

Documentación Spring API

1. Parte de Spring.

1.1. Acceso a la aplicación.

- **Login:** <https://ud4springgrupo1.onrender.com/api/login>

Parámetros: email y password.

Ejemplo:

<https://ud4springgrupo1.onrender.com/api/login?email=student2@example.com&password=password123>

- **Register:** <https://ud4springgrupo1.onrender.com/api/register>

Parámetros necesarios: username, name, surname, email, password y professional family.

Ejemplo: En el body, en el apartado de “Raw”, poner lo siguiente:

```
{
  "username": "a",
  "name": "p1",
  "surname": "prueba1",
  "email": "pruebecita@example.com",
  "password": "password123",
  "professionalFamily": {"id": 1}
}
```

1.2. Funcionalidades del alumno.

Para estos 3 endpoints, le pasamos el token.

- **Recoger todos los servicios del alumno logueado:**
<https://ud4springgrupo1.onrender.com/student/viewServices>
- **Recoger todos los servicios asignados del alumno logueado:**
<https://ud4springgrupo1.onrender.com/student/viewServicesAssigned>
- **Recoger todos los servicios no asignados del alumno logueado:**
<https://ud4springgrupo1.onrender.com/student/viewServicesUnassigned>

1.3. Funcionalidades de la empresa.

Lo mismo que hacemos con el alumno, le pasamos el token.

- **Recuperar todos los servicios de la empresa logueada:**

<https://ud4springgrupo1.onrender.com/business/servicios>

Este no hay que pasarle ningún parámetro porque devuelve todos los servicios.

- **Recuperar un determinado servicio de la empresa logueada:**

<https://ud4springgrupo1.onrender.com/business/servicios/4>

En este endpoint, como puedes observar, se le pasa al final del servicios/ un número que sería el id del servicio, que en este caso te puse como ejemplo el servicioid = 4. Si da error es porque la empresa que está logueada no tiene ese servicio (es de otra empresa).

- **Recuperar todos los servicios de la empresa logueada filtrando por familia profesional:**

<https://ud4springgrupo1.onrender.com/business/servicios/proFam/1>

En este endpoint, se hace justo igual que el anterior, pero en vez de pasarle el servicioid, se le pasa el id de la familia profesional que en este caso puse como ejemplo la familia profesional con id 1.

- **Crear un nuevo servicio por parte de la empresa logueada:**

<https://ud4springgrupo1.onrender.com/business/newServicio>

Parámetros requeridos: title, description y profesionalFamilyId.

Ejemplo:

```
{
  "title" : "ServicioQueso554",
  "description" : "caserio354",
  "professionalFamilyId" : {
    "id" : 2
  }
}
```

- **Actualizar un servicio por parte de la empresa logueada:**

<https://ud4springgrupo1.onrender.com/business/servicios/52>

Este método es exactamente igual que el endpoint de delete de abajo, solo que en vez de ser un método DELETE, sería PUT.

Para editar los parámetros son los mismos que pone arriba en el crear y el body también es igual

- **Eliminar un servicio por parte de la empresa logueada:**

<https://ud4springgrupo1.onrender.com/business/servicios/52>

A este método se le pasa también el id del servicio al final y es un método DELETE. Cambia el id 52 por el id del servicio que quieras eliminar.

Juan José Piñero Orellana.
Alejandro Rodríguez Fénix.
Fernando Villalba Varela.