

## הרשאות למתודה

כדי להשתמש בהרשאות ברמת המתודה אנחנו צרכים באנוטציה `PreAuthorize`

ואנחנו נשתמש בה ממש כמו שהשתמשנו קודם עם `hasAnyAuthority` וכו'.

נגיד ונרצה להחליף את ה `antMatchers` זה ייראה ככה

```
@RestController
@RequestMapping("management/api/v1/students")
public class StudentManagementController {

    private static final List<Student> studentList = List.of(
        new Student( studentId: 1, studentName: "James Bond"),
        new Student( studentId: 2, studentName: "Shay Gali"),
        new Student( studentId: 3, studentName: "Anna Smith")
    );

    @GetMapping
    @PreAuthorize("hasAnyRole('ROLE_ADMIN', 'ROLE_ADMIN_TRAINEE')")
    public List<Student> getStudentList() {
        return studentList;
    }

    @PostMapping
    @PreAuthorize("hasAuthority('student:write')")
    public void registerNewStudent(@RequestBody Student student) {
        System.out.println(student);
    }

    @DeleteMapping("/{id}")
    @PreAuthorize("hasAuthority('student:write')")
    public void deleteStudent(@PathVariable("id") Integer studentId) {
        System.out.println(studentId);
    }

    @PutMapping("/{id}")
    @PreAuthorize("hasAuthority('student:write')")
    public void updateStudent(@PathVariable("id") Integer studentId, @RequestBody Student student) {
        System.out.printf("%s %s\n", studentId, student);
    }
}

// .antMatchers(HttpMethod.DELETE, "/management/api/**").hasAnyAuthority(COURSE_WRITE.getPermission())
// .antMatchers(HttpMethod.POST, "/management/api/**").hasAnyAuthority(COURSE_WRITE.getPermission())
// .antMatchers(HttpMethod.PUT, "/management/api/**").hasAnyAuthority(COURSE_WRITE.getPermission())
// .antMatchers(HttpMethod.GET, "/management/api/**").hasAnyRole(ADMIN.name(), ADMIN_TRAINEE.name())
```

ובconfig שלנו נוסיף

```
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class ApplicationSecurityConfig extends WebSecurityConfigurerAdapter {
```

וזוהו עכשיו יש לנו הרשאות למתדות

האלו