

```
1  /**
2   * Rest-Controller zum Registrieren und für den Login bzw. das Erzeugen eines
3   * JWT.
4   */
5   @RestController
6   @RequestMapping("/api/auth/v1")
7   public class AuthController {
8
9       /**
10        * Erzeugt bei erfolgreicher Authentifizierung ein JWT.
11        *
12        * @param authentication Spring Security Objekt mit Username und Passwort
13        * @return JWT
14        */
15        @GetMapping("/token")
16        public ResponseEntity<String> token(Authentication authentication) {
17        }
18
19        /**
20        * Registriert einen neuen User.
21        *
22        * @param userCredentialsDto request enthält Username, Mail (optional),
23        Password
24        */
25        @PostMapping("/register")
26        public ResponseEntity<Void> register(@Valid @RequestBody UserCredentialsDto
27        userCredentialsDto) {
28        }
29
30        /**
31        * Gibt die Details zu einem User zurück.
32        *
33        * @param principal enthält die UserId und wird durch Spring Security im
34        authentication Prozess erzeugt.
35        * @return Details zum User
36        */
37        @GetMapping("/details")
38        public ResponseEntity<UserDetailsDto> details(Principal principal) {
39        }
40    }
41
42    /**
43    * Rest-Controller zum Erstellen, Betreten und Auflisten von Lobbies.
44    */
45    @RestController
46    @RequestMapping("/api/lobby/v1")
47    public class LobbyController {
48
49        private final LobbyService lobbyService;
50
51        public LobbyController(LobbyService lobbyService) {
52            this.lobbyService = lobbyService;
53        }
54
55        /**
```

```
54     * Erstellt eine neue Lobby.
55     *
56     * @param principal enthält die UserId und wird durch Spring Security im
57     *                   authentication Prozess erzeugt.
58     * @param request   enthält den Namen der zu erstellenden Lobby.
59     * @return neu erstellte Lobby.
60     */
61     @PostMapping("/create")
62     public ResponseEntity<Lobby> create(Principal principal, @RequestBody
CreateRequest request) {
63     }
64
65     /**
66     * Verbinden mit einer Lobby.
67     *
68     * @param principal enthält die UserId und wird durch Spring Security im
69     *                   authentication Prozess erzeugt.
70     * @param request   enthält die LobbyId, mit der sich der User verbinden
möchte.
71     * @return Lobby die der User beigetreten ist.
72     */
73     @PostMapping("/connect")
74     public ResponseEntity<Lobby> connect(Principal principal, @RequestBody
ConnectRequest request) {
75     }
76
77     /**
78     * Eine Lobby verlassen.
79     *
80     * @param principal enthält die UserId und wird durch Spring Security im
81     *                   authentication Prozess erzeugt.
82     * @param request   enthält die LobbyId, die der User verlassen möchte.
83     */
84     @PostMapping("/disconnect")
85     public ResponseEntity<Void> disconnect(Principal principal, @RequestBody
DisconnectRequest request) {
86     }
87
88     /**
89     * Gibt eine angefragte Lobby zurück.
90     *
91     * @param request enthält die LobbyId
92     * @return Lobby
93     */
94     @GetMapping("/get")
95     public ResponseEntity<Lobby> get(@RequestBody LobbyRequest request) {
96     }
97
98     /**
99     * Gibt alle bestehenden Lobbies zurück.
100    *
101    * @return Liste mit allen bestehenden Lobbies.
102    */
103    @GetMapping("/all")
104    public ResponseEntity<List<Lobby>> all() {
105    }
```

```
106 }
107
108 /**
109  * Rest-Controller für das Quiz
110  */
111 @RestController
112 @RequestMapping("/v1")
113 public class QuizController {
114
115     /**
116      * Erstellt eine Quiz-Session.
117      *
118      * @param request enthält die LobbyId und die playerId's.
119      * @return Quiz
120      */
121     @PostMapping("/create")
122     public ResponseEntity<Quiz> create(@RequestBody CreateRequest request) {
123     }
124
125     /**
126      * Gib eine angefragte Quiz-Session zurück.
127      *
128      * @param request enthält die QuizId
129      * @return Quiz
130      */
131     @GetMapping("/get")
132     public ResponseEntity<Quiz> get(@RequestBody QuizRequest request) {
133     }
134 }
```