
            maxDateInSchedule.plusDays( daysToAdd: 1));
    // use strategy to determine a date to schedule the job for
    var dateToScheduleJob :LocalDate = this.strategy.scheduleJobFor(availableResourcesPerDay, job);
    // assign scheduled date to job
    job.setScheduledFor(dateToScheduleJob);
    // save to schedule
    this.schedulingDataService.saveInSchedule(job);
    return dateToScheduleJob;
}
```

Method after:

```
public LocalDate scheduleJob(Job job) {
    // available resources from tomorrow to day after last scheduled job, inclusive
    // this way, since a check whether this job can be scheduled has been passed, the job can always be fit into
    // the schedule
    var maxDateInSchedule :LocalDate = this.schedulingDataService.findLatestDateWithReservedResources();
    if (job.getPreferredCompletionDate().isAfter(maxDateInSchedule)) {
        maxDateInSchedule = job.getPreferredCompletionDate();
    }
    var availableResourcesPerDay :List<AvailableResourcesForDate> = this.schedulingDataService
        .getAvailableResourcesForGivenFacultyUntilDay(job.getFacultyId(),
            maxDateInSchedule.plusDays( daysToAdd: 1));
    // use strategy to determine a date to schedule the job for
    var dateToScheduleJob :LocalDate = this.strategy.scheduleJobFor(availableResourcesPerDay, job);
    // assign scheduled date to job
    job.setScheduledFor(dateProvider.getTomorrow());
    // save to schedule
    this.schedulingDataService.saveInSchedule(job);
    return dateToScheduleJob;
}
```

Additional test that catches the mutation:

```
@Test
public void mutantTest() {
    this.jobSchedulingService.getNodeDataProcessingService().save(this.node2);

    this.jobSchedulingService.scheduleJob(this.job2);
    assertThat(this.jobSchedulingService.getSchedulerDataProcessingService()
        .existsInScheduleByFacultyId(this.job2.getFacultyId())).isTrue();

    this.job1 = new JobBuilder().preferredCompletedBeforeDate(dateProvider.getTomorrow())
        .needingMemoryResources(10.0).needingGpuResources(10.0).needingCpuResources(10.0).withDescription("desc")
        .havingName("jobb").requestedByUserWithNetId(userNetId: "ariel").requestedThroughFaculty(facultyId: "AE")
        .constructJobInstance();

    this.jobSchedulingService.scheduleJob(this.job1);
    assertThat(this.job1.getScheduledFor()).isNotEqualTo(dateProvider.getTomorrow());
}
```

4. FacultyResourcesInformationController

Why is this class critical?

This class is vital to the system as a whole, and to both cluster and request microservices in particular. First, this controller provides support to faculty members for accessing the information regarding faculty resources, which help them to decide whether to accept or reject a merge request. The scheduling and rescheduling functionalities also depend on the faculties information, such as the amount of resources for a given date

Our choice for method to mutate:

We decided to manually mutate the “getAvailableResourcesForGivenFaculty BeforeGivenDate”. This is because we realized this method had no coverage at all, and therefore it contains the greatest number of surviving mutants in the class. For that reason, we also decided to write two different tests instead of one, in that manner killing two mutants instead of only one.

In the following screenshots, the mutations are commented out:

```
@GetMapping(value = @Value"/resources/availableUntil/{date}/{facultyId}")
public ResponseEntity<List<FacultyResourcesResponseModel>> getAvailableResourcesForGivenFacultyBeforeGivenDate(
    @PathVariable("date") String rawDate, @PathVariable("facultyId") String facultyId) {
    // anti-corruption
    /*try {
        LocalDate.parse(rawDate);
    } catch (DateTimeParseException e) {
        return ResponseEntity.badRequest().build();
    }*/
    LocalDate date = LocalDate.parse(rawDate);
    if (date.isBefore(this.dateProvider.getTomorrow())
        || !this.nodeDataProcessingService.existsByFacultyId(facultyId)) {
        return ResponseEntity.badRequest().build();
    }
    var rawResources :List<AvailableResourcesForDate> = this.schedulingDataProcessingService
        .getAvailableResourcesForGivenFacultyUntilDay(facultyId, date);
    return ResponseEntity.ok(FacultyResourcesResponseModel
        .convertForRequestService(rawResources, facultyId));
}
```

```

@GetMapping(value = @Value("/resources/availableUntil/{date}/{facultyId}"))
public ResponseEntity<List<FacultyResourcesResponseModel>> getAvailableResourcesForGivenFacultyBeforeGivenDate(
    @PathVariable("date") String rawDate, @PathVariable("facultyId") String facultyId) {
    // anti-corruption
    try {
        LocalDate.parse(rawDate);
    } catch (DateTimeParseException e) {
        return ResponseEntity.badRequest().build();
    }
    LocalDate date = LocalDate.parse(rawDate);
    /*if (date.isBefore(this.dateProvider.getTomorrow()))
        || !this.nodeDataProcessingService.existsByFacultyId(facultyId)) {
        return ResponseEntity.badRequest().build();
    */
    var rawResources : List<AvailableResourcesForDate> = this.schedulingDataProcessingService
        .getAvailableResourcesForGivenFacultyUntilDay(facultyId, date);
    return ResponseEntity.ok(FacultyResourcesResponseModel
        .convertForRequestService(rawResources, facultyId));
}

```

Additional tests that catches the mutations:

```

@Test
void getAvailableResourcesForGivenFacultyBeforeGivenDateIsInvalidDate() throws Exception {
    node1.setGpuResources(1);
    node1.setMemoryResources(1);
    node1.setCpuResources(5);
    nodeRepository.save(node1);

    this.job.setScheduledFor(this.dateProvider.getTomorrow());
    jobScheduleRepository.save(job);

    ResultActions result = mockMvc.perform(get( urlTemplate: "/resources/availableUntil/2022-12-14/EWI")
        .contentType(MediaType.APPLICATION_JSON)
        .header( name: "Authorization", ...values: "Bearer MockedToken"));

    result.andExpect(status().isBadRequest());
}

@Test
void getAvailableResourcesForGivenFacultyBeforeGivenDateIsInvalidDateWordAsDate() throws Exception {
    node1.setGpuResources(1);
    node1.setMemoryResources(1);
    node1.setCpuResources(5);
    nodeRepository.save(node1);

    this.job.setScheduledFor(this.dateProvider.getTomorrow());
    jobScheduleRepository.save(job);

    ResultActions result = mockMvc.perform(get( urlTemplate: "/resources/availableUntil/twenty/EWI")
        .contentType(MediaType.APPLICATION_JSON)
        .header( name: "Authorization", ...values: "Bearer MockedToken"));

    result.andExpect(status().isBadRequest());
}

```