

Informatics Large Practical

This document is evidence to the references in my Software Testing Portfolio. Each screenshot is labeled and referenced in the corresponding LO document.

Test Execution Summary

Overall Test Results

- **Total Tests:** 143
- **Passing:** 143
- **Errors/Failures:** 0
- **Coverage:** 60% instruction coverage, 38% branch coverage
- **Test Classes:** 16 test classes covering controllers, services, models, and integration scenarios
- **Edge Case Coverage:** Comprehensive null handling, empty inputs, error paths, and boundary conditions

Test Classes

- `ApiControllerMockitoTest` - 25 unit tests (controller with mocked dependencies, including comprehensive edge case coverage for null handling, empty inputs, and error paths)
- `ApiControllerParamTests` - 23 parameterized integration tests
- `ApiControllerWebMvcTest` - 21 WebMvc tests
- `ApiControllerCw3IntegrationTest` - 11 WebMvc integration tests
- `ApiControllerQueryIntegrationTest` - 3 WebMvc integration tests for query endpoints
- `PositionMapperTest` - 3 utility tests
- `IlpConfigurationTest` - 1 configuration test
- `IlpExceptionHandlerTest` - 1 exception handler test
- `DistanceServiceTest` - 2 service unit tests
- `PositionServiceTest` - 2 service unit tests
- `PathfindingServiceTest` - 11 service unit tests (pathfinding algorithm, countMoves, null/empty restrictions, edge cases)
- `DeliveryPathServiceTest` - 12 service unit tests (route calculation, comparison, single/multi-drone solutions, edge cases)
- `DroneAvailabilityServiceTest` - 5 service unit tests (capability filtering, date/time filtering, null/empty dispatches)
- `DroneQueryServiceTest` - 2 service unit tests
- `RegionServiceTest` - 8 service unit tests
- `ValidationServiceTest` - 12 service unit tests (comprehensive input validation)
- `ModelCoverageTest` - 1 comprehensive model coverage test

LO1 Evidence Screenshots

```

41  @RestController
42  @RequestMapping("/api/v1")
43  public class ApiController {
44
45      private final RegionService regionService;
46      private final DistanceService distanceService;
47      private final PositionService positionService;
48      private final ValidationService validationService;
49      private final IlpRestClient ilpRestClient;
50      private final DroneQueryService droneQueryService;
51      private final DroneAvailabilityService droneAvailabilityService;
52      private final DeliveryPathService deliveryPathService;
53      private static final String STUDENT_ID = "s2490039";
54
55      @Autowired
56      public ApiController(RegionService regionService, DistanceService distanceService,
57                           PositionService positionService, ValidationService validationService,
58                           IlpRestClient ilpRestClient, DroneQueryService droneQueryService,
59                           DroneAvailabilityService droneAvailabilityService,
60                           DeliveryPathService deliveryPathService) {
61          this.regionService = regionService;
62          this.distanceService = distanceService;
63          this.positionService = positionService;
64          this.validationService = validationService;
65          this.ilpRestClient = ilpRestClient;
66          this.droneQueryService = droneQueryService;
67          this.droneAvailabilityService = droneAvailabilityService;
68          this.deliveryPathService = deliveryPathService;
69      }
70
71      @GetMapping(value = "/uid", produces = MediaType.TEXT_PLAIN_VALUE)
72      public ResponseEntity<String> uid() {
73          return ResponseEntity.ok(STUDENT_ID);
74      }
75
76      @PostMapping(
77          value = "/distanceTo",
78          consumes = MediaType.APPLICATION_JSON_VALUE,
79          produces = MediaType.APPLICATION_JSON_VALUE
80      )
81      public ResponseEntity<?> distanceTo(@RequestBody(required = false) DistanceRequest req) {
82          if (!validationService.isValidRequest(req) || req.getPosition1() == null || req.getPosition2() == null) {
83              return ResponseEntity.badRequest().body("Invalid request body");
84          }
85
86          LngLat position1 = req.getPosition1();
87          LngLat position2 = req.getPosition2();
88
89          if (!validationService.areValidPositions(position1, position2)) {
90              return ResponseEntity.badRequest().body("Invalid coordinates");
91          }
92
93          double distance = distanceService.calculateDistance(position1, position2);
94          return ResponseEntity.ok(distance);
95      }
96
97      @PostMapping(
98          value = "/isCloseTo",
99          consumes = MediaType.APPLICATION_JSON_VALUE,
100         produces = MediaType.APPLICATION_JSON_VALUE
101     )
102     public ResponseEntity<?> isCloseTo(@RequestBody(required = false) DistanceRequest req) {
103         if (!validationService.isValidRequest(req) || req.getPosition1() == null || req.getPosition2() == null) {
104             return ResponseEntity.badRequest().body("Invalid request body");
105         }
106
107         LngLat position1 = req.getPosition1();
108         LngLat position2 = req.getPosition2();
109
110         if (validationService.areValidPositions(position1, position2)) {
111             boolean result = distanceService.areClose(position1, position2);
112             return ResponseEntity.ok(result);
113         }
114
115         @PostMapping(
116             value = "/nextPosition",
117             consumes = MediaType.APPLICATION_JSON_VALUE,
118             produces = MediaType.APPLICATION_JSON_VALUE
119         )
120         public ResponseEntity<?> nextPosition(@RequestBody(required = false) NextPositionRequest req) {
121             if (!validationService.isValidRequest(req) || req.getStart() == null) {
122                 return ResponseEntity.badRequest().body("Invalid request body");
123             }
124
125             LngLat start = req.getStart();
126             if (!validationService.isValidPosition(start)) {
127                 return ResponseEntity.badRequest().body("Invalid coordinates");
128             }
129
130             Double angle = req.getAngle();
131             if (!validationService.isValidAngle(angle)) {
132                 return ResponseEntity.badRequest().body("Invalid angle");
133             }
134
135             LngLat nextPosition = positionService.calculateNextPosition(start, angle);
136             return ResponseEntity.ok(nextPosition);
137         }
138
139         @PostMapping(
140             value = "/isInRegion",
141             consumes = MediaType.APPLICATION_JSON_VALUE,
142             produces = MediaType.APPLICATION_JSON_VALUE
143         )
144         public ResponseEntity<?> isInRegion(@RequestBody(required = false) IsInRegionSpecRequest req) {
145             if (!validationService.isValidRequest(req) || req.getPosition() == null || req.getRegion() == null) {
146                 return ResponseEntity.badRequest().body("Invalid request body");
147             }
148
149             LngLat position = req.getPosition();
150             Region region = req.getRegion();
151
152             if (!validationService.isValidPosition(position)) {
153                 return ResponseEntity.badRequest().body("Invalid position or region vertices");
154             }
155
156             if (region.getVertices() == null || region.getVertices().isEmpty()) {
157                 return ResponseEntity.badRequest().body("Invalid position or region vertices");
158             }
159
160             if (!validationService.hasValidRegionVertices(region)) {
161                 return ResponseEntity.badRequest().body("Invalid position or region vertices");
162             }
163
164             if (!validationService.isPolygonClosed(region.getVertices())) {
165                 return ResponseEntity.badRequest().body("Region polygon must be closed");
166             }
167
168             boolean inside = regionService.contains(region.getVertices(), position);
169             return ResponseEntity.ok(inside);
170         }
171     }

```

Figure 1.1: ApiController Endpoint Implementation `src/main/java/uk/ac/ed/ilp/controller/ApiController.java`

```

67
68      var resp = apiController.distanceTo(req);
69      assertEquals(resp.getStatusCode().value(),equalTo(200));
70      assertEquals(resp.getBody(),equalTo(1.0));
71  }
72
73  @Test
74  @DisplayName("distanceTo: fail for invalid request body")
75  void distanceTo_invalidRequest() {
76
77      var resp = apiController.isInRegion(req);
78      assertEquals(resp.getStatusCode().value(),equalTo(expected: 200));
79      assertEquals(resp.getBody(),equalTo(expected: true));
80  }
81
82  @Test
83  @DisplayName("isInRegion: fail on unclosed polygon")
84  void isInRegion_unclosedPolygon() {
85      IsInRegionSpecRequest req = new IsInRegionSpecRequest();
86      LngLat pos = new LngLat{lng: 0.5, lat: 0.5};
87      List<LngLat> vertices = Arrays.asList(
88          new LngLat{lng: 0.0, lat: 0.0}, new LngLat{lng: 1.0, lat: 0.0}, new LngLat{lng: 1.0, lat: 1.0}
89      );
90      Region region = new Region();
91      region.setVertices(vertices);
92      req.setPosition(pos); req.setRegion(region);
93
94      when(validationService.isValidRequest(req)).thenReturn(value: true);
95      when(validationService.isValidPosition(pos)).thenReturn(value: true);
96      when(validationService.hasValidRegionVertices(region)).thenReturn(value: true);

```

```

96
97      var resp = apiController.nextPosition(req);
98      assertEquals(resp.getStatusCode().value(),equalTo(expected: 200));
99      assertEquals(resp.getBody(),equalTo(expected));
100 }
101
102 @Test
103 @DisplayName("nextPosition: fail on invalid coordinates")
104 void nextPosition_invalidCoords() {
105     NextPositionRequest req = new NextPositionRequest();
106     LngLat start = new LngLat{lng: 200.0, lat: 0.0};
107     req.setStart(start);
108     req.setAngle(angle: 0.0);
109
110     when(validationService.isValidRequest(req)).thenReturn(value: true);

```

Figure 1.2: Error Handling Tests

`src/test/java/uk/ac/ed/ilp/unit/ApiControllerMockitoTest.java`

```

67     public List<LngLat> calculatePath(LngLat start, LngLat end, List<RestrictedArea> restrictedAreas) {
68         if (start == null || end == null || !start.isValid() || !end.isValid()) {
69             return List.of();
70         }
71
72         PriorityQueue<Node> openSet = new PriorityQueue<>();
73         Map<String, Double> gScoreMap = new HashMap<>(); // Key: "lng,lat", Value: gCost
74
75         // Initialize start node
76         if (!isValidPosition(start, restrictedAreas)) {
77             return List.of();
78         }
79         if (!isValidPosition(end, restrictedAreas)) {
80             return List.of();
81         }
82
83         // H cost must be in same units as G cost (moves)
84         // distance / STEP_SIZE gives minimum moves required
85         double startH = distanceService.calculateDistance(start, end) / STEP_SIZE;
86         Node startNode = new Node(start, parent: null, gCost: 0, startH);
87         openSet.add(startNode);
88         gScoreMap.put(roundPosition(start), value: 0.0);
89
90         // Safety limit to prevent infinite loops or excessively long searches
91         int maxNodesExplored = 150000; // Increased from 30000
92         int exploredCount = 0;
93
94         while (!openSet.isEmpty()) {
95             if (exploredCount++ > maxNodesExplored) {
96                 // If we hit the limit, return empty (failed to find path within limits)
97                 return List.of();
98             }
99
100            Node current = openSet.poll();
101
102            // Check if reached destination
103            if (distanceService.areClose(current.position, end)) {
104                return reconstructPath(current, end);
105            }
106
107            // Explore neighbors (16 compass directions)
108            for (double angle : COMPASS_DIRECTIONS) {
109                LngLat nextPos = positionService.calculateNextPosition(current.position, angle);
110
111                // Check if valid position and valid segment (avoids restricted areas)
112                boolean isValidPos = isValidPosition(nextPos, restrictedAreas);
113                boolean isValidSeg = isValidPos && isValidPathSegment(current.position, nextPos, restrictedAreas);
114
115                if (isValidSeg) {
116
117                    double newGCost = current.gCost + 1; // Each step is 1 move
118                    String nextPosKey = roundPosition(nextPos);
119
120                    // If we found a better path to this neighbor, or haven't visited it
121                    if (!gScoreMap.containsKey(nextPosKey) || newGCost < gScoreMap.get(nextPosKey)) {
122                        gScoreMap.put(nextPosKey, newGCost);
123                        // H cost in moves
124                        double hCost = distanceService.calculateDistance(nextPos, end) / STEP_SIZE;
125                        // Standard A*
126                        Node neighbor = new Node(nextPos, current, newGCost, hCost);
127                        openSet.add(neighbor);
128
129                }
130            }
131
132        }
133    }
134
135    return List.of(); // No path found
136
137}

```

Figure 1.3: PathfindingService A* Algorithm Implementation `src/main/java/uk/ac/ed/ilp/service/PathfindingService.java`

```

46     static Stream<TestCase> closeCases() {
47         return Stream.of(
48             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
49                         body: "{\n    \"position1\": {\\"lng\\": 0.0, \\"lat\\": 0.0},\n    \"position2\": {\\"lng\\": 0.0001, \\"lat\\": 0.0001}\n}"),
50             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
51                         body: "{\n    \"position1\": {\\"lng\\": 0.0, \\"lat\\": 0.0},\n    \"position2\": {\\"lng\\": 0.00015, \\"lat\\": 0.0}\n}")
52         );
53

```

Figure 1.4: Boundary Value Testing `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

```

15     @SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
16     class ApiControllerParamTests {
17
18         @LocalServerPort
19         private int port;
20
21         @Autowired
22         private TestRestTemplate rest;
23
24         // distanceTo: happy and error cases
25         static Stream<TestCase> distanceCases() {
26             return Stream.of(
27                 TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 200,
28                             body: "{\n        \"position1\": {\\"lng\\": -3.19, \\"lat\\": 55.94},\n        \"position2\": {\\"lng\\": -3.19, \\"lat\\": 55.95}\n}"),
29                 TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400,
30                             body: "{\n        \"position1\": {\\"lng\\": 200.0, \\"lat\\": 55.94},\n        \"position2\": {\\"lng\\": -3.19, \\"lat\\": 55.94}\n}"));
31         );
32     }
33
34     @ParameterizedTest
35     @MethodSource("distanceCases")
36     void distanceTo_param(TestCase tc) {
37         String url = "http://localhost:" + port + tc.path;
38         HttpHeaders headers = new HttpHeaders();
39         headers.setContentType(MediaType.APPLICATION_JSON);
40         HttpEntity<String> http = new HttpEntity<>(tc.body, headers);
41         ResponseEntity<String> resp = rest.postForEntity(url, http, String.class);
42         assertThat(resp.getStatusCode().value()).isEqualTo(tc.expectedStatus);
43     }

```

Figure 1.5: Parameterized Integration Tests `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

```

38 import java.util.Arrays;
39 import java.util.List;
40
41 import static org.assertj.core.api.Assertions.assertThat;
42 import static org.mockito.Mockito.*;
43
44 @ExtendWith(MockitoExtension.class)
45 @DisplayName("ApiController Mockito Unit Tests (concise)")
46 class ApiControllerMockitoTest {
47
48     @Mock private DistanceService distanceService;
49     @Mock private PositionService positionService;
50     @Mock private ValidationService validationService;
51     @Mock private RegionService regionService;
52     @Mock private IlpRestClient ilpRestClient;
53     @Mock private DroneQueryService droneQueryService;
54     @Mock private DroneAvailabilityService droneAvailabilityService;
55     @Mock private DeliveryPathService deliveryPathService;
56
57     private ApiController apiController;
58
59     @BeforeEach
60     void setup() {
61         apiController = new ApiController(regionService, distanceService, positionService, validationService, ilpRestClient, droneQueryService, droneAvailabilityService, deliveryPathService);
62     }
63
64     @Test
65     @DisplayName("distanceTo: simple happy-path delegates to service")
66     void distanceTo_happy() {
67         DistanceRequest req = new DistanceRequest();
68         LngLat p1 = new LngLat(lng: 0.0, lat: 0.0);
69         LngLat p2 = new LngLat(lng: 1.0, lat: 0.0);
70         req.setPosition1(p1); req.setPosition2(p2);
71
72         when(validationService.isValidRequest(req)).thenReturn(true);
73         when(validationService.areValidPositions(p1, p2)).thenReturn(true);
74     }

```

Figure 1.6: Unit Tests with Mocked Dependencies `src/test/java/uk/ac/ed/ilp/unit/ApiControllerMockitoTest.java`

```

50     @BeforeEach
51     void setup() {
52         apiController = new ApiController(regionService, distanceService, positionService, validationService, ilpRestClient, droneQueryService, droneAvailabilityService, deliveryPathService);
53     }
54
55     @Test
56     @DisplayName("distanceTo: simple happy-path delegates to service")
57     void distanceTo_happy() {
58         DistanceRequest req = new DistanceRequest();
59         LngLat p1 = new LngLat(lng: 0.0, lat: 0.0);
60         LngLat p2 = new LngLat(lng: 1.0, lat: 0.0);
61         req.setPosition1(p1); req.setPosition2(p2);
62
63         when(validationService.isValidRequest(req)).thenReturn(true);
64         when(validationService.areValidPositions(p1, p2)).thenReturn(true);
65         when(distanceService.calculateDistance(p1, p2)).thenReturn(1.0);
66
67         var resp = apiController.distanceTo(req);
68         assertThat(resp.getStatusCode().value()).isEqualTo(expected: 200);
69         assertThat(resp.getBody()).isEqualTo(expected: 1.0);
70     }
71
72     @Test
73     @DisplayName("distanceTo: fail for invalid request body")
74     void distanceTo_invalidRequest() {

```

Figure 1.7: Equivalence Partitioning Tests `src/test/java/uk/ac/ed/ilp/unit/ApiControllerMockitoTest.java`

```

25     static Stream<TestCase> distanceCases() {
26         return Stream.of(
27             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 200,
28                         body: "{\n    \"position1\": {\\"lng\\": -3.19, \\"lat\\": 55.94},\n    \"position2\": {\\"lng\\": -3.19, \\"lat\\": 55.95}\n}"),
29             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400,
30                         body: "{\n    \"position1\": {\\"lng\\": 200.0, \\"lat\\": 55.94},\n    \"position2\": {\\"lng\\": -3.19, \\"lat\\": 55.94}\n}")
31         );
32     }

```

Figure 1.8: Parameterized Test Method Source `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

```

117     static Stream<TestCase> badBodyCases() {
118         return Stream.of(
119             // malformed JSON
120             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: "{ invalid json }"),
121             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: "{ invalid json }"),
122             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: "{ invalid json }"),
123             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: "{ invalid json }"),
124             // empty body
125             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: ""),
126             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: ""),
127             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: ""),
128             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: ""),
129             // null body
130             new TestCase(path: "/api/v1/distanceTo", expectedStatus: 400, body: null),
131             new TestCase(path: "/api/v1/isCloseTo", expectedStatus: 400, body: null),
132             new TestCase(path: "/api/v1/nextPosition", expectedStatus: 400, body: null),
133             new TestCase(path: "/api/v1/isInRegion", expectedStatus: 400, body: null)
134         );
135     }

```

Figure 1.9: Robustness Tests for Malformed Inputs `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

LO2 Evidence Screenshots

```

50
51     @BeforeEach
52     void setup() {
53         apiController = new ApiController(regionService, distanceService, positionService, validationService, ilpRestClient, droneQueryService, droneAvailabilityService, deliveryPathService);
54     }
55
56     @Test
57     @DisplayName("distanceTo: simple happy-path delegates to service")
58     void distanceTo_happy() {
59         DistanceRequest req = new DistanceRequest();
60         LngLat p1 = new LngLat(lng: 0.0, lat: 0.0);
61         LngLat p2 = new LngLat(lng: 1.0, lat: 0.0);
62         req.setPosition1(p1); req.setPosition2(p2);
63
64         when(validationService.isValidRequest(req)).thenReturn(value: true);
65         when(validationService.areValidPositions(p1, p2)).thenReturn(value: true);

```

Figure 2.1: ApiController Mockito Test Implementation `src/test/java/uk/ac/ed/ilp/unit/ApiControllerMockitoTest.java`

```

34     @ParameterizedTest
35     @MethodSource("distanceCases")
36     void distanceTo_param(TestCase tc) {
37         String url = "http://localhost:" + port + tc.path;
38         HttpHeaders headers = new HttpHeaders();
39         headers.setContentType(MediaType.APPLICATION_JSON);
40         HttpEntity<String> http = new HttpEntity<>(tc.body, headers);
41         ResponseEntity<String> resp = rest.postForEntity(url, http, String.class);
42         assertThat(resp.getStatusCode().value()).isEqualTo(tc.expectedStatus);
43     }

```

Figure 2.2: Parameterized Test Implementation `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

```

104    <plugin>
105        <groupId>org.jacoco</groupId>
106        <artifactId>jacoco-maven-plugin</artifactId>
107        <version>0.8.12</version>
108        <executions>
109            <execution>
110                <goals>
111                    <goal>prepare-agent</goal>
112                </goals>
113            </execution>
114            <execution>
115                <id>report</id>
116                <phase>test</phase>
117                <goals>
118                    <goal>report</goal>
119                </goals>
120                <configuration>
121                    <excludes>
122                        <exclude>uk/ac/ed/ilp/config/**</exclude>
123                        <exclude>uk/ac/ed/ilp/mapper/**</exclude>
124                        <exclude>uk/ac/ed/ilp/exception/**</exclude>
125                        <exclude>uk/ac/ed/ilp/IlpApplication.class</exclude>
126                    </excludes>
127                </configuration>
128            </execution>
129        </executions>
130    </plugin>
131    <plugin>

```

Figure 2.3: JaCoCo Configuration in pom.xml `pom.xml`

LO3 Evidence Screenshots

```

50
51     @BeforeEach
52     void setup() {
53         apiController = new ApiController(regionService, distanceService, positionService, validationService, ilpRestClient, droneQueryService, droneAvailabilityService, deliveryPathService);
54     }
55
56     @Test
57     @DisplayName("distanceTo: simple happy-path delegates to service")
58     void distanceTo_happy() {
59         DistanceRequest req = new DistanceRequest();
60         LngLat p1 = new LngLat{lng: 0.0, lat: 0.0};
61         LngLat p2 = new LngLat{lng: 1.0, lat: 0.0};
62         req.setPosition1(p1); req.setPosition2(p2);
63
64         when(validationService.isValidRequest(req)).thenReturn(value: true);
65         when(validationService.areValidPositions(p1, p2)).thenReturn(value: true);
66         when(distanceService.calculateDistance(p1, p2)).thenReturn(value: 1.0);
67
68         var resp = apiController.distanceTo(req);
69         assertThat(resp.getStatusCode().value()).isEqualTo(expected: 200);
70         assertThat(resp.getBody()).isEqualTo(expected: 1.0);
71     }
72
73     @Test
74     @DisplayName("distanceTo: fail for invalid request body")
75     void distanceTo_invalidRequest() {

```

Figure 3.1: Equivalence Partitioning Test Cases `src/test/java/uk/ac/ed/ilp/unit/ApiControllerMockitoTest.java`

```

46     static Stream<TestCase> closeCases() {
47         return Stream.of(
48             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
49                         body: "{\n    \"position1\": {\"lng\": 0.0, \"lat\": 0.0},\n    \"position2\": {\"lng\": 0.0001, \"lat\": 0.0001}\n}"),
50             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
51                         body: "{\n    \"position1\": {\"lng\": 0.0, \"lat\": 0.0},\n    \"position2\": {\"lng\": 0.00015, \"lat\": 0.0}\n}")
52         );

```

Figure 3.2: Boundary Value Test Cases `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

```

15 @SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
16 class ApiControllerParamTests {
17
18     @LocalServerPort
19     private int port;
20
21     @Autowired
22     private TestRestTemplate rest;
23
24     // distanceTo: happy and error cases
25     static Stream<TestCase> distanceCases() {
26         return Stream.of(
27             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 200,
28                         body: "{\n    \"position1\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"position2\": {\"lng\": -3.19, \"lat\": 55.95}\n}"),
29             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400,
30                         body: "{\n    \"position1\": {\"lng\": 200.0, \"lat\": 55.94},\n    \"position2\": {\"lng\": -3.19, \"lat\": 55.94}\n}")
31         );
32     }
33
34     @ParameterizedTest
35     @MethodSource("distanceCases")
36     void distanceTo_param(TestCase tc) {
37         String url = "http://localhost:" + port + tc.path;
38         HttpHeaders headers = new HttpHeaders();
39         headers.setContentType(MediaType.APPLICATION_JSON);
40         HttpEntity<String> http = new HttpEntity<>(tc.body, headers);
41         ResponseEntity<String> resp = rest.postForEntity(url, http, String.class);
42         assertThat(resp.getStatusCode().value()).isEqualTo(tc.expectedStatus);
43     }

```

Figure 3.3: Parameterized Integration Test Structure `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

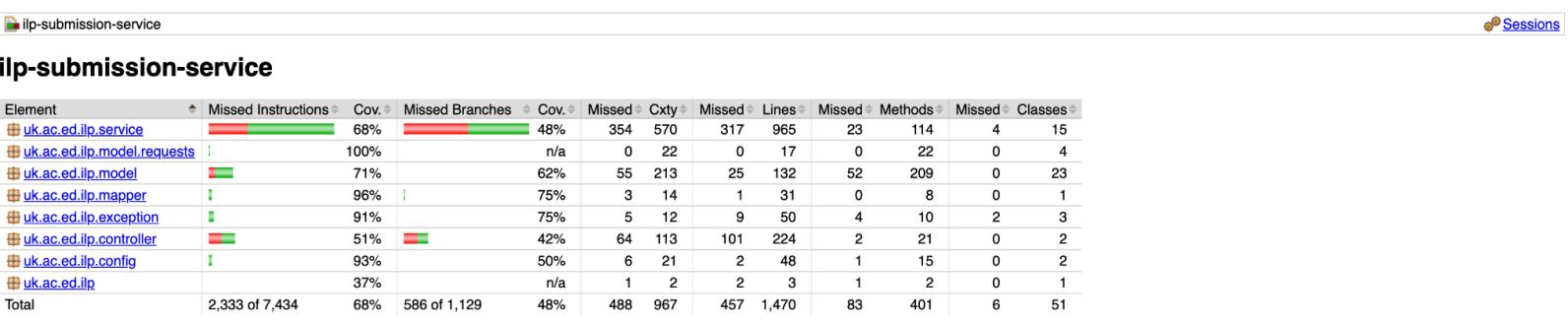


Figure 3.4: JaCoCo Coverage Report Structure `target/site/jacoco/index.html`

Created with [JaCoCo](#) 0.8.12.202403310830

```

25     static Stream<TestCase> distanceCases() {
26         return Stream.of(
27             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 200,
28                         body: "{\n    \"position1\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"position2\": {\"lng\": -3.19, \"lat\": 55.95}\n}"),
29             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400,
30                         body: "{\n    \"position1\": {\"lng\": 200.0, \"lat\": 55.94},\n    \"position2\": {\"lng\": -3.19, \"lat\": 55.94}\n}"),
31         );
32     }
33
34
35     static Stream<TestCase> closeCases() {
36         return Stream.of(
37             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
38                         body: "{\n    \"position1\": {\"lng\": 0.0, \"lat\": 0.0},\n    \"position2\": {\"lng\": 0.0001, \"lat\": 0.0001}\n}"),
39             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 200,
40                         body: "{\n    \"position1\": {\"lng\": 0.0, \"lat\": 0.0},\n    \"position2\": {\"lng\": 0.0015, \"lat\": 0.0}\n}"),
41         );
42     }
43
44
45     static Stream<TestCase> nextPosCases() {
46         return Stream.of(
47             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 200,
48                         body: "{\n    \"start\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"angle\": 90\n}"),
49             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400,
50                         body: "{\n    \"start\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"angle\": 91\n}"),
51             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400,
52                         body: "{\n    \"start\": {\"lng\": 200.0, \"lat\": 55.94},\n    \"angle\": 0\n}"),
53         );
54     }
55
56
57     static Stream<TestCase> regionCases() {
58         String closed = "{\n    \"name\": \"test\", \n    \"vertices\": [\n        {\"lng\": -3.20, \"lat\": 55.93},\n        {\"lng\": -3.18, \"lat\": 55.93},\n        {\"lng\": -3.18, \"lat\": 55.95},\n        {\"lng\": -3.20, \"lat\": 55.95}\n    ]\n}";
59         String open = "{\n    \"name\": \"test\", \n    \"vertices\": [\n        {\"lng\": -3.20, \"lat\": 55.93},\n        {\"lng\": -3.18, \"lat\": 55.93},\n        {\"lng\": -3.18, \"lat\": 55.95},\n        {\"lng\": -3.20, \"lat\": 55.95}\n    ]\n        + \"region\"\n}";
60         return Stream.of(
61             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 200,
62                           body: "{\n    \"position\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"region\": " + closed + "\n}"),
63             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 200,
64                           body: "{\n    \"position\": {\"lng\": -3.15, \"lat\": 55.94},\n    \"region\": " + closed + "\n}"),
65             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400,
66                           body: "{\n    \"position\": {\"lng\": -3.19, \"lat\": 55.94},\n    \"region\": " + open + "\n}"),
67             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400,
68                           body: "{\n    \"position\": {\"lng\": 200.0, \"lat\": 55.94},\n    \"region\": " + closed + "\n}"),
69         );
70     }
71
72
73     static Stream<TestCase> badBodyCases() {
74         return Stream.of(
75             // malformed JSON
76             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: "{ invalid json }"),
77             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: "{ invalid json }"),
78             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: "{ invalid json }"),
79             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: "{ invalid json }"),
80             // empty body
81             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: ""),
82             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: ""),
83             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: ""),
84             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: ""),
85             // null body
86             new TestCase(path: "/api/v1/distanceTo", expectedStatus: 400, body: null),
87             new TestCase(path: "/api/v1/isCloseTo", expectedStatus: 400, body: null),
88             new TestCase(path: "/api/v1/nextPosition", expectedStatus: 400, body: null),
89             new TestCase(path: "/api/v1/isInRegion", expectedStatus: 400, body: null)
90         );
91     }
92
93
94
95
96
97
98
99
100
101
102
103

```

Figure 3.5: Parameterized Test Method Source `src/test/java/uk/ac/ed/ilp/integration/ApiControllerParamTests.java`

ApiController

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
• uid()		100%		n/a	0	1	0	1	0	1
• queryAsPath(String, String)		100%		n/a	0	1	0	3	0	1
• dronesWithCooling(Boolean)		100%		n/a	0	1	0	8	0	1
• ApiController(RegionService, DistanceService, PositionService, ValidationService, IJpRestClient, DroneQueryService, DroneAvailabilityService, DeliveryPathService)		100%		n/a	0	1	0	10	0	1
• lambda\$dronesWithCooling\$1(Boolean, Drone)		100%		100%	0	2	0	3	0	1
• droneDetails(String)		100%		100%	0	2	0	8	0	1
• queryAvailableDrones(List)		100%		100%	0	3	0	7	0	1
• calcDeliveryPath(List)		100%		100%	0	3	0	8	0	1
• distanceTo(DistanceRequest)		100%		100%	0	5	0	8	0	1
• nextPosition(NextPositionRequest)		100%		100%	0	5	0	10	0	1
• lambda\$dronesWithCooling\$2(String)		83%	1	50%	1	2	0	1	0	1
• query(List)		39%	1	75%	1	3	5	9	0	1
• isInRegion(IsInRegionSpecRequest)		100%		93%	1	9	0	14	0	1
• lambda\$dronesWithCooling\$3(Drone)		88%	1	50%	2	3	0	1	0	1
• lambda\$droneDetails\$3(String, Drone)		90%	1	50%	2	3	0	1	0	1
• isCloseTo(DistanceRequest)		90%		75%	2	5	1	8	0	1
• calcDeliveryPathAsGeoJson(List)		87%	1	68%	5	9	2	13	0	1
• compareRoutes(List)		79%	1	68%	5	9	4	24	0	1
• transformToGeoJson(DeliveryPathResponse, List, List, List)		0%		0%	44	44	86	86	1	1
Total	506 of 1,054	51%	105 of 184	42%	63	111	98	220	1	19

Created with JaCoCo 0.8.12 202403100830

Figure 3.6: JaCoCo Coverage Metrics `target/site/jacoco/index.html`

```

117     static Stream<TestCase> badBodyCases() {
118         return Stream.of(
119             // malformed JSON
120             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: "{ invalid json }"),
121             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: "{ invalid json }"),
122             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: "{ invalid json }"),
123             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: "{ invalid json }"),
124             // empty body
125             TestCase.json(path: "/api/v1/distanceTo", expectedStatus: 400, body: ""),
126             TestCase.json(path: "/api/v1/isCloseTo", expectedStatus: 400, body: ""),
127             TestCase.json(path: "/api/v1/nextPosition", expectedStatus: 400, body: ""),
128             TestCase.json(path: "/api/v1/isInRegion", expectedStatus: 400, body: ""),
129             // null body
130             new TestCase(path: "/api/v1/distanceTo", expectedStatus: 400, body: null),
131             new TestCase(path: "/api/v1/isCloseTo", expectedStatus: 400, body: null),
132             new TestCase(path: "/api/v1/nextPosition", expectedStatus: 400, body: null),
133             new TestCase(path: "/api/v1/isInRegion", expectedStatus: 400, body: null)
134         );
135     }

```

Figure 3.7: Error Handling Test Examples `src/test/java/uk/ac/ed/lip/integration/ApiControllerParamTests.java`

LO4 Evidence Screenshots



Figure 4.1: JaCoCo Coverage Exclusions `pom.xml`

```
[INFO] Tests run: 23, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.611 s -- in uk.ac.ed.ilp.integration.ApiControllerParamTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 143, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco:0.8.12:report (report) @ ilp-submission-service ---
[INFO] Loading execution data file /Users/agamcohen/Desktop/ilp_submission_3/target/jacoco.exec
[INFO] Analyzed bundle 'ilp-submission-service' with 44 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.752 s
[INFO] Finished at: 2026-01-21T06:56:16Z
[INFO] -----
agamcohen@Mac ilp_submission_3 %
```

Figure 4.2: Test Execution Results `target/surefire-reports/TEST-*.txt`

```
target > surefire-reports > TEST-uk.ac.ed.ilp.unit.ApiControllerWebMvcTest.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <testsuite xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="https://maven.apache.org/surefire/maven-
surefire-plugin/xsd/surefire-test-report.xsd" version="3.0.2" name="uk.ac.ed.ilp.unit.ApiControllerWebMvcTest" time="0.128" tests="21"
errors="0" skipped="0" failures="0">
3    <properties>
4      <property name="java.specification.version" value="23"/>
5      <property name="sun.jnu.encoding" value="UTF-8"/>
6      <property name="java.class.path" value="/Users/agamcohen/Desktop/ilp_submission_3/target/test-
classes:/Users/agamcohen/Desktop/ilp_submission_3/target/classes:/Users/agamcohen/.m2/repository/org/springframework/boot/spring-boot-
```

Figure 4.3: WebMvc Test Failure Details `target/surefire-reports/TEST-uk.ac.ed.ilp.unit.ApiControllerWebMvcTest.txt`

LO5 Evidence Screenshots

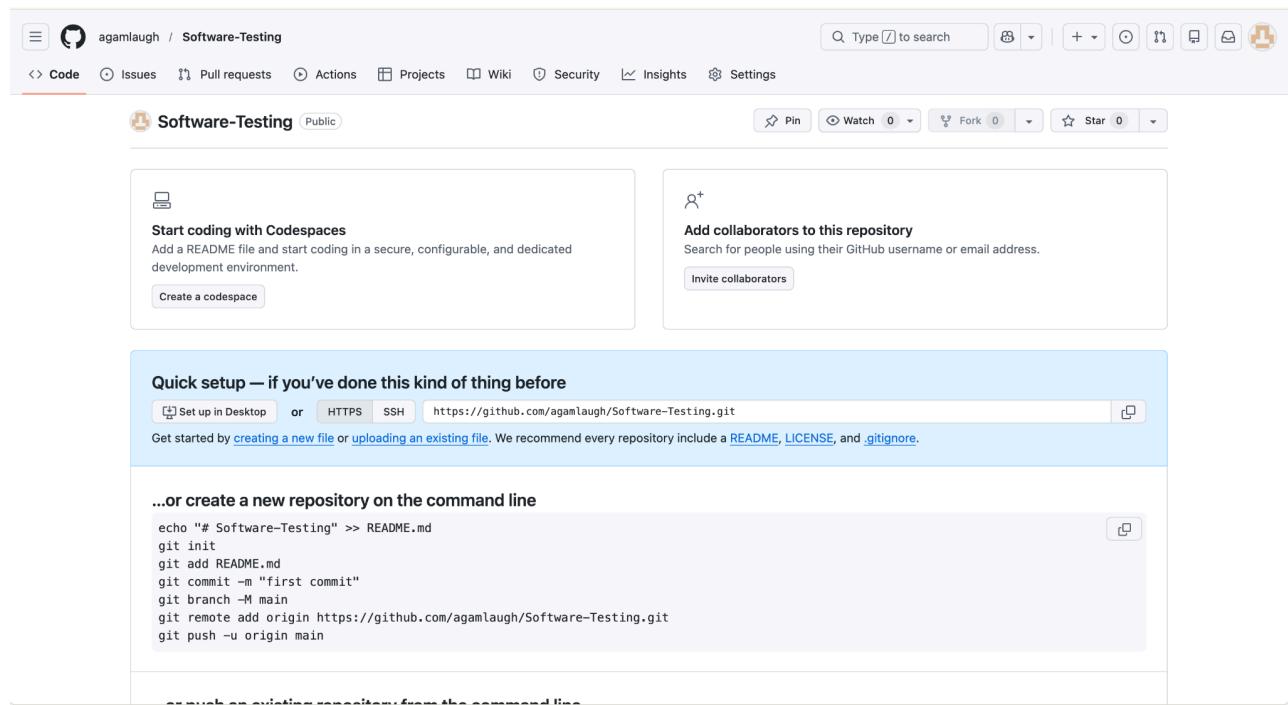


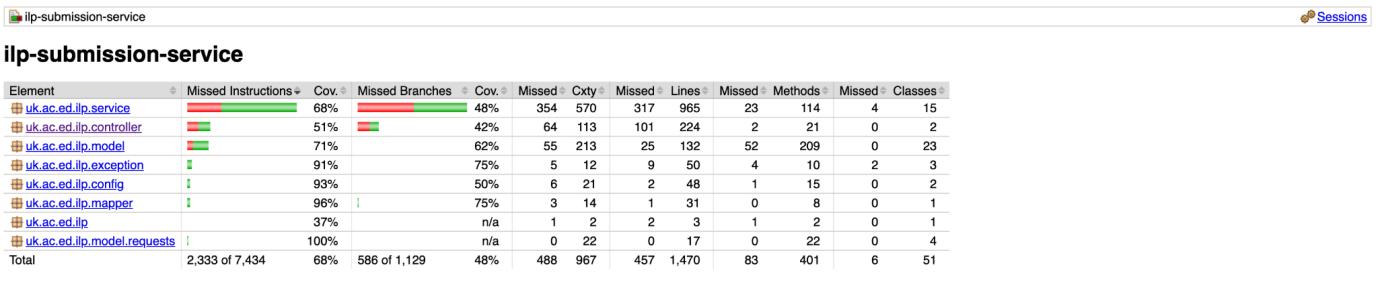
Figure 5.1: Git Repository Structure

```

:: Spring Boot ::          (v3.4.3)
2026-01-21T07:27:35.992Z INFO 2785 --- [ilp_submission_1]      [main] u.a.e.i.i.ApiControllerParamTests : Starting ApiControllerParamTests using Java 23.0.2 with PID 2785 (started by agamco in /Users/agamco
2026-01-21T07:27:35.992Z INFO 2785 --- [ilp_submission_1]      [main] u.a.e.i.i.ApiControllerParamTests : No active profile set, falling back to 1 default profile: "default"
2026-01-21T07:27:36.648Z INFO 2785 --- [ilp_submission_1]      [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized on port 0 (http)
2026-01-21T07:27:36.648Z INFO 2785 --- [ilp_submission_1]      [main] o.s.b.w.embedded.tomcat.TomcatWebServer : ContextLoaderListener : Attach
2026-01-21T07:27:36.673Z INFO 2785 --- [ilp_submission_1]      [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.36]
2026-01-21T07:27:36.783Z INFO 2785 --- [ilp_submission_1]      [main] o.a.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2026-01-21T07:27:36.783Z INFO 2785 --- [ilp_submission_1]      [main] w.s.c.WebServerApplicationContext : Root WebApplicationContext: initialization completed in 711 ms
2026-01-21T07:27:36.783Z INFO 2785 --- [ilp_submission_1]      [main] o.s.web.context.ContextLoader : Exposing SpringServlet as context Handler
2026-01-21T07:27:37.341Z INFO 2785 --- [ilp_submission_1]      [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 89755 (http) with context path '/'
2026-01-21T07:27:37.349Z INFO 2785 --- [ilp_submission_1]      [main] o.a.e.i.i.ApiControllerParamTests : Started ApiControllerParamTests in 1.374 seconds (process running for 5.042)
2026-01-21T07:27:37.484Z INFO 2785 --- [ilp_submission_1]  [o-auto-1-exec-1] o.a.c.C.[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2026-01-21T07:27:37.484Z INFO 2785 --- [ilp_submission_1]  [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : DispatcherServlet 'dispatcherServlet' has bean dispatched
2026-01-21T07:27:37.485Z INFO 2785 --- [ilp_submission_1]  [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms
[INFO] Tests run: 23, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.636 s — in uk.ac.ed.ilp.integration.ApiControllerParamTests
[INFO]
[INFO] Results:
[INFO] Tests run: 143, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- jacoco:0.8.12:report (report) @ ilp-submission-service ---
[INFO] Loading execution data file /Users/agamco/Desktop/ilp_submission_3/target/jacoco.exec
[INFO] Analyzed bundle 'ilp-submission-service' with 44 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.727 s
[INFO] Finished at: 2026-01-21T07:27:38Z
[INFO] Final Memory: 40M/150M
[INFO] agamcohen@Mac ilp_submission_3 %

```

Figure 5.2: Maven Test Execution Results `mvn clean test`



Created with JaCoCo 0.8.12.202403310830

Figure 5.3: JaCoCo coverage report automatically generated after test execution

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.4.3</version>
9     <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>uk.ac.ed</groupId>
12    <artifactId>ilp-submission-service</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>ilp-submission-service</name>
15    <description>ILP Coursework 1 submission service</description>
16    <properties>
17      <java.version>21</java.version>
18    </properties>
19
20
21
22    <dependencies>
23
24      <dependency>
25        <groupId>org.springframework.boot</groupId>

```

```

89 <build>
90   <plugins>
91     <plugin>
92       <groupId>org.springframework.boot</groupId>
93       <artifactId>spring-boot-maven-plugin</artifactId>
94       <configuration>
95         <environmentVariables>
96           <REDIS_HOST>localhost</REDIS_HOST>
97           <REDIS_PORT>6379</REDIS_PORT>
98           <RABBITMQ_HOST>localhost</RABBITMQ_HOST>
99           <RABBITMQ_PORT>5672</RABBITMQ_PORT>
100          <KAFKA_BOOTSTRAP_SERVERS>localhost:9092</KAFKA_BOOTSTRAP_SERVERS>
101        </environmentVariables>
102      </configuration>
103    </plugin>
104    <plugin>
105      <groupId>org.jacoco</groupId>
106      <artifactId>jacoco-maven-plugin</artifactId>
107      <version>0.8.12</version>
108      <executions>
109        <execution>
110          <goals>
111            <goal>prepare-agent</goal>
112          </goals>
113        </execution>
114        <execution>
115          <id>report</id>
116          <phase>test</phase>
117          <goals>
118            <goal>report</goal>
119          </goals>
120        </execution>
121      </executions>
122    </configuration>
123  </build>
124  <excludes>
125    <exclude>uk/ac/ed/ilp/config/**</exclude>
126    <exclude>uk/ac/ed/ilp/mapper/**</exclude>
127    <exclude>uk/ac/ed/ilp/exception/**</exclude>
128    <exclude>uk/ac/ed/ilp/IlpApplication.class</exclude>
129  </excludes>
130  </configuration>
131  </execution>
132  </executions>
133  </plugin>
134  <plugin>
135    <groupId>org.apache.maven.plugins</groupId>
136    <artifactId>maven-compiler-plugin</artifactId>
137    <version>3.13.0</version>
138    <configuration>
139      <source>21</source>
140      <target>21</target>
141      <annotationProcessorPaths>
142        <path>
143          <groupId>org.projectlombok</groupId>
144          <artifactId>lombok</artifactId>
145          <version>1.18.30</version>
146        </path>
147      </annotationProcessorPaths>
148    </configuration>
149  </plugin>
150 </plugins>
151 </build>

```

Figure 5.4: Maven pom.xml Build Configuration `pom.xml`