
# CompletableFuture: Advanced

Alexey Zinovyev, Java/BigData Trainer in EPAM

With IT since 2007

With Java since 2009

With Hadoop since 2012

With EPAM since 2015

## About

# Contacts

E-mail : Alexey_Zinovyev@epam.com

Twitter : @zaleslaw @BigDataRussia

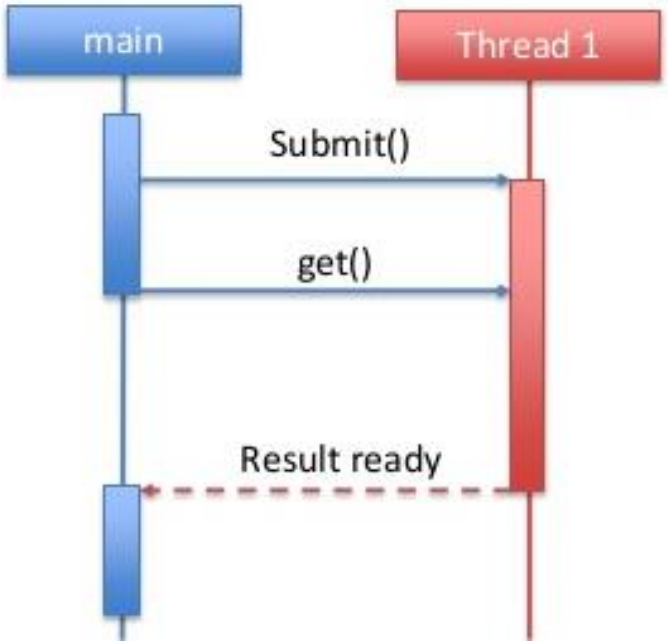
Facebook: https://www.facebook.com/zaleslaw

vk.com/big_data_russia **Big Data Russia**

vk.com/java_jvm **Java & JVM langs**

# ASYNC MOTIVATION

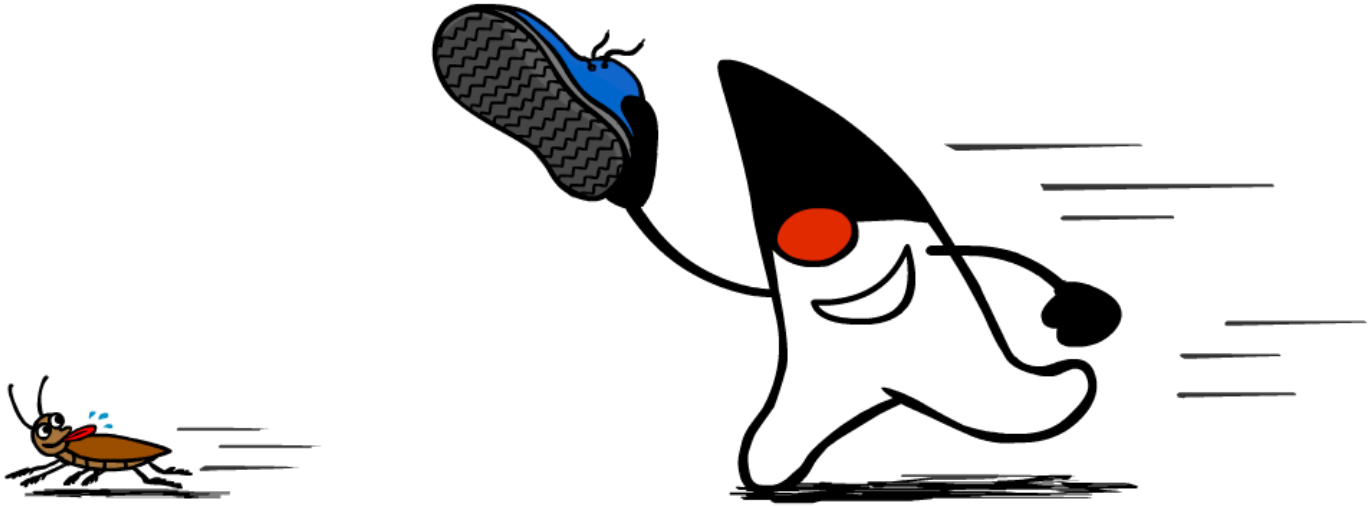
# Async with Future



# Error handling

```
try {
   try {
      return parse(fileObject, result.get());
   } catch (ExecutionException e) {
      log.error(e);
   }
} catch (IOException e) {
   log.error(e, fileObject);
} catch (InterruptedException e) {
   throw new RuntimeException (e);
}
```

# Future



# CF CLASS

# CompletableFuture as is …

```
public <U,V> CompletableFuture<V> thenCombineAsync(
    CompletableFuture<? extends U> other,
    BiFunction<? super T, ? super U, ? extends V> fn,
    Executor executor)
```

# Main parts of CF

- Method Chaining
- No blocking calls
- Combination of a few CFs
- CompletionStage is not a CF

# Create methods

- CompletableFuture()
- completedFuture()
- supplyAsync()
- runAsync()

# Transformation methods [map(λ(x))]

- thenApply()
- thenApplyAsync()

# Subscription methods [subscribeOn()]

- thenAccept()
- thenAcceptAsync()
- thenRun()
- thenRunAsync()

# Exception handling

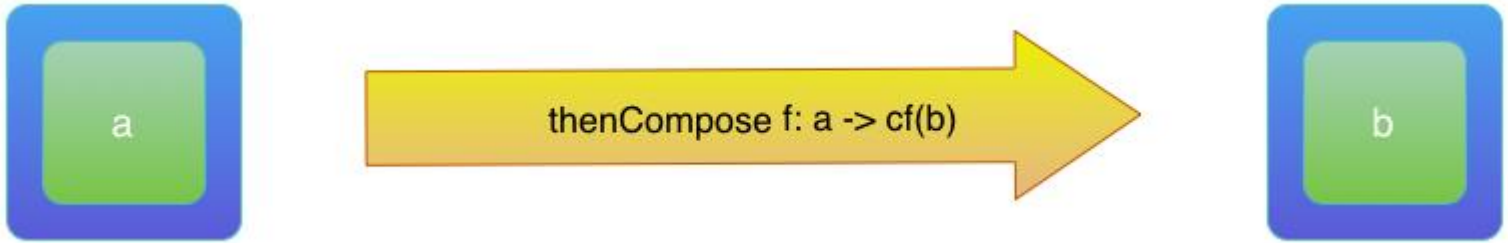
- exceptionally()
- handle()

# Combination method [reduce(λ(x))]

- thenCombine()
- allOf()

# Composition method [flatMap(λ(x))]

- thenCompose()



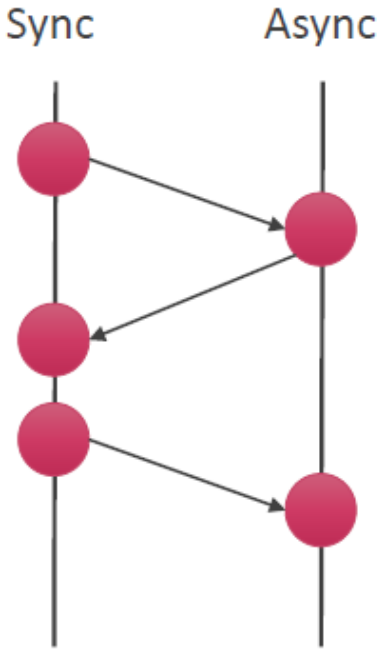
# Write methods

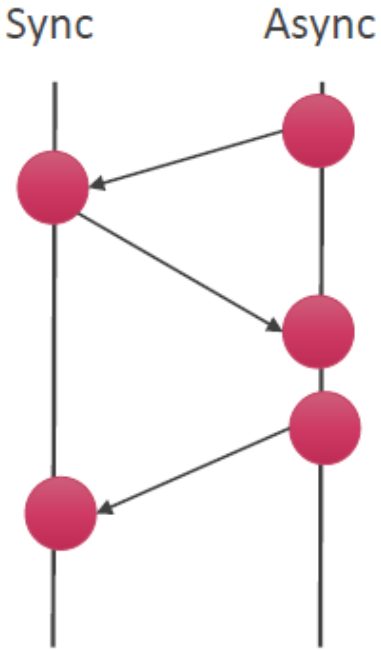
- boolean complete(T value)
- boolean completeExceptionally(Throwable ex)
- boolean cancel(boolean mayInterruptIfRunning)
- void obtrudeValue(T value)
- void obtrudeException(Throwable ex)

# Read methods

- boolean isDone()
- T get()
- T getNow(T valueIfAbsent)
- T join()
- int getNumberOfDependents()

# Async with CF



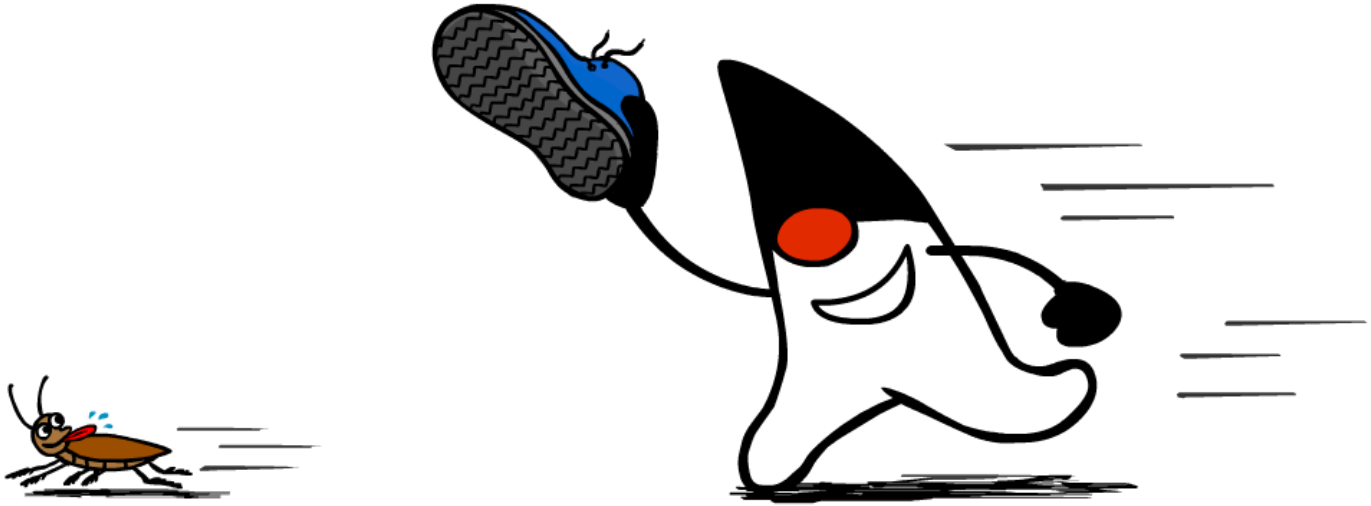



# Pipelines

```
CompletableFuture.completedFuture("str")
    .thenApplyAsync(s->s+"1")
    .thenApply(s->s+"2")
    .thenAccept(System.out::println)
    .thenRunAsync(()->{System.out.println("end");});
```

```
CompletableFuture.supplyAsync(()->"srt")
    .thenApply(s->s+"1")
    .thenApply(s->s+"2")
    .thenAcceptAsync(System.out::println)
    .thenRun(()->{System.out.println("end");});
```

# Go to the stars!



# Contacts

E-mail : Alexey_Zinovyev@epam.com

Twitter : @zaleslaw @BigDataRussia

Facebook: https://www.facebook.com/zaleslaw

vk.com/big_data_russia **Big Data Russia**

vk.com/java_jvm **Java & JVM langs**

# Any questions?