Poleposition with Spring Boot and Micrometer

Ko Turk

You’re part of a racing team


@REDBULLRACING
edBull
Mobil 1
LAT IMAGES

https://youtu.be/unszkypamzc?t=688

MONITORING


Image from:
https://www.formula1.com
Charles Leclerc
DRIVER OF THE DAY
But now you are the driver (developer)


Image from:
https://img.redbull.com
Red Bull
RACING
Red Bull
RAUCH
BYBIT
Tezos
Mobil 1
ORACLE
infiniti
citrix
RAUCH
inter.mx
HRC
Honda Racing
ARCTIC WOLF
Or ;-)

Our goal

PURE PITWALL

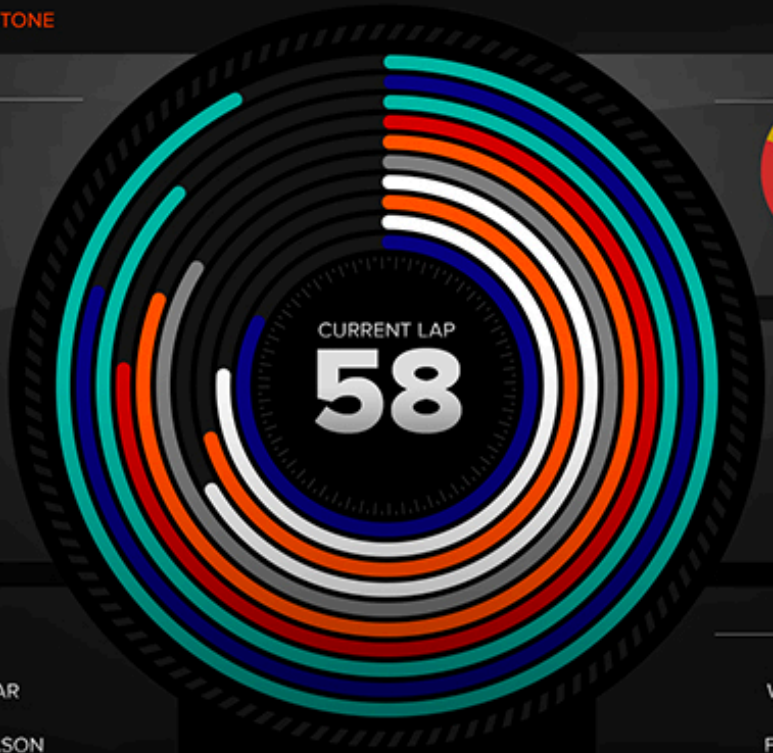
OFFICIAL TEAM PARTNER

2016 FORMULA 1 BRITISH GRAND PRIX // SILVERSTONE

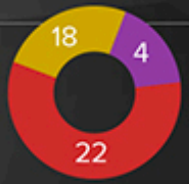
TRACK TEMP: 29°C AIR TEMP: 20°C

PIT STOP TIMES

ROS	FASTEST
VET	+0.51
BUT	+0.74
BOT	+0.78
RIC	+0.90
HUL	+1.12
HAM	+1.19
MAS	+1.34
ALO	+1.95
RAI	+2.40



TYRE COMPOUNDS



SETS USED

- SUPER SOFT
- ULTRA SOFT
- SOFT

ROS 1:40.418 **FASTEST LAPS:** 10/67

TRACK RECORD: 1:39.768

SAFETY CAR LAPS

07

4 LAST YEAR

28 THIS SEASON

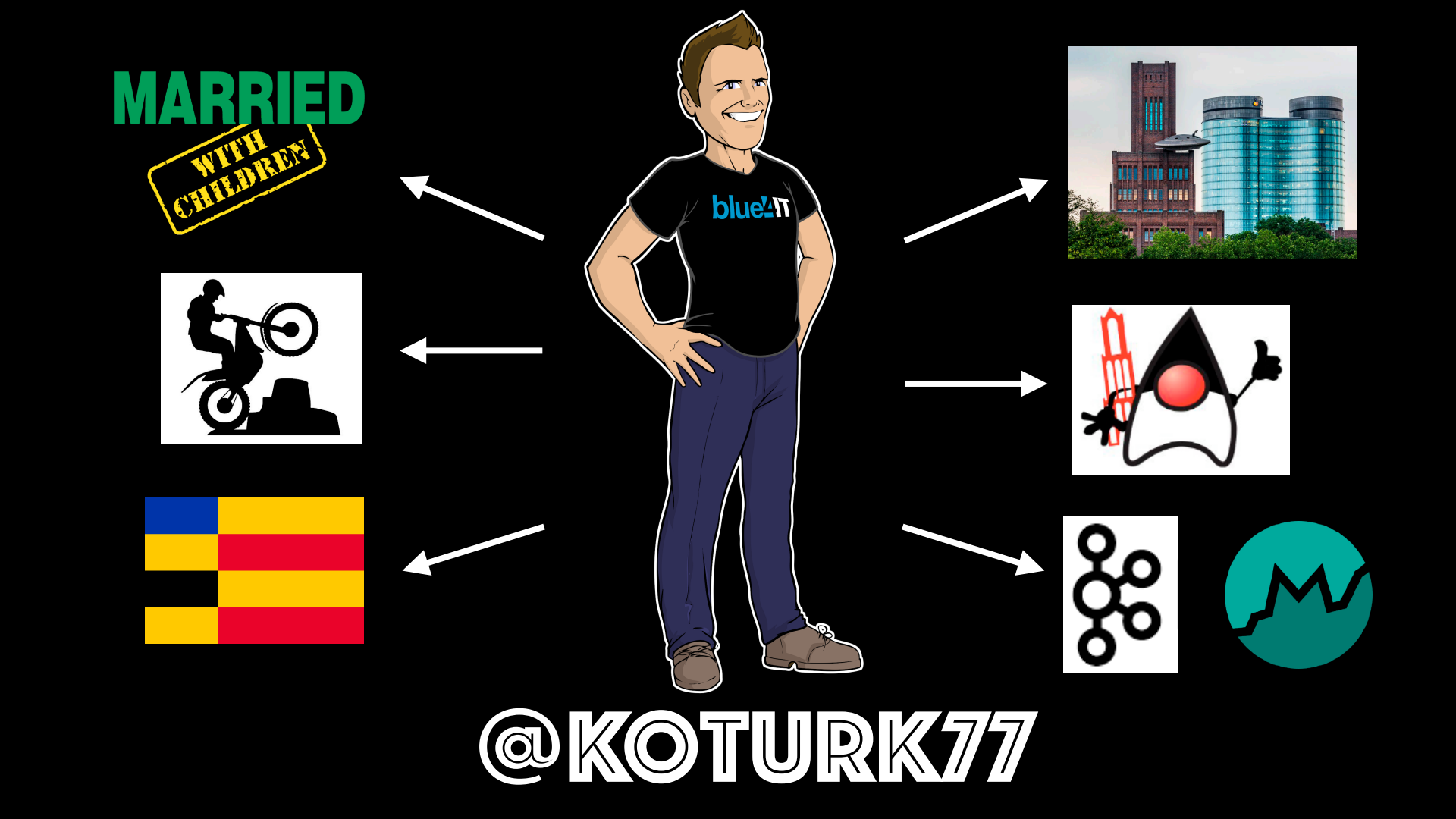
TOTAL OVERTAKES

WITH DRS 28

FOR LEAD 2

35

https://blog.purestorage.com/wp-content/uploads/2016/07/Pure-MBF1-Pitwall-v6.jpg


MARRIED
WITH CHILDREN
blue4IT
@KOTURK77


ORACLE

What do you learn today (goals)?

1. Creating Timers
   a. Doing pitstops and measure them -> 5 minutes
   b. For different drivers (tags) -> 5 minutes
   c. Getting the max seconds of the pitstops -> 5 minutes
   d. Deleting fastest laps -> 5 minutes
   e. Comparing pitstops -> 5 minutes

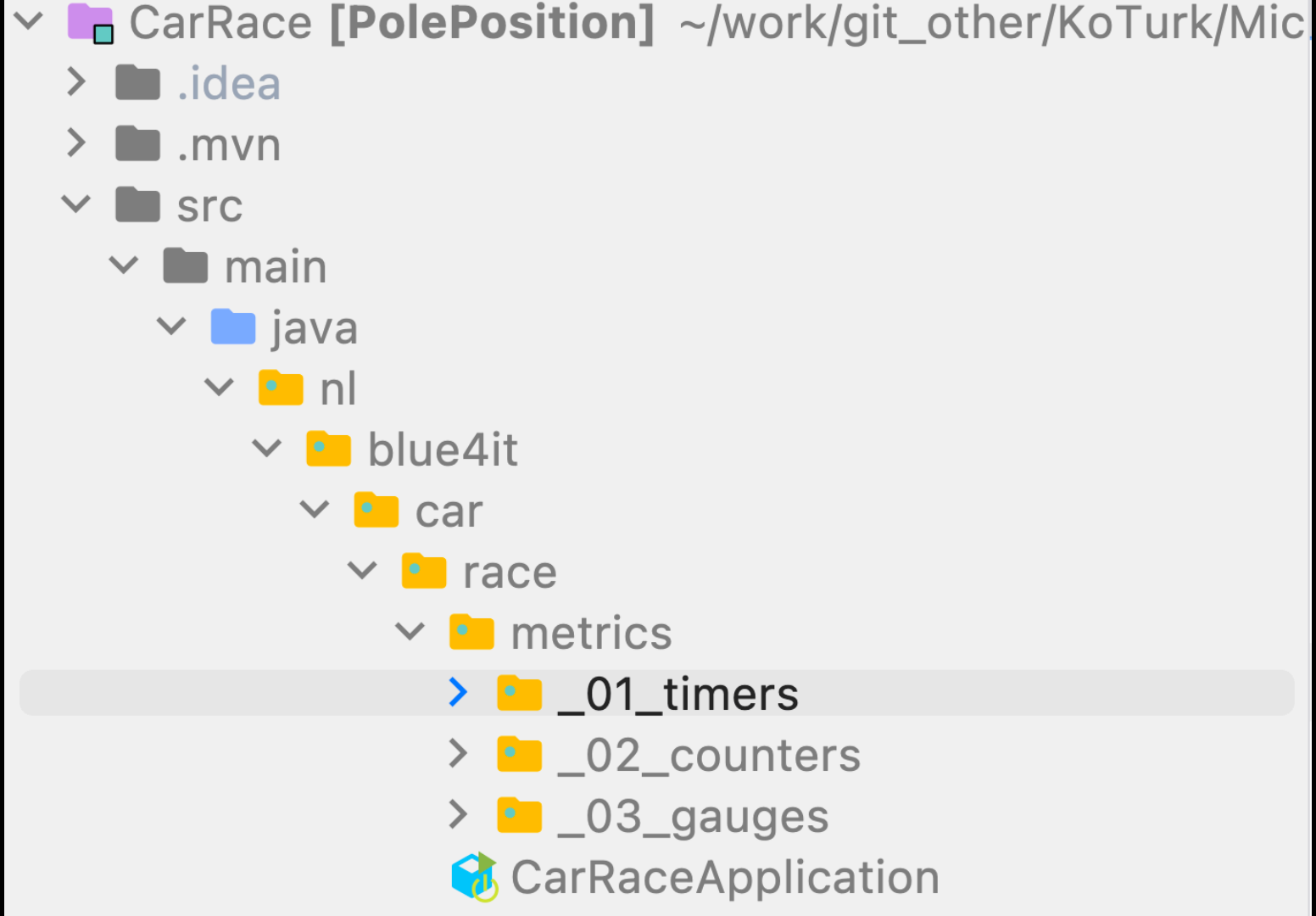
2. Creating Counters
   a. Counting the overtakes -> 5 minutes

3. Creating Gauges
   a. Builder (with tags) -> 5 minutes

If we have time……

Pushing to a datasource (Prometheus)

Creating a dashboard in Grafana


CarRace [PolePosition] ~/work/git_other/KoTurk/Mic
.idea
.mvn
src
main
java
nl
blue4it
car
race
metrics
_01_timers
_02_counters
_03_gauges
CarRaceApplication

But first

- We need to add the dependency:
  `spring-boot-starter-actuator`
- And need an application.yml like this

```
1  management.metrics.export.prometheus.enabled: true
2  spring.application.name: poleposition
3  management.endpoints.web.exposure.include: 'metrics, prometheus'
```

Next

1. Creating Timers

   a. Annotations / metrics class / builder interface

   b. Tags

   c. Registry (max / removing)

   d. Prometheus


https://s1.cdn.autoevolution.com/images/news/gallery/red-bull-racing-beats-record-to-set-new-fastest-pit-stop-in-the-history-of-f1_2.jpg

Creating a timer with an annotation

-> @Timed

-> Give it a value

Exercise

```
// TODO add the timer annotation and give it a name.
```

But sometimes you want to measure some task / calls

-> Metrics.timer

-> record method with a Duration in it

Exercise

```
// TODO create timer from Metrics class,
//  give it a name, and only measure the front seconds
```

But sometimes you want to measure some task / calls

```
-> Timer.builder

-> register it to Metrics.globalRegistry

-> Record it


Exercise
// TODO create timer from Timer interface,
// give it a name, and only record the penalty seconds

// or create timer in taggedTimer (less memory needed)
```

Image from:
https://img.bleacherreport.net

Creating a longtasksimer with an annotation

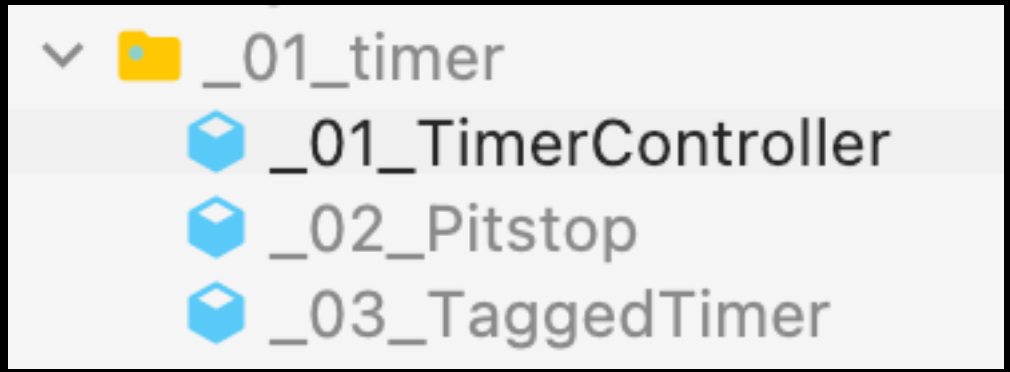
```
-> @Timed(value = “an example”)

-> longtask to true
```

Exercise

```
// TODO create a longtasktimer and give it a name
```

5 minute exercise

- Check the TODO and code!



Next

1. Creating Timers

   a. Annotations / metrics class / builder interface

   b. Tags

   c. Registry (max / removing)

   d. Prometheus

https://www.formula1.com/content/dam/fom-website/manual/Misc/PORGPManualAdds/GettyImages-1306846855.jpg.transform/9col/image.jpg

Creating tags with an annotation

```
-> @Timed(value = “an example”)

—> now add extraTags
```

```
Exercise
// TODO create timer annotation with tags:
//  tag id driver, value verstappen
//  tag id changed, value tires
//  tag id changed, value front
```

Creating tags with a builder

```
-> Metrics.timer(“an example",
                 “mytag", "value").record(pitstop);


-> add your own tag


Exercise
// TODO Create Timer with Metrics class, tag driver
and LeClerq as value
```

Now with builder

```
-> Timer.builder(name)
        .tags(tagName, tagValue)
        .register(Metrics.globalRegistry);
```

-> add tags method to it with name and value

Exercise

```
// TODO create timer with builder
```

5 minute exercise

- Check the TODO and code!

Next

1. Creating Timers

   a. Annotations / metrics class / builder interface

   b. Tags

   c. Registry (max / removing)

   d. Prometheus

Image from:
https://d2n9h2wits23hf.cloudfront.net

Getting information from the registry

```
-> return ((Timer) Metrics.globalRegistry.getMeters()
            .stream()
              .filter(meter -> meter.getId().getName().equals(timerName))
              .findFirst()
              .get())
```

-> use the max from Timer

Exercise

```
// TODO get the max pitstop time
// with max from the Timer
```

Image from:
https://sportsbase.io

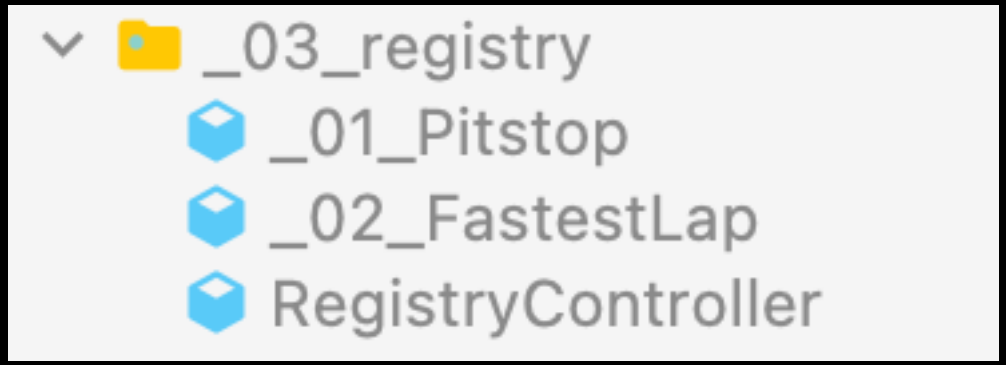
Deleting information from the registry

```
-> With Metrics.globalRegistry

-> Remove
```

```
Exercise
// TODO remove the fastest lap from the registry
//  for a driver because crossing a white line
//  use a method from Metrics.globalRegistry
```

5 minute exercise

- Check the TODO and code!



Next

1. Creating Timers

   a. Annotations / metrics class / builder interface

   b. Tags

   c. Registry (max / removing)

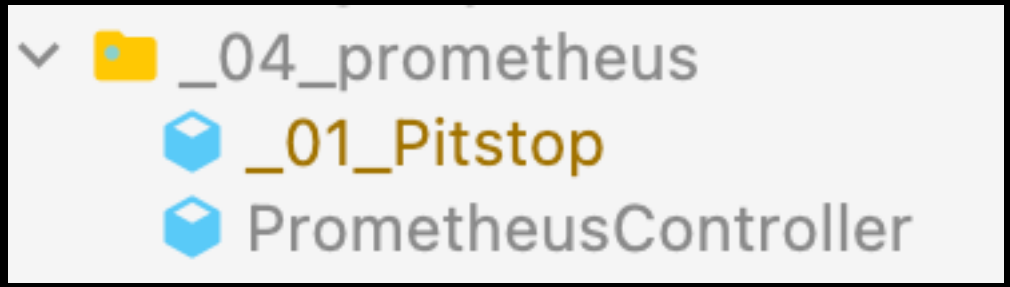
   d. Prometheus

Comparing with Prometheus

```
-> Timer.builder(“an example")
       .sla(…)
       .minimumExpectedValue(….)
       .maximumExpectedValue(…)
       .register(prometheusMeterRegistry)
       .record(ThreadLocalRandom.current().nextLong(3,20), TimeUnit.SECONDS);

-> After coding, call the endpoint
http://localhost:8080/car/race/pitstop/sla


Exercise
// TODO create a Timer
//  sla of 8 seconds
//  minimum expected of 4 seconds
//  maximum expected 7 seconds
//  register it to prometheus
//  record the randomSeconds with a unit of seconds
```

5 minute exercise

- Check the TODO and code!



(2/3) What do you learn today (goals)?

2. Creating Counters
   a. Counting the overtakes

Image from:
https://pbs.twimg.com

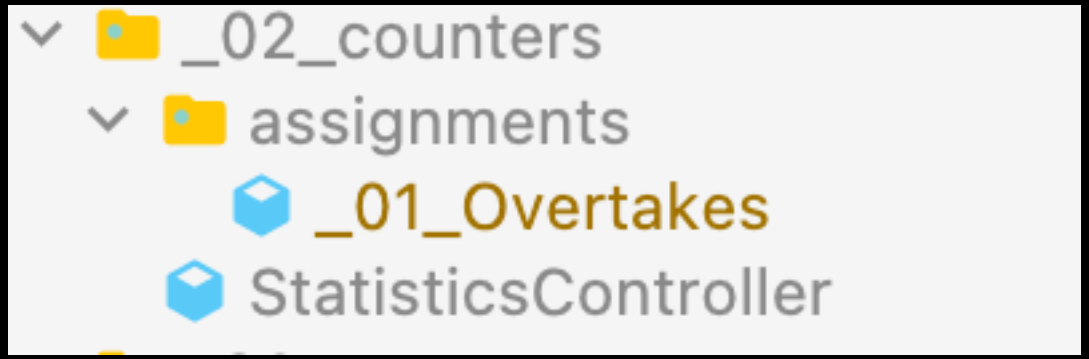
Returning the overtakes

```
-> Metrics.counter
-> increment
-> and return the count
```

```
Exercise
// TODO Create a counter and add one up
```

5 minute exercise

- Check the TODO and code!



(3/3) What do you learn today (goals)?

3. Creating Gauges
    a. Builder (with tags)

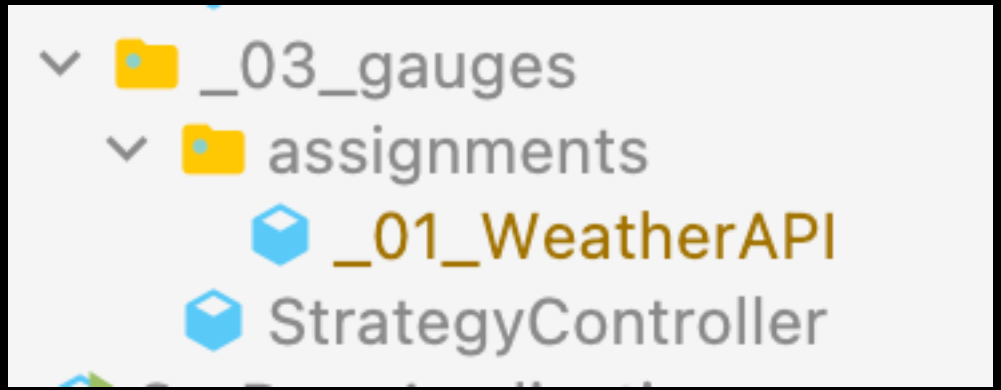

Image from:
https://www.formula1.com

Forecasting the weather

```
-> uncomment

-> Gauge.builder(“assignment_3.1”, <value_here>)
        .tags("condition", <weather_here>)
        .strongReference(true)
        .register(Metrics.globalRegistry);


Exercise
// TODO create a Gauge, with value randomHardness
//  and tag condition and put weather in it
//  and register it
```

5 minute exercise

- Check the TODO and code!



Pushing to a datasource

Goto https://prometheus.io/download/

prometheus

The Prometheus monitoring system and time series database. prometheus/prometheus

2.35.0 / 2022-04-21 Release notes

File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.35.0.darwin-amd64.tar.gz	darwin	amd64	77.01 MiB	a4f4d0ea38addc179bd37a0282738a8b03a6e58ae3706f09be3178e883c44abc
prometheus-2.35.0.linux-amd64.tar.gz	linux	amd64	76.89 MiB	e4546960688d1c85530ec3a93e109d15b540f3251e1f4736d0d9735e1e857faf
prometheus-2.35.0.windows-amd64.zip	windows	amd64	78.41 MiB	782323b31ef8f99159ec7a9042b3d553c4e49c39af8e3afe3f85e6711dad0018

Running on http://localhost:9090/

RACE ENGINEER DASHBOARD

Pilot

#88 No Driver#88

McLaren

Position

11

-6

Gap Leader

Δ Pilot Ahead

Δ Pilot Above

Pénalties

0

Les pénas purgeables ne sont pas comptabilisées

Last Lap: 01:06.238

Best Lap: 01:06.238

MAX SPEED

286 KM/H

ENGINE DATA

5 6 7 8

1

(0.00)

bar 75 125

BREAK

THROTTLE

% 25 50 75 100

OVERTAKE

0 0%

% 25 50 75 100

TYRES DATA

PIRELLI P ZERO N/A Tours

0 psi | 0 psi

0 psi | 0 psi

Télémetry Public ? Non

GRAND PRIX

GP: AUSTRIA

TOURS: 2 / 99

Track length: 4323 m

TOURS RESTANTS: 98 / 99

SESSION: SHORT QUALIFYING

WEATHER:

T° AIR: 17°C T° PISTE : 25°C

SIMPLIFIED DASHBOARD

+/- Pos	#	Pilots	FI	Best Lap	Gap	S1 Last	S1 Best	S2 Last	S2 Best	S3 Last	S3 Best
1	#14	No Driver#14		01:03.937	-	15.963	–	29.823	–	19.149	–
2	#15	No Driver#15		01:04.220	+0.283	15.964	–	29.932	–	19.323	–
3	#21	No Driver#21		01:04.345	+0.408	15.777	–	29.168	–	19.400	–
4	#69	No Driver#69		01:04.707	+0.770	16.222	–	29.176	–	19.307	–
5	#9	No Driver#9		01:04.719	+0.782	15.963	–	29.1	–	19.587	–
6	#30	No Driver#30		01:05.199	+1.262	16.435	–	29.114	–	19.649	–
7	#22	No Driver#22		01:05.211	+1.274	16.206	17.320	29.244	–	19.759	–
8	#24	No Driver#24		01:05.261	+1.324	16.127	–	29.657	–	19.476	
9	#22	No Driver#22	PIT	01:05.579	+1.642	16.439	–	29.291	–	19.847	–
10	#13	No Driver#13		01:06.065	+2.128	16.374	–	29.525	–	20.165	–
11	#88	No Driver#88		01:06.238	+2.301	16.333	–	29.819	–	20.085	–
12	#13	No Driver#13	AI	--:--.---	-	00.000	–	0	–	00.000	–
13	#13	No Driver#13		--:--.---	-	16.530	16.530	0	–	00.000	–
14	#14	No Driver#14		--:--.---	-	00.000	–	0	–	00.000	–
15	#15	No Driver#15	AI	--:--.---	-	00.000	–	0	–	00.000	–
16	#24	No Driver#24	AI	--:--.---	-	00.000	–	0	–	00.000	–
17	#25	No Driver#25	AI	--:--.---	-	00.000	–	0	–	00.000	–
18	#32	No Driver#32	AI	--:--.---	-	00.000	–	0	–	00.000	–
19	#45	No Driver#45	AI	--:--.---	-	00.000	–		–	00.000	–
20	#48	No Driver#48	AI	--:--.---	-	00.000	–	0	–	00.000	–

Image from:
https://www.racedepartment.com/

Goto https://grafana.com/grafana/download

Standalone MacOS/Darwin Binaries (64 Bit) SHA256: 32c75b0c0535317b14de0da0bebf2cf1d0de29206c19d316c5266c9c6082fbeb

Read the MacOS installation guide for more information.

```
curl -O https://dl.grafana.com/enterprise/release/grafana-enterprise-8.5.3.darwin-amd64.tar.gz
tar -zxvf grafana-enterprise-8.5.3.darwin-amd64.tar.gz
```

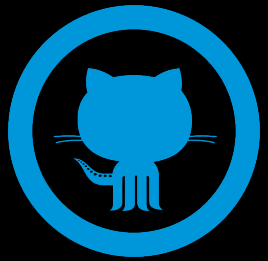
Running on http://localhost:3000/


Image from:
https://www.bleachersnews.com
33
HONDA
RedBull
Esso
citrix
RAUCH
YES

Image from:
https://mediacloud.theweek.co.uk

Thank you for listening

blue4IT



All examples you can find at:
https://github.com/KoTurk/Micrometer/tree/master/WorkshopFest/CarRace

@KoTurk77