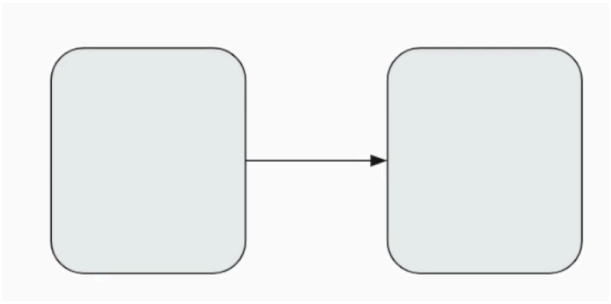
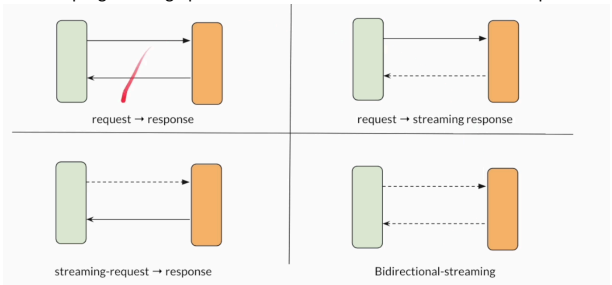
Communication Patterns

Sunday 26 January 2025 16:27



- We all have been typing code in the request and response format
- Reactive programming opens the door for 3 additional communication patterns



- The first one we know and is simple, we send a request and we get a response back.
- If our requirement is simple and we just send a request and get a response back, then we can make use of Virtual Threads (a separate course) for improvement
- The 2nd one (top-right),
  - What is streaming response? - You send a request and you get multiple response.
  - How it will look like? For e.g. Lets consider you are trying to order a pizza. So you send a e request i.e. you submit a order for pizza. Now pizza website will be sending some kind of streaming updates like 'Your pizza is being prepared', 'Your pizza is prepared', 'Now its out for delivery', 'The driver is 5 miles away', 'The driver is 2 miles away', 'Pizza delivered'. So they are giving streaming response to your mobile device.
  - So we send one request but we get streaming periodic response.
- The 3rd (bottom-left):
  - In some cases, we will be sending a streaming request to the remote server. E.g. we are wearing a apple watch and it keeps on sending the heart rate to the remote server. Or, you are typing a Google doc and when you are typing something, it gets saved to the remote server.
  - One thing to remember here is that when we say streaming request, we are not sending multiple HTTP requests. We just open 1 single connection through which we send multiple streaming messages to the remote server.
- The 4th (bottom-right): aka Bidirectional streaming
  - This is combination of the above 2
  - Here, e.g. the green and the orange applications can talk just like a human being. They can keep on exchanging messages
- So these type of communication patterns are possible which we can easily achieve using reactive programming.
- And these 3 communication patterns cannot be solved by virtual threads or structured concurrency.