

En la capa “Service” del servidor Webflux es dónde se construyen las personas que se devuelven al cliente. Allí, a través del método “delayElements” le indico una duración determinada en segundos de “delay” que quiero que se construyan esas personas.

```
@Override
public Flux<Person> personList1() {
    return Flux.just(new Person("Angel", "Fuente", 20)).delayElements(Duration.ofSeconds(5));
}

@Override
public Flux<Person> personList2() {
    Flux<Person> personFlux = Flux.just(new Person("Jaime", "Tirado", 50)).delayElements(Duration.ofSeconds(1));
    return personFlux;
}

@Override
public Flux<Person> personList3() {
    // TODO Auto-generated method stub
    return Flux.just(new Person("Manuel", "Perez", 40)).delayElements(Duration.ofSeconds(8));
}

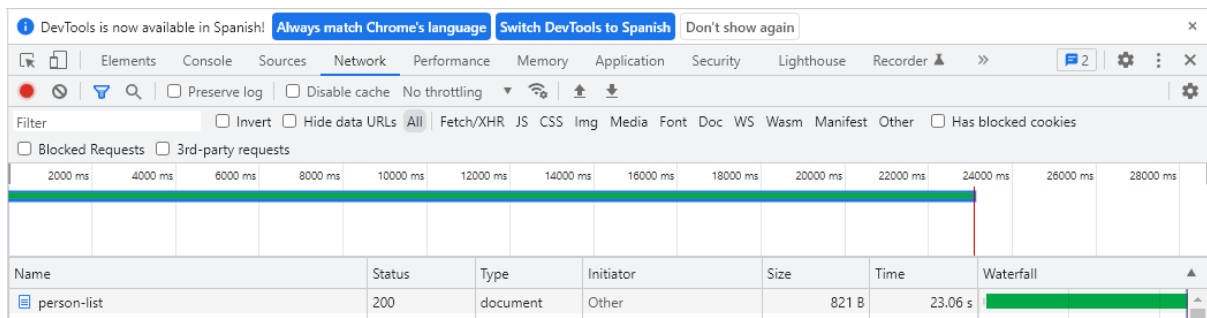
@Override
public Flux<Person> personList4() {
    return Flux.just(new Person("Antonio", "Vargas", 30)).delayElements(Duration.ofSeconds(9));
}
```

Como se puede ver, se tarda un total de 23 segundos a la hora de crear los cuatro flujos con una persona creada dentro de cada uno. (5+1+8+9 = 23s).

Una vez arrancado el servidor y cliente Webflux, indico esta dirección en mi navegador.



Como se puede ver en la imagen, se muestra una lista con las cuatro personas creada en el servidor. Y a través de la consola (apartado Network) del navegador se podrá ver cuanto tiempo ha tardado en devolverlos.



Efectivamente como se puede ver en la imagen, “person-list” ha tardado un total de 23,06s en devolver los flujos con las personas.