Backend

J2EE Web module

Versiones

Java SE 17

Servidor: Tomcat 9.0

**Endpoints**

<http://localhost:8080/backend/AlumnosController>

* GET Obtener lista de alumnos
* GET <matricula> Obtener alumnos con grupo
* POST Registrar alumno
* PUT Actualizar información de un alumno
* DELETE Eliminar alumno

<http://localhost:8080/backend/GrupoController>

* GET Obtener lista de grupos
* POST Registrar grupo
* PUT Actualizar información de un grupo
* DELETE Eliminar grupo

<http://localhost:8080/backend/InscripcionController>

* GET Obtener lista de grupos
* GET <matricula> Obtener grupos del alumno
* POST Inscribir alumno en grupo
* PUT Actualizar información de un alumno
* DELETE Dar de baja alumno en grupo

<http://localhost:8080/backend/LoginController>

* POST Iniciar Sesión
* PUT Validar token

<http://localhost:8080/backend/AlumnosController>

**Obtener lista de alumnos**

Método: GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo en JSON con el siguiente formato de alumnos

[{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo"}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

**Obtener alumno con grupos**

Método: GET

Parámetros: matricula

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un JSON con información del alumno

{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo", Grupos: []}

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

**Registrar alumno**

Método: POST

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

[{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo"}, …]

Status 200 – OK: Retorna un JSON con información del alumno

[{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo"}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Actualizar información de un alumno**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

[{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo"}, …]

Status 200 – OK: Retorna un arreglo JSON con los alumnos

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Eliminar alumno**

Método: DELETE

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“Matricula": 1234567}

Status 200 – OK: Retorna un arreglo JSON con los alumnos

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

<http://localhost:8080/backend/GrupoController>

**Obtener lista de grupos**

Método: GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo en JSON con el siguiente formato de grupos

[{“IdGrupo": 0, “CveMateria": "", "Estatus": "", "CantidadAlumnos": "", "Nombre": ""} …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

**Registrar grupo**

Método: POST

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“Clave”: 0, CantidadAlumnos: 0}

Status 200 – OK: Retorna un arreglo en JSON con el siguiente formato de grupos

[{“IdGrupo": 0, “CveMateria": "", "Estatus": "", "CantidadAlumnos": "", "Nombre": ""} …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Actualizar información de un grupo**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“CveMateria": "", "Estatus": "", "CantidadAlumnos": ""}

Status 200 – OK: Retorna un arreglo en JSON con el siguiente formato de grupos

[{“IdGrupo": 0, “CveMateria": "", "Estatus": "", "CantidadAlumnos": "", "Nombre": ""} …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Eliminar grupo**

Método: DELETE

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“Clave": ""}

Status 200 – OK: Retorna un arreglo JSON con los grupos

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

<http://localhost:8080/backend/InscripcionController>

**Obtener lista de grupos**

Método: GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo JSON con información de los grupos

[{"IdGrupo": 0, "CantidadAlumnos": 0, "AlumnosInscritos": 0, "Clave": 0, "Nombre": "", "Estatus": ""}]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

**Obtener grupos del alumno**

Método: GET

Parámetros: Usuario

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un JSON con el siguiente formato

{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo", Grupos: []}

Status 401 – No Autorizado: El token no es valido

**Inscribir alumno en grupo**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

[{“Matricula": 1234567, “Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": "Activo"}, …]

Status 200 – OK: Retorna un JSON con información del alumno

[{"IdGrupo": 0, "CantidadAlumnos": 0, "AlumnosInscritos": 0, "Clave": 0, "Nombre": "", "Estatus": ""}]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Actualizar información de un alumno**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“Matricula": 0, “IdGrupo”: 0, “Semestre”: 0}

Status 200 – OK: Retorna un arreglo JSON con los alumnos

[{"IdGrupo": 0, "CantidadAlumnos": 0, "AlumnosInscritos": 0, "Clave": 0, "Nombre": "", "Estatus": ""}]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Dar de baja alumno en grupo**

Método: DELETE

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“Matricula": 0, “IdGrupo”: 0}

Status 200 – OK: Retorna un arreglo JSON con los grupos

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

<http://localhost:8080/backend/LoginController>

**Iniciar Sesión**

Método POST

Body:

{"Usuario": "", "Contrasena": ""}

Status 200 – OK: Retorna un JSON con el siguiente formato

{"usuario": "", "contrasena": "", "token": “”, “rol”: “”}

Status 401 – No Autorizado: Las credenciales son invalidas

**Validar token**

Método PUT

Body:

{"token": ""}

Status 200 – OK: Retorna True o False dependiendo de la validez del token

<http://localhost:8080/backend/MaestroController>

**Obtener lista de maestros**

Método GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo en JSON con el siguiente formato

[{"NoEmpleado": 0, "Tipo": {"IdTipo": 0, "Tipo": ""}, "Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": ""}, …]

Status 401 – No Autorizado: Las credenciales son invalidas

**Registrar maestro**

Método POST

Body:

{"NoEmpleado": 0, "Nombre": "", "ApellidoPaterno": "", "ApellidoMaterno": "", "Usuario": “", "Contrasena": "", "Tipo": {"IdTipo": 0}}

Status 200 – Retorna un arreglo de maestros

[{"NoEmpleado": 0, "Tipo": {"IdTipo": 0, "Tipo": ""}, "Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": ""}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Actualizar información de un maestro**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{"Nombre": "", "ApellidoPaterno": "", "ApellidoMaterno": "", "Contrasena": "", "Tipo": {"IdTipo": 0}}

Status 200 – OK: Retorna un arreglo JSON con los maestros

[{"NoEmpleado": 0, "Tipo": {"IdTipo": 0, "Tipo": ""}, "Nombre": "", "ApellidoMaterno": "", "ApellidoPaterno": "", "Usuario": "", "Estatus": ""}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Eliminar maestro**

Método: DELETE

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

{“NoEmpleado": 0}

Status 200 – OK: Retorna un arreglo JSON con los maestros

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

<http://localhost:8080/backend/MateriaController>

**Obtener lista de materias**

Método: GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo JSON con información de las materias

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Agregar materia**

Método: POST

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 200 – OK: Retorna un arreglo JSON con información de las materias

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Actualizar materia**

Método: PUT

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 200 – OK: Retorna un arreglo JSON con información de las materias

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

**Eliminar materia**

Método: DELETE

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Body:

[{"Clave": 0}, …]

Status 200 – OK: Retorna un arreglo JSON con información de las materias

[{"Clave": 0, "Nombre": 0, “Estatus":””}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error

<http://localhost:8080/backend/TipoController>

**Obtener lista de tipo de maestros**

Método: GET

Headers: Se necesita de un token JWT el cual se genera al iniciar sesión

Status 200 – OK: Retorna un arreglo JSON con información de las materias

[{"IdTipo": 0, "Tipo":””}, …]

Status 401 – No Autorizado: Retorna un arreglo vacío ya que el token no es valido

Status 409 – Conflict: Retorna un mensaje de error