isec
Politécnico de Coimbra

Licenciatura em Engenharia Informática

Relatório de Trabalho Prático Meta 2
Programação Distribuída

Ana Cardoso - 2021133158
Miguel Stamm - 2021146924
Diogo Pinto - 2020133653

Endpoints

Method	URI	Operation	Description	Request body	Response body
POST	/login	Read	Login	Login object (username and password)	Request Result JWT Token
POST	/register	Create	Registers a User	Register object (user info)	Request Result JWT Token
POST	/codEvent	Create	Register a user in an event	JWT Token Presence Code	Request Result
GET	/presences	Read	Returns the events associated with the user, and supports filters	JWT Token (filtros opcionais:timeBegin,timeEnd,eventDesignation,place)	Request Result EventResult (JSON Object)
GET	/events	Read	Returns every event with filters	JWT Token (has to be admin) (filtros opcionais:timeBegin,timeEnd,eventDesignation,place)	Request Result EventResult (JSON Object)
POST	/events/newEvent	Create	Creates a new Event	JWT Token (has to be admin) String with Event Details	Request Result
DELETE	/events/delete/{eventDesignation}	Delete	Deletes the event by its designation	JWT Token (has to be admin)	Request Result
POST	/events/genCode	Create	Generates a code for a specified event with a specified duration	JWT Token (has to be admin) Event Designation Time Active	Request Result Generated Code
GET	/events/presences/{eventDesignation}	Read	Returns the Presences associated with a specified event	JWT Token (has to be admin)	Request Result EventResult (JSON Object)

A resposta do servidor quando o cliente pede ou os eventos ou as presenças no evento é feita enviando um objeto deste tipo (columns é o nome das colunas e cada elemento do arraylist de eventos contém uma String com várias informações separadas por vírgulas.

```java
public class EventResult implements Serializable {
    private String columns;
    public ArrayList<String> events;

    public EventResult(String columns) {
        this.columns = columns;
        events = new ArrayList<String>();
    }

    public String getColumns() { return columns; }

    public void setColumns(String columns) { this.columns = columns; }
}
```

Enviamos um objeto desta classe quando é para ser efetuado o registo de um novo utilizador.

```java
public class Register implements Serializable {
    private String name;
    private String id;
    private String username; //AKA email address
    private String password;

    public Register(String name, String id, String username, String password) {
        this.name = name;
        this.id = id;
        this.username = username;
        this.password = password;
    }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getId() { return id; }
    public void setId(String id) { this.id = id; }
    public String getUsername() { return username; }
    public void setUsername(String username) { this.username = username; }
    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }
}
```

Nos endpoints que estamos a receber uma String chamada args, estamos basicamente a enviar uma única string com vários parâmetros lá dentro, separados por vírgulas, optamos por fazer assim uma vez que na primeira meta usamos isso e assim teríamos que fazer o menor de alterações possíveis.

```java
@PostMapping("/codEvent")
public ResponseEntity<String> registerUserInEvent(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @RequestBody @NotNull String args
)
```

```java
@GetMapping("/events")
public ResponseEntity<EventResult> getAllEvents(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @RequestParam(value = "timeBegin", required = false, defaultValue = " ") String timeBegin,
    @RequestParam(value = "timeEnd", required = false, defaultValue = " ") String timeEnd,
    @RequestParam(value = "eventDesignation", required = false, defaultValue = " ") String eventDesignation,
    @RequestParam(value = "place", required = false, defaultValue = " ") String place
)
```

```
@DeleteMapping("/events/delete/{eventDesignation}")
public ResponseEntity<String> deleteEvent(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @PathVariable("eventDesignation") String eventDesignation
)
```

```
@PostMapping("/events/genCode")
public ResponseEntity<String> generatePresenceCode(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @RequestBody String args
)
```

```
@PostMapping("/events/newEvent")
public ResponseEntity<String> createEvent(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @RequestBody String args
)
```

```
@GetMapping("/events/presences/{eventDesignation}")
public ResponseEntity<EventResult> getPresencesInEvent(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @PathVariable("eventDesignation") String eventDesignation
)
```

```
@GetMapping("/isAdmin")
public ResponseEntity<String> isAdmin(
    @AuthenticationPrincipal @NotNull Jwt principal
)
```

```
@PostMapping("/login")
public String login(
    Authentication authentication
)
```

```java
@GetMapping("/presences/{username}")
public ResponseEntity<EventResult> getPresencesByUsername(
    @AuthenticationPrincipal @NotNull Jwt principal,
    @PathVariable String username,
    @RequestParam(value = "timeBegin", required = false, defaultValue = " ") String timeBegin,
    @RequestParam(value = "timeEnd", required = false, defaultValue = " ") String timeEnd,
    @RequestParam(value = "eventDesignation", required = false, defaultValue = " ") String eventDesignation,
    @RequestParam(value = "place", required = false, defaultValue = " ") String place
)
```

```java
@PostMapping("/register")
public ResponseEntity<String> register(
    @RequestBody Register register
)
```