

# Asynchronous



# Why?

- We have two separate tasks. One takes 2 seconds, the other one 3 seconds.
- Executing them synchronous will take at least **5 seconds**.
- Executing them asynchronous will take at least **3 seconds**.

# POM dependencies

```
<dependency>  
  <groupid>org.springframework.boot</groupid>  
  <artifactid>spring-boot-starter-web</artifactid>  
</dependency>
```

# SleepService

```
@Component
public class SleepService {

    @Async
    public void sleep() throws InterruptedException {
        Thread.sleep(3000L);
    }
}
```

# Asynchronous example

```
@SpringBootApplication
@EnableAsync
public class App implements CommandLineRunner {
    ...

    @Autowired
    private SleepService sleepService;

    public void run(String... strings) throws InterruptedException {
        long startTime = System.currentTimeMillis();
        sleepService.sleep();
        sleepService.sleep();
        long endTime = System.currentTimeMillis();
        long time = endTime - startTime;
        System.out.println("Elapsed time: " + time);
    }
}
```

# Time to run the application

Synchronous

Elapsed time: 6001

---

Asynchronous

Elapsed time: 21

A future for your result

# SleepService

```
@Component
public class SleepService {

    @Async
    public Future sleep() throws InterruptedException {
        Thread.sleep(3000L);
        return new AsyncResult("OK");
    }
}
```



# Future example

```
public void run(String... strings) throws InterruptedException,
    ExecutionException, TimeoutException {
    long startTime = System.currentTimeMillis();
    Future result1 = sleepService.sleep();
    Future result2 = sleepService.sleep();

    System.out.println(
        "Result 1: " + result1.get(5000L, TimeUnit.MILLISECONDS));
    System.out.println(
        "Result 2: " + result2.get(5000L, TimeUnit.MILLISECONDS));

    long endTime = System.currentTimeMillis();
    long time = endTime - startTime;
    System.out.println("Elapsed time: " + time);
}
```

Result 1: OK

Result 2: OK

Elapsed time: 3040