



Università degli Studi dell'Aquila

Documentazione progetto

AuleWebREST

Gruppo *Two Lazy Guys*

Anghel Alex – Croce Marco

Esame di Sviluppo Web Avanzato

A.A. 2022-23

Dipendenze Software

- **Client**

Javascript:

jquery-3.6.0.min.js

CSS:

bootstrap.min.css

- **Server**

Apache Commons Proper CSV - 1.10.0

Connettore MySQL-Java - 8.0.26

iCal4j - 4.0.0-beta7

JWT Jackson – 0.11.5

Jackson Jakarta RS JSON Provider – 2.15.0

Jersey HK2 – 3.1.1

Jersey Container Servlet – 3.1.1

Jakarta EE API – 10.0.0

Funzionalità

Le funzionalità richieste nella specifica sono state realizzate tutte, per alcune di esse sono state effettuate delle piccole “modifiche” in quanto considerate funzionali sia dal punto di vista degli utenti finali sia dal punto di vista della struttura dell'applicazione web.

Gli amministratori effettuano il login utilizzando *email* e *password*, in quanto all'interno del sistema non viene associata una username univoca ad ogni amministratore, ma egli viene identificato mediante l'indirizzo email.

Per quanto riguarda la “lettura delle informazioni di base relative a un'aula” è stato deciso che le informazioni di base sono: *nome*, *luogo*, *edificio*, *piano* ed *email* del responsabile.

Per quanto riguarda “l'esportazione di tutti gli eventi relativi a un certo intervallo di tempo, possibilmente in formato iCalendar”, ad ogni evento è stata associata una proprietà *summary* in cui vengono riportati il nome dell'evento, la descrizione dell'evento, il nome dell'aula in cui si terrà l'evento e l'edificio in cui si trova l'aula.

Inoltre, è stata implementata una funzionalità extra molto semplice, essa permette di “Leggere le informazioni relative ad un gruppo”, difatti utilizza un parametro numerico che rappresenta l'ID del gruppo di cui leggere le informazioni e, tramite esso, ricava e restituisce il *nome* e la *descrizione* associati a quel gruppo, oltre a una *lista* di URL che identificano tutte le aule che appartengono al gruppo stesso.

Nel client sono state implementate le operazioni richieste dalla specifica, ma anche “l'esportazione di tutti gli eventi relativi ad un certo intervallo di tempo in formato iCalendar” e “la lettura delle informazioni di base relative a un'aula”.

NOTA:

Il progetto è stato sviluppato da un gruppo di due persone, in linea di massima il lavoro è stato svolto “insieme”, anche se *Marco Croce* si è dedicato maggiormente allo sviluppo del codice in Java, mentre *Alex Anghel* si è dedicato maggiormente alla realizzazione del client.

Specifica dettagliata dell'API

```
openapi: 3.0.1
info:
  title: aulewebREST
  description: "Corso di Sviluppo Web Avanzato - Specifica AuleWeb Services"
  version: '1.0'
servers:
  - url: 'http://localhost:8080/aulewebREST'
paths:
  #1
  '/rest/auth/login':
    post:
      operationId: doLoginAdmin
      tags:
        - autenticazione amministratore
      requestBody:
        content:
          application/x-www-form-urlencoded:
            schema:
              type: object
              properties:
                email:
                  type: string
                password:
                  type: string
      responses:
        '200':
          description: login riuscito
          content:
            text/plain:
              schema:
                type: string
          headers:
            Authorization:
              schema:
                type: string
                pattern: 'Bearer [a-z0-9A-Z-]+'
        '401':
          $ref: '#/components/responses/UnauthorizedError'
        '500':
          $ref: '#/components/responses/GenericError'
  #1
  '/rest/auth/logout':
    delete:
      operationId: doLogoutAdmin
      tags:
        - autenticazione amministratore
      security:
```

```

    - bearerAuth: [ ]
  responses:
    '200':
      description: logout effettuato
    '401':
      $ref: '#/components/responses/UnauthorizedError'
    '500':
      $ref: '#/components/responses/GenericError'
#2
'/rest/gruppi/configuration':
  get:
    operationId: exportCSVConfiguration
    tags:
      - configurazione gruppi-aule
    security:
      - bearerAuth: [ ]
    responses:
      default:
        description: lista delle configurazioni gruppi-aule in CSV
        content:
          text/csv:
            schema:
              type: string
      '401':
        $ref: '#/components/responses/UnauthorizedError'
      '500':
        $ref: '#/components/responses/GenericError'
  post:
    operationId: importCSVConfiguration
    tags:
      - configurazione gruppi-aule
    security:
      - bearerAuth: [ ]
    requestBody:
      content:
        multipart/form-data:
          schema:
            type: object
            properties:
              file:
                type: string
                format: binary
    responses:
      '204':
        description: no content
      '401':
        $ref: '#/components/responses/UnauthorizedError'
      '500':
        $ref: '#/components/responses/GenericError'

```

#3

```

'/rest/aule':
  get:
    operationId: getAule
    tags:
      - aule
    responses:
      '200':
        description: lista delle aule
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ListaAule'
      '500':
        $ref: '#/components/responses/GenericError'
  post:
    operationId: addAula
    tags:
      - aule
    security:
      - bearerAuth: [ ]
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Aula'
    responses:
      '201':
        description: l'aula è stata creata con successo
        headers:
          location:
            description: url dell'aula creata
            schema:
              type: string
              format: uri
        content:
          application/json:
            schema:
              type: string
              format: uri
      '401':
        $ref: '#/components/responses/UnauthorizedError'
      '500':
        $ref: '#/components/responses/GenericError'
#4
'/rest/gruppi/{gruppo}/{aula}':
  post:
    operationId: setAulaforGruppo
    tags:
      - configurazione gruppi-aule
    security:

```

```

    - bearerAuth: [ ]
  parameters:
    - $ref: '#/components/parameters/idGruppo'
    - $ref: '#/components/parameters/idAula'
  requestBody:
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ConfigurazioneGruppoAula'
  responses:
    '204':
      description: no content
    '401':
      $ref: '#/components/responses/UnauthorizedError'
    '500':
      $ref: '#/components/responses/GenericError'

```

#5

```

'/rest/aule/{aula}':
  get:
    operationId: getAula
    tags:
      - aula
    parameters:
      - $ref: '#/components/parameters/idAula'
    responses:
      '200':
        description: l'aula è stata trovata
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InfoBaseAula'
      '404':
        $ref: '#/components/responses/NotFound'
      '500':
        $ref: '#/components/responses/GenericError'

```

#6

```

'/rest/aule/{aula}/attrezzature':
  get:
    operationId: getAttrezzatureinAula
    tags:
      - aula
    parameters:
      - $ref: '#/components/parameters/idAula'
    responses:
      '200':
        description: lista delle attrezzature all'interno di un'aula
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ListaAttrezzature'

```

```
    '404':
      $ref: '#/components/responses/NotFound'
    '500':
      $ref: '#/components/responses/GenericError'
#7 | #10
'/rest/eventi':
  get: #10
    operationId: getEventi
    tags:
      - eventi
    parameters:
      - name: aula
        in: query
        required: false
        schema:
          type: integer
          format: int32
          example: 1
      - name: from
        in: query
        required: false
        schema:
          type: string
          format: date
          example: "2001-01-01"
      - name: to
        in: query
        required: false
        schema:
          type: string
          format: date
          example: "2001-01-30"
    responses:
      '200':
        description: lista degli eventi
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ListaEventi'
      '500':
        $ref: '#/components/responses/GenericError'
  post: #7
    operationId: addEvento
    tags:
      - eventi
    security:
      - bearerAuth: [ ]
    requestBody:
      content:
        application/json:
```



```

        schema:
          $ref: '#/components/schemas/Evento'
responses:
  '201':
    description: l'evento è stato creato con successo
    headers:
      location:
        description: url dell'evento creato
        schema:
          type: string
          format: uri
    content:
      application/json:
        schema:
          type: string
          format: uri
  '401':
    $ref: '#/components/responses/UnauthorizedError'
  '500':
    $ref: '#/components/responses/GenericError'
#8 | #9
'/rest/eventi/{evento}':
  parameters:
    - $ref: '#/components/parameters/idEvento'
  get: #9
    operationId: getEvento
    tags:
      - evento
    responses:
      '200':
        description: l'evento è stato trovato
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Evento'
      '404':
        $ref: '#/components/responses/NotFound'
      '500':
        $ref: '#/components/responses/GenericError'
  put: #8
    operationId: updateEvento
    tags:
      - evento
    security:
      - bearerAuth: [ ]
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Evento'

```

```

    responses:
      '204':
        description: no content
      '401':
        $ref: '#/components/responses/UnauthorizedError'
      '404':
        $ref: '#/components/responses/NotFound'
      '500':
        $ref: '#/components/responses/GenericError'
#9
'/rest/eventiricorrenti/{eventoRic}':
  get:
    operationId: getEventoRicorrente
    tags:
      - evento ricorrente
    parameters:
      - $ref: '#/components/parameters/idEventoRic'
    responses:
      '200':
        description: l'evento ricorrente è stato trovato
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Evento'
      '404':
        $ref: '#/components/responses/NotFound'
      '500':
        $ref: '#/components/responses/GenericError'
#11
'/rest/eventi/attuali':
  get:
    operationId: getEventiAttuali
    tags:
      - eventi
    responses:
      '200':
        description: lista degli eventi attuali e delle prossime tre ore
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ListaEventi'
      '500':
        $ref: '#/components/responses/GenericError'
#12
'/rest/eventi/export':
  get:
    operationId: getEventibetweenTimeInterval
    tags:
      - eventi
    parameters:

```

```

- name: from
  in: query
  required: true
  schema:
    type: string
    format: date
    example: "2001-01-01"
- name: to
  in: query
  required: true
  schema:
    type: string
    format: date
    example: "2001-01-30"
responses:
  default:
    description: lista degli eventi relativi all'intervallo di tempo
    content:
      text/calendar:
        schema:
          type: string
    '500':
      $ref: '#/components/responses/GenericError'
#EXTRA 1
'/rest/gruppi':
  get:
    operationId: getGruppi
    tags:
      - gruppi
    responses:
      '200':
        description: lista dei gruppi
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ListaGruppi'
      '500':
        $ref: '#/components/responses/GenericError'
#EXTRA 2
'/rest/gruppi/{gruppo}':
  get:
    operationId: getGruppo
    tags:
      - gruppo
    parameters:
      - $ref: '#/components/parameters/idGruppo'
    responses:
      '200':
        description: il gruppo è stato trovato
        content:

```

```

        application/json:
          schema:
            $ref: '#/components/schemas/Gruppo'
      '404':
        $ref: '#/components/responses/NotFound'
      '500':
        $ref: '#/components/responses/GenericError'
components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
      bearerFormat: internalToken
  responses:
    GenericError:
      description: errore generico
      content:
        text/plain:
          schema:
            type: string
            example: "Messaggio di errore!"
    UnauthorizedError:
      description: bearer token mancante o non valido
    NotFound:
      description: l'elemento richiesto non è stato trovato
      content:
        text/plain:
          schema:
            type: string
            example: "Elemento non trovato!"

```

PARAMETERS

```

parameters:
  idAula:
    name: aula
    in: path
    required: true
    schema:
      type: integer
      format: int32
      example: 1
  idEvento:
    name: evento
    in: path
    required: true
    schema:
      type: integer
      format: int32
      example: 1

```

```
idEventoRic:
  name: eventoRic
  in: path
  required: true
  schema:
    type: integer
    format: int32
    example: 1
idGruppo:
  name: gruppo
  in: path
  required: true
  schema:
    type: integer
    format: int32
    example: 1
idAttrezzatura:
  name: attrezzatura
  in: path
  required: true
  schema:
    type: integer
    format: int32
    example: 1
```

SCHEMAS

```
schemas:
  Evento:
    type: object
    properties:
      ID:
        type: integer
        format: int32
        readOnly: true
      data_inizio:
        type: string
        format: date-time
        pattern: "DD-MM-YYYY HH:MM"
        example: "01-01-2001T08:00"
      data_fine:
        type: string
        format: date-time
        pattern: "DD-MM-YYYY HH:MM"
        example: "01-01-2001T09:00"
      nome:
        type: string
        example: "Seminario sulla CyberSecurity"
      descrizione:
        type: string
```

```
    example: "Cos'è la sicurezza informatica?"
email_responsabile:
  type: string
  format: email
  example: "resp@resp.uni.it"
aula:
  $ref: '#/components/schemas/Aula'
tipologia:
  $ref: '#/components/schemas/Tipologia'
nome_corso:
  type: string
  example: "ANALISI MATEMATICA I"
  nullable: true
tipo_ricorrenza:
  $ref: '#/components/schemas/Ricorrenza'
data_fine_ricorrenza:
  type: string
  format: date
  nullable: true
  pattern: "DD-MM-YYYY"
  example: "01-01-2001"
Aula:
  type: object
  properties:
    ID:
      type: integer
      format: int32
      readOnly: true
    nome:
      type: string
      example: "Aula Rossa"
    luogo:
      type: string
      example: "Coppito"
    edificio:
      type: string
      example: "Alan Turing"
    piano:
      type: integer
      format: int32
      minimum: 0
      maximum: 3
      example: 2
    capienza:
      type: integer
      format: int32
      minimum: 1
      example: 25
    email_responsabile:
      type: string
```

```
    format: email
    example: "resp@resp.uni.it"
  numero_prese_rete:
    type: integer
    format: int32
    example: 10
  numero_prese_elettriche:
    type: integer
    format: int32
    example: 10
  note:
    type: string
    example: "L'aula ha una finestra rotta!"
  attrezzatura:
    $ref: '#/components/schemas/ListaAttrezzature'
```

```
Attrezzatura:
  type: object
  properties:
    numero_seriale:
      type: string
      example: "123456ABCD"
    descrizione:
      type: string
      example: "Proiettore Epson"
```

```
Gruppo:
  type: object
  properties:
    aule:
      $ref: '#/components/schemas/ListaAule'
```

```
  ID:
    type: integer
    format: int32
    readOnly: true
  nome:
    type: string
    example: "DISIM"
  descrizione:
    type: string
    nullable: true
    example: "Dipartimento di Ingegneria e Scienze dell'Informazione e
```

Matematica"

```
  InfoBaseAula:
    type: object
    properties:
      nome:
        type: string
        example: "Aula Rossa"
      luogo:
        type: string
        example: "Coppito"
```

```
    edificio:
      type: string
      example: "Alan Turing"
    piano:
      type: integer
      format: int32
      minimum: 0
      maximum: 3
      example: 1
    email_responsabile:
      type: string
      format: email
      example: "resp@resp.uni.it"
  ConfigurazioneGruppoAula:
    type: object
    properties:
      gruppo:
        type: integer
        format: int32
        example: 1
      aula:
        type: integer
        format: int32
        example: 1
  ListaEventi:
    type: array
    items:
      type: string
      format: uri
      example: 'http://localhost:8080/aulewebREST/rest/eventi/7,
http://localhost:8080/aulewebREST/rest/eventiricorrenti/4'
  ListaAule:
    type: array
    items:
      type: string
      format: uri
      example: 'http://localhost:8080/aulewebREST/rest/aule/7,
http://localhost:8080/aulewebREST/rest/aule/4'
  ListaGruppi:
    type: array
    items:
      type: string
      format: uri
      example: 'http://localhost:8080/aulewebREST/rest/gruppi/1,
http://localhost:8080/aulewebREST/rest/gruppi/3'
  ListaAttrezzature:
    type: array
    items:
      $ref: '#/components/schemas/Attrezzatura'
```



```
##### ENUMS #####
```

Ricorrenza:

type: string

nullable: true

enum:

- GIORNALIERA
- SETTIMANALE
- MENSILE
- NULL

Tipologia:

type: string

enum:

- LEZIONE
- ESAME
- SEMINARIO
- PARZIALE
- RIUNIONE
- LAUREE
- ALTRO