

## Junit Test

The screenshot shows an IDE window with the following components:

- Run Console (Top):** Displays "Finished after 7.719 seconds", "Runs: 5/5", "Errors: 0", and "Failures: 0".
- Package Explorer (Left):** Shows the project structure with a folder "EmployeeControllerTest (Runner: JUnit 5 (JUnit 5))". It lists five test methods: `getByIdTest` (0.002 s), `updateTest` (0.004 s), `getByIdTest` (0.000 s), `createTest` (0.007 s), and `deleteTest` (0.004 s).
- Code Editor (Right):** Contains the source code for `EmployeeControllerTest.java`. The code includes imports for `JUnit`, `MockMvc`, `HttpHeaders`, `Employee`, `Mapper`, and `HttpEntity`. It defines three test methods: `getByIdTest`, `createTest`, and `updateTest`. Each method uses `MockMvc` to perform HTTP requests and `andExpect` to verify the response content.

```
53 * @Test
54 public void getByIdTest() throws Exception {
55     Employee entry = new Employee(1L, "Boris", "Johnson", "bjohnson@uk.com", "Administration", "10 Downing Street");
56     String entryAsJSON = this.mapper.writeValueAsString(entry);
57
58     mockMvc.perform(get("/employees/getById/1")
59         .contentType(MediaType.APPLICATION_JSON)
60         .andExpect(content().json(entryAsJSON)));
61
62 }
63
64 * @Test
65 public void createTest() throws Exception {
66     Employee entry = new Employee("Rishi", "Sunak", "rsunak@uk.com", "Finance", "11 Downing Street");
67     String entryAsJSON = mapper.writeValueAsString(entry);
68
69     Employee result = new Employee(1L, "Rishi", "Sunak", "rsunak@uk.com", "Finance", "11 Downing Street");
70     String resultAsJSON = mapper.writeValueAsString(result);
71
72     mockMvc.perform(post("/employee/create")
73         .contentType(MediaType.APPLICATION_JSON)
74         .content(entryAsJSON)
75         .andExpect(content().json(resultAsJSON)));
76
77 }
78
79 * @Test
80 public void updateTest() throws Exception {
81     Employee entry = new Employee ("Bright", "James", "bjj@uk.com", "Admin", "16 Browning Street");
82     Employee result = new Employee(1L, "Bright", "James", "bjj@uk.com", "Admin", "16 Browning Street");
83     String entryAsJSON = this.mapper.writeValueAsString(entry);
84     String resultAsJSON = this.mapper.writeValueAsString(result);
85
86     mockMvc.perform(put("/employee/update/1")
87         .contentType(MediaType.APPLICATION_JSON)
88         .content(entryAsJSON)
89         .andExpect(content().json(resultAsJSON)));
90
91 }
92
93 * @Test
94 public void deleteTest() throws Exception {
95     mockMvc.perform(delete("/employee/delete/1")
96         .contentType(MediaType.APPLICATION_JSON)
97         .andExpect(content().string("true")));
98
99 }
100
101 }
102
```