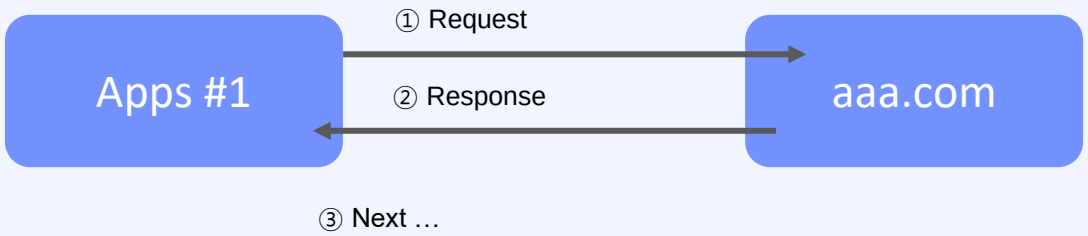
Webflux

3 sync/async & block/non-block

sync 용어

Sync

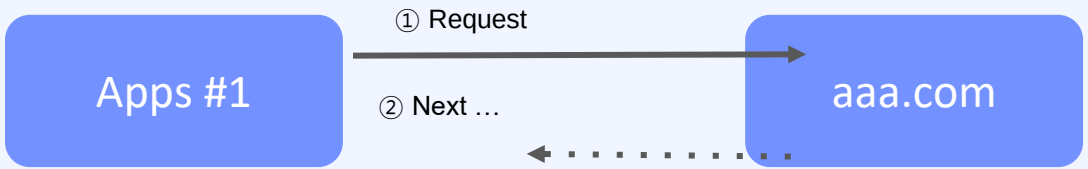


sync 예시

```java
public class SynchronousExample {
    public static void main(String[] args) throws IOException, InterruptedException {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
                .uri(URI.create("https://www.naver.com"))
                .build();

        HttpResponse<String> response =
                client.send(request, HttpResponse.BodyHandlers.ofString());

        System.out.println(response.body());
    }
}
```

async 용어

ASync



async 예시

```java
public class AsynchronousExample {
    public static void main(String[] args) {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
                .uri(URI.create("https://www.naver.com"))
                .build();

        CompletableFuture<HttpResponse<String>> future =
                client.sendAsync(request, HttpResponse.BodyHandlers.ofString());

        System.out.println("This is Next...");

        future.thenApply(HttpResponse::body) CompletableFuture<String>
                .thenAccept(System.out::println) CompletableFuture<Void>
                .join();
    }
}
```

async 예시

```java
public class AsynchronousExample {
    public static void main(String[] args) {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
                .uri(URI.create("https://www.naver.com"))
                .build();

        CompletableFuture<HttpResponse<String>> future =
                client.sendAsync(request, HttpResponse.BodyHandlers.ofString());

        System.out.println("This is Next...");

        future.thenApply(HttpResponse::body) CompletableFuture<String>
                .thenAccept(System.out::println) CompletableFuture<Void>
                .join();
    }
}
```

async 예시

```java
public class AsynchronousExample {
    public static void main(String[] args) {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
                .uri(URI.create("https://www.naver.com"))
                .build();

        CompletableFuture<HttpResponse<String>> future =
                client.sendAsync(request, HttpResponse.BodyH

        System.out.println("This is Next...");

        future.thenApply(HttpResponse::body) CompletableFuture<String>
                .thenAccept(System.out::println) CompletableFuture<Void>
                .join();
    }
}
```

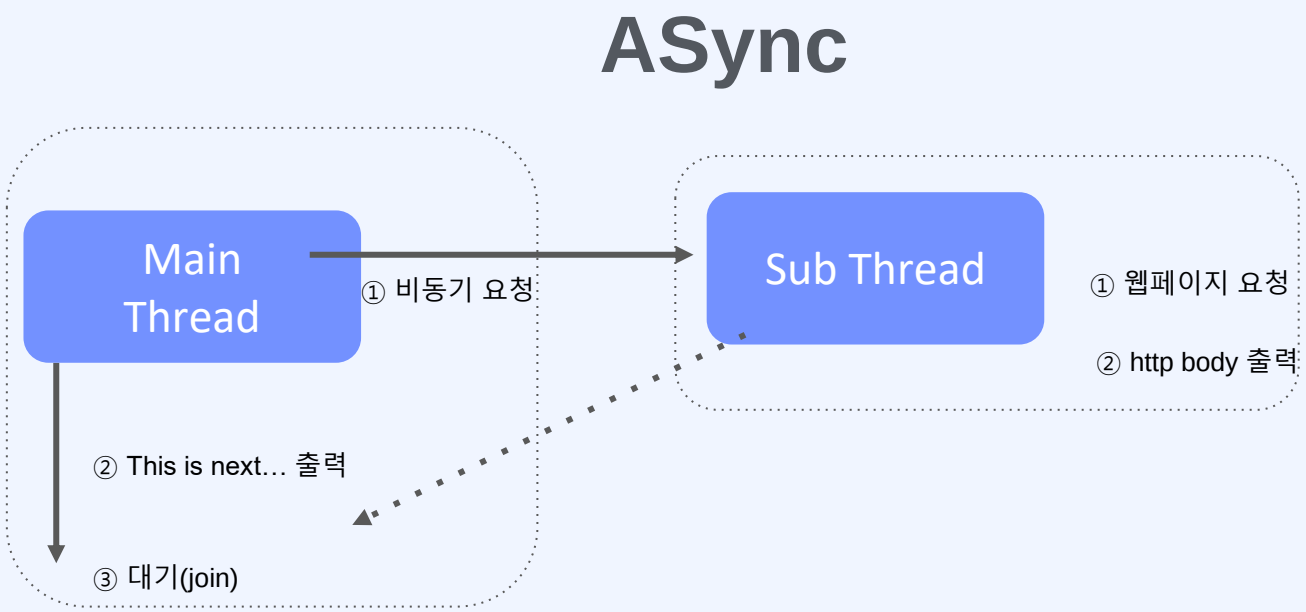
```
This is Next...
<!doctype html> <html lang="ko" dat
<title>NAVER</title> <meta name="a
```

async 예시

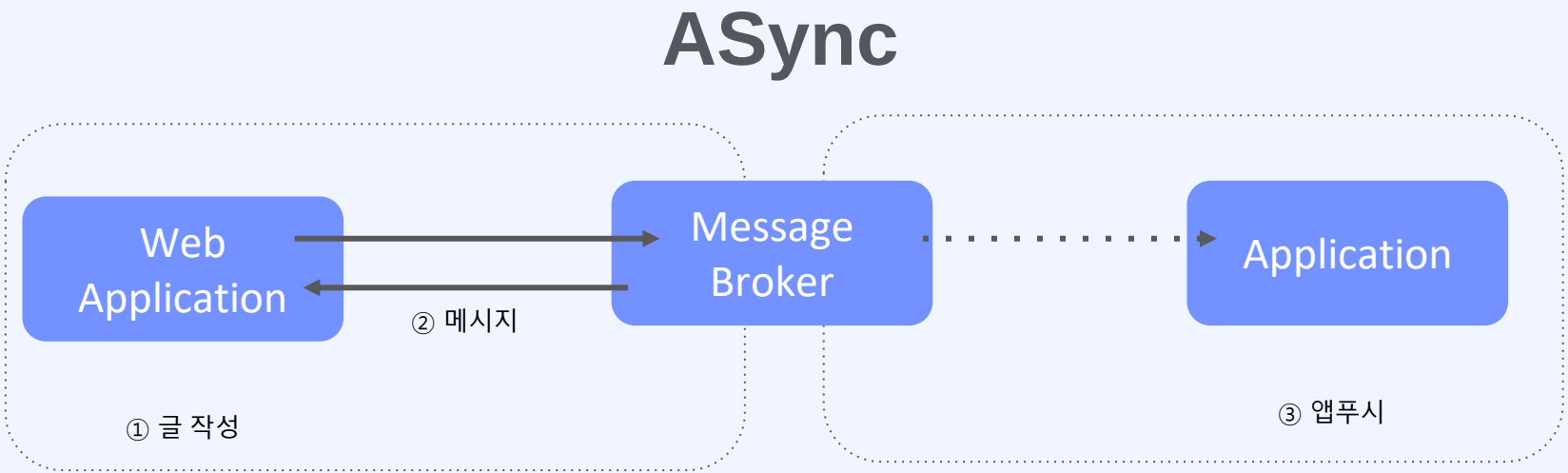
```java
public class AsynchronousExample {
    public static void main(String[] args) {
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request = HttpRequest.newBuilder()
                .uri(URI.create("https://www.naver.com"))
                .build();

        CompletableFuture<HttpResponse<String>> fu
                client.sendAsync(request, HttpResp

        System.out.println("This is Next...");

        future.thenApply(HttpResponse::body) CompletableFuture<String>
                .thenAccept(System.out::println) CompletableFuture<Void>
                .join();
    }
}
```

```
[main] This is Next...
[ForkJoinPool.commonPool-worker-1]    <!doctype htt
name="viewport" content="width=1190"> <title>NAVE
```

async 예시



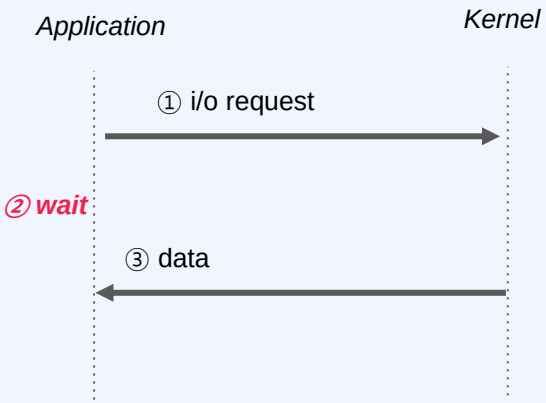
async 예시



Block / Non-Block

blocking 용어

Blocking



blocking 예시

```java
public static void main(String[] args) {
    int port = 8080;
    try (ServerSocket serverSocket = new ServerSocket(port)) {
        System.out.println("Started server and port " + port);

        while (true) {
            Socket clientSocket = serverSocket.accept();
            System.out.println("Connected Client: " + clientSocket.getInetAddress());

            BufferedReader reader = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
            String inputLine = reader.readLine();
            System.out.println("Received message: " + inputLine);

            clientSocket.close();
            System.out.println("Closed Client: " + clientSocket.getInetAddress());
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

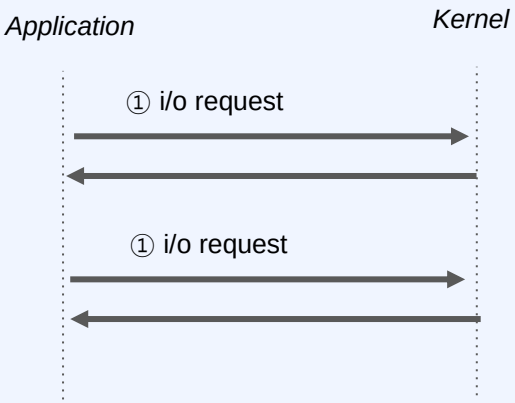
blocking 예시

```java
public static void main(String[] args) {
    int port = 8080;
    try (ServerSocket serverSocket = new ServerSocket(port)) {
        System.out.println("Started server and port " + port);

        while (true) {
            Socket clientSocket = serverSocket.accept();
            System.out.println("Connected Client: " + clientSocket.getInetAddr

            BufferedReader reader = new BufferedReader(new InputStreamReader(
            String inputLine = reader.readLine();
            System.out.println("Received message: " + inputLine);

            clientSocket.close();
            System.out.println("Closed Client: " + clientSocket.getInetAddress());
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
Started server and port 8080
Connected Client: /127.0.0.1
Received message: hello blocking
Closed Client: /127.0.0.1
```

non-blocking 용어

Non Blocking



non-blocking 예시

```java
public static void main(String[] args) {
    int port = 8090;
    try (ServerSocketChannel serverSocketChannel = ServerSocketChannel.open()) {
        serverSocketChannel.bind(new InetSocketAddress(port));
        serverSocketChannel.configureBlocking(false);
        System.out.println("Started server and port " + port);

        while (true) {
            System.out.println(LocalTime.now());

            SocketChannel clientChannel = serverSocketChannel.accept();
            if (clientChannel == null) {
                Thread.sleep( millis: 500);
                continue;
            }

            System.out.println("Connected Client: " + clientChannel.getRemoteAddress());

            ByteBuffer buffer = ByteBuffer.allocate( capacity: 1024);
            int bytesRead = clientChannel.read(buffer);

            if (bytesRead != -1) {
                buffer.flip();
```

non-blocking 예시

```java
public static void main(String[] args) {
    int port = 8090;
    try (ServerSocketChannel serverSocketChannel = ServerSocketChannel.open()) {
        serverSocketChannel.bind(new InetSocketAddress(port));
        serverSocketChannel.configureBlocking(false);
        System.out.println("Started server and port " + port);

        while (true) {
            System.out.println(LocalTime.now());

            SocketChannel clientChannel = serverSocketChannel.accept();
            if (clientChannel == null) {
                Thread.sleep( millis: 500);
                continue;
            }

            System.out.println("Connected Client: " + clientChannel.getRemoteAddress());

            ByteBuffer buffer = ByteBuffer.allocate( capacity: 1024);
            int bytesRead = clientChannel.read(buffer);

            if (bytesRead != -1) {
                buffer.flip();
```

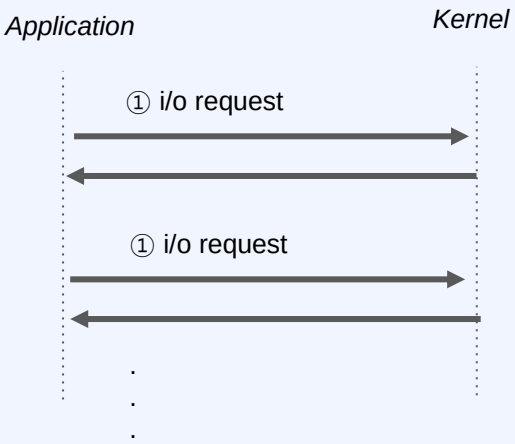
non-blocking 예시

```java
public static void main(String[] args) {
    int port = 8090;
    try (ServerSocketChannel serverSocketChannel = ServerSocketChannel.open()) {
        serverSocketChannel.bind(new InetSocketAddress(port));
        serverSocketChannel.configureBlocking(false);
        System.out.println("Started server and port " + port);

        while (true) {
            System.out.println(LocalTime.now());

            SocketChannel clientChannel = serverSocketChannel.accept();
            if (clientChannel == null) {
                Thread.sleep( millis: 500);
                continue;
            }

            System.out.println("Connected Client: " + clientChannel.getRemoteAddress());

            ByteBuffer buffer = ByteBuffer.allocate( capacity: 1024);
            int bytesRead = clientChannel.read(buffer);

            if (bytesRead != -1) {
                buffer.flip();
```

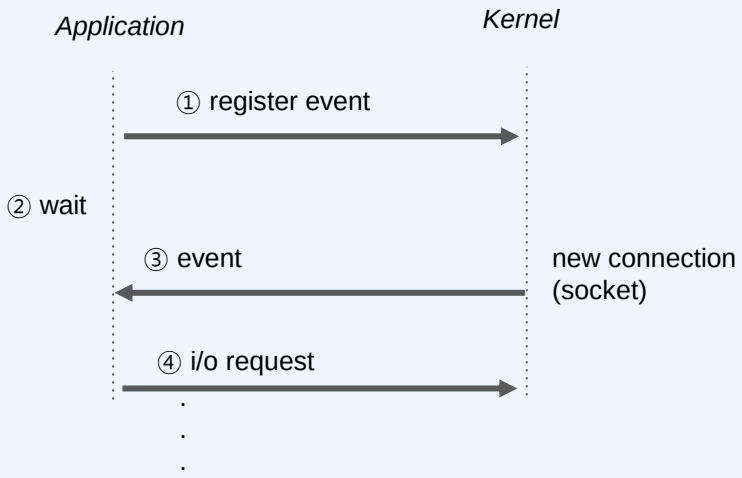
```
Started server and port 8090
01:59:17.593760
01:59:18.099179
01:59:18.604687
01:59:19.110343
01:59:19.613209
01:59:20.117399
01:59:20.621428
01:59:21.123505
01:59:21.626205
```

i/o multiplexing 용어

Non Blocking



I/O Multiplexing

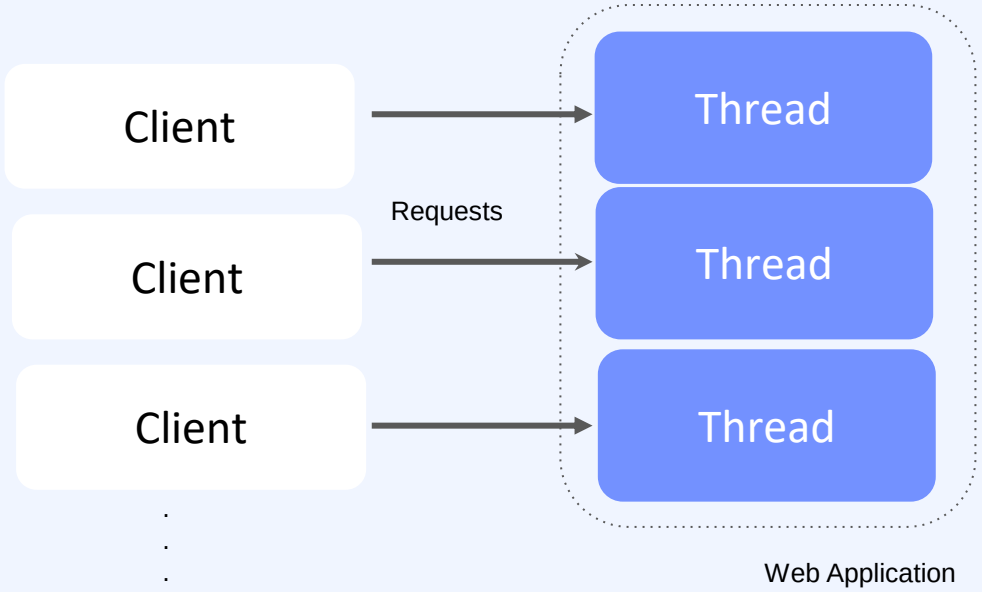


i/o multiplexing

용어

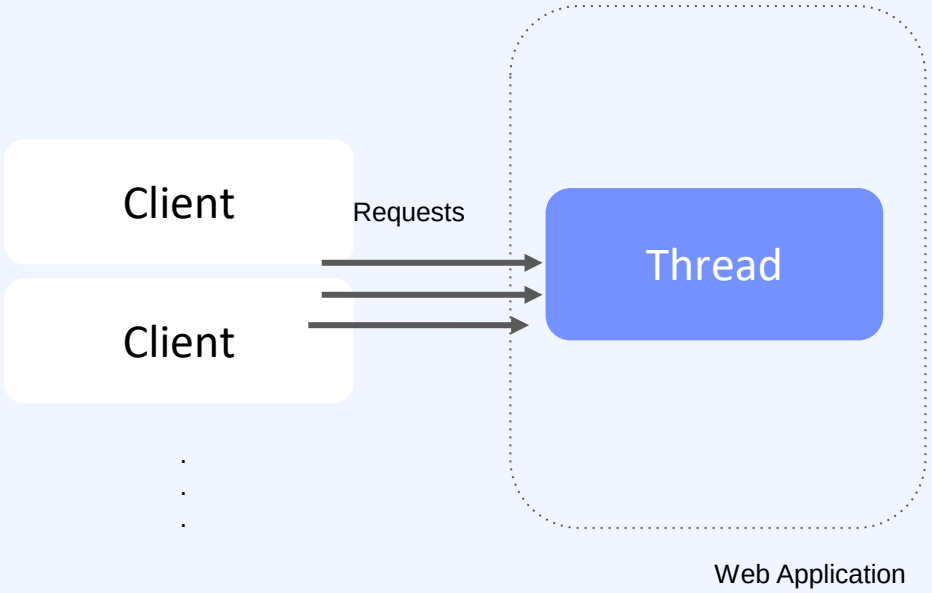
```
public static void main(String[] args) {
    try {
        Selector selector = Selector.open();

        ServerSocketChannel serverChannel = ServerSocketChannel.open();
        serverChannel.bind(new InetSocketAddress(PORT));
        serverChannel.configureBlocking(false);

        serverChannel.register(selector, SelectionKey.OP_ACCEPT);
        System.out.println("Started server and port " + PORT);

        while (true) {
            selector.select();

            Set<SelectionKey> selectedKeys = selector.selectedKeys();
            for (SelectionKey key : selectedKeys) {
                if (key.isAcceptable()) {
                    ServerSocketChannel serverSocketChannel = (ServerSocketChannel) key.channel();
                    SocketChannel clientChannel = serverSocketChannel.accept();
                    clientChannel.configureBlocking(false);
                    clientChannel.register(selector, SelectionKey.OP_READ);
                    System.out.println("Accepted new connection " + clientChannel.getRemoteAddress());
                } else if (key.isReadable()) {
                    SocketChannel clientChannel = (SocketChannel) key.channel();
                    ByteBuffer buffer = ByteBuffer.allocate(BUFFER_SIZE);
                    int bytesRead = clientChannel.read(buffer);
```

blocking 예시



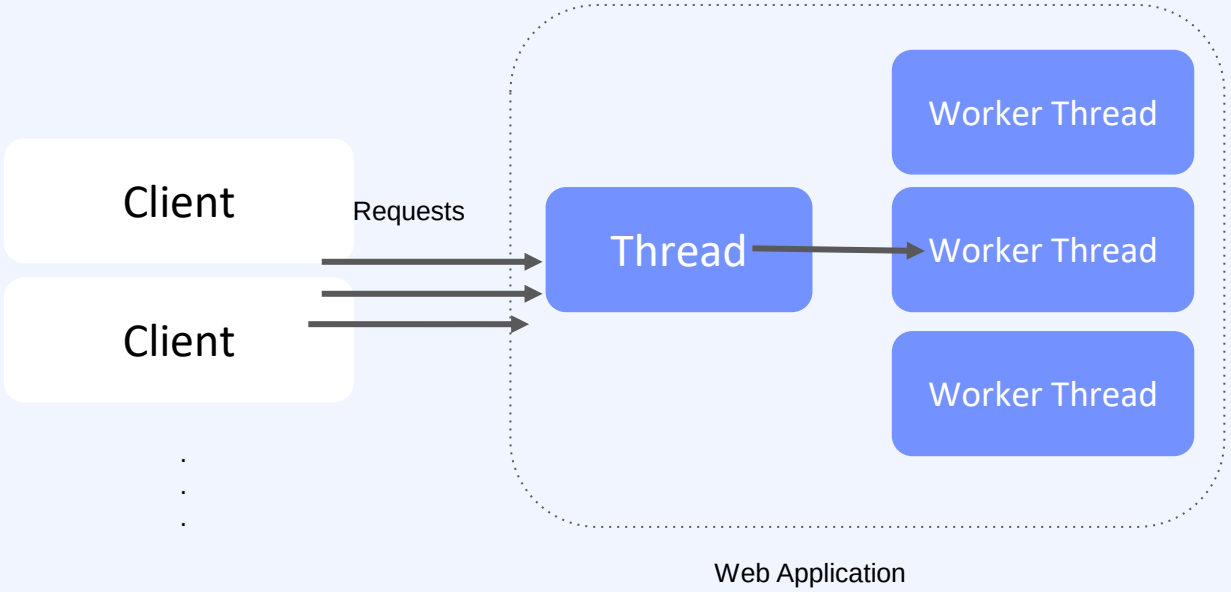
Thread per Request

i/o multiplexing 예시



I/O Multiplexing
(Non blocking)

i/o multiplexing 예시



I/O Multiplexing (Non blocking)

마무리

1. sync/async
2. block/non-block
3. i/o multiplexing