Documentación de la API de Spring

Ruta Base: https://apispringadriancentenojosemanuelreyes.onrender.com

Endpoints

POST /login

Inicio de sesion de usuario.

Parámetros

username
password

Respuestas

Si esta todo correcto:

- 200 OK:
  {
  "data": {
  "idFamilia"
  "id"
  "ROL"
  "token"
  },
  "success": true,
  "message": "Inicio de sesión exitoso"
  }

Si la contraseña o email son incorrectos:

- 401 Unauthorized:
  {
  "success": false,
  "message": "Usuario o contraseña incorrectos"
  }

- 403 Forbidden: falta algun parametro.
- 500 Internal Server Error: si ocurre un error en el servidor.

POST /register

Crea un nuevo usuario.

Parámetros (en Body como raw, json)

- `nombre` : Nombre del usuario (obligatorio)
- `apellidos` : Apellidos del Usuario (Obligatorio)
- `username` : Correo electrónico del usuario (obligatorio)
- `password` : Password del usuario (obligatorio)
- `idFamilia` : IdFamilia del usuario (obligatorio)

{
"nombre": "x",
"apellidos": "x",
"username": "x",
"password": "x",
"idFamilia": x
}

Respuestas

- 201 Created:
  {
  "success": true,
  "message": "Usuario registrado con éxito"
  }
- 500 Internal Server Error: si ocurre un error en el servidor.
  *Usuario duplicado
  {
  "success": false,
  "message": "Error al registrar el usuario: could not execute statement [Duplicate entry 'marta.florin@example.com' for key 'alumno.UK_dlqu0nadbeiwn48uty3rus51y'] [insert into alumno (apellidos,deleted,enabled,familia_id,nombre,password,role,token,email,id) values (?,?,?,?,?,?,?,?,?,?)]; SQL [insert into alumno

(apellidos,deleted,enabled,familia_id,nombre,password,role,token,email,id) values (?,?,?,?,?,?,?,?,?,?)]; constraint [alumno.UK_dlqu0nadbeiwn48uty3rus51y]"
}
*Falta parametro
{
"success": false,
"message": "Error al registrar el usuario: Cannot invoke "String.matches(String)" because "email" is null"
}

GET /api/Admin/empresas

Obtiene las empresas registradas

Headers

- `Authorization` : Token devuelto por el login (obligatorio)

Respuestas

- 200 OK: devuelve las empresas

{
"data": [
{
"id"
"nombre"
"direccion"
"telefono"
"email"
"logo"
"deleted"
},
{
"id"
"nombre"
"direccion"
"telefono"
"email"
"logo"

"deleted"
}
],
"success": true,
"message": "Empresas obtenidas con éxito"
}

403 Forbidden: Si no tienes permisos (por ejemplo si el token no corresponde con un usuario con rol Admin)

- 500 Internal Server Error: si ocurre un error en el servidor.

Post /api/Admin/addEmpresa

Crea una empresa específica.

Parámetros

- `nombre` : Nombre de la Empresa (Obligatorio)
- `direccion` : Direccion de la Empresa (Obligatorio)
- `telefono` : Telefono de la Empresa (Obligatorio)
- `email` : Email de la Empresa (Obligatorio)
- `logo` : Logo de la Empresa (Obligatorio)

Headers:

- `Authorization` : token devuelto por el login

Respuestas

- 200 OK: si la Empresa se crea correctamente y devuelve sus datos.
  {
  "data": {
  "id"
  "nombre"
  "direccion"
  "telefono"
  "email"
  "logo"

"deleted"
},
"success": true,
"message": "Empresa creada con éxito"
}

- 400 Bad Request: si los parámetros son inválidos (estan vacios).
  {
  "success": false,
  "message": "La x de la empresa es obligatoria"
  }
- 403 Forbidden: si falta algun parametro o el token no es valido.
- 500 Internal Server Error: si ocurre un error en el servidor.

Put /api/Admin/updateEmpresa/

Actualiza los parametros de una empresa específica.

Parámetros

- `nombre` : Nombre de la Empresa (Opcional)
- `direccion` : Direccion de laEmpresa (Opcional)
- `telefono` : Telefono de la Empresa (Opcional)
- `email` : Email de la Empresa (Opcional)
- `logo` : Logo de la Empresa (Opcional)

Headers:

- `Authorization` : token devuelto por el login

Respuestas

- 200 OK: si la Empresa existe, se actualiza correctamente y devuelve sus datos.
  {
  "data": {
  "id"
  "nombre"
  "direccion"
  "telefono"
  "email"

"logo"
"deleted"
},
"success": true,
"message": "Empresa actualizada con éxito"
}

- 400 Bad Request: si los parámetros son inválidos (estan vacios).
  {
  "success": false,
  "message": "La dirección de la empresa es obligatoria"
  }
- 403 Forbidden: el token no es valido.
- 500 Internal Server Error: si ocurre un error en el servidor.

DELETE /api/Admin/deleteEmpresa/

Elimina una Empresa específica.

Parámetros

- `id` : ID de la Empresa (obligatorio)

Respuestas

- 200 Ok:
  {
  "success": true,
  "message": "Empresa eliminada con éxito"
  }
- 403 Forbidden: si falta algun parametro o el token no es valido.
- 500 Internal Server Error: si ocurre un error en el servidor
  *La empresa tiene Servicios asociados o no existe
  {
  "success": false,
  "message": "Error al eliminar la empresa"
  }

GET /api/getFamilias

Obtiene todas las Familias Profesionales existentes.

Respuestas

- 200 OK: Devuelve todas las Familias Profesionales
  {
  "data": [
  {
  "id"
  "nombre"
  },
  ],
  "success": true,
  "message": "Familias profesionales obtenidas con éxito"
  }
- 500 Internal Server Error: si ocurre un error en el servidor.

GET /apiAlumno/getServicios/

Obtiene los servicios del alumno, de su familia profesional tanto asignados como no.

Parámetros

- `idAlumno` : ID del alumno (obligatorio)

Respuestas

- 200 Ok:
  {
  "data": [
  {
  "id"
  "titulo"
  "descripcion"
  "fechaAlta"
  "fechaRealizacion"
  "idAlumno"
  "idEmpresa"

"idFamilia"
"valoracionEmpresa"
"realizado"
"comentarioEmpresa"
},
{
"id"
"titulo"
"descripcion"
"fechaAlta"
"fechaRealizacion"
"idAlumno"
"idEmpresa"
"idFamilia"
"valoracionEmpresa"
"realizado"
"comentarioEmpresa"
}
],
"success": true,
"message": "Servicios obtenidos con exito"
}

- 403 Forbidden: si falta algun parametro o el token no es valido.
  {
  "success": false,
  "message": "No tienes permiso para ver estos servicios"
  }
- 500 Internal Server Error: si ocurre un error en el servidor
  *La empresa tiene Servicios asociados o no existe,
  idAlumno no existe (lo ponemos asi para no dar informacion sobre los id que existen)
  {
  "success": false,
  "message": "Error al obtener los servicios"
  }

POST apiAlumno/setServicio

Asigna al alumno un servicio

Parámetros

- `idAlumno` : ID del alumno (obligatorio)
- `idServicio` : ID del servicio (obligatorio)

Respuestas

- 200 OK:
  {
  "success": true,
  "message": "El servicio asignado con exito"
  }
- 403 Bad Request:
  {
  "success": false,
  "message": "El servicio ya esta asignado a otro alumno"
  }
  {
  "success": false,
  "message": "El servicio con el id proporcionado no existe"
  }
  {
  "success": false,
  "message": "No hay usuario con ese Id"
  }
- 403 Forbidden: si falta algun parametro o el token no es valido.
  *Si el token no corresponde con el token del usuario del idAlumno introduce (no puedes asignar servicios a otros alumnos)
  {
  "success": false,
  "message": "No tienes permisos "
  }
- 500 Internal Server Error: si ocurre un error en el servidor.