Camunda BPM is a lightweight, open source platform for business process management

**Events in Camunda:**

**Message start event**: only after the message comes the process will start

**Intermediate event:** Boundary(Interrupting, Non Interrupting), Non Boundary(Throw, Catch)

**Intermediate timer Interrupting event** will interrupt the normal flow and part ways with the normal execution

**Intermediate timer Non Interrupting event** will not interrupt the normal flow and it will have its own path of execution along with the normal

**Link Event:**  It is useful to communicate between lanes

**Message Start Event:** Only after sending the message the process will start

<http://localhost:8080/engine-rest/message> (POST)

{

    "messageName":"Message\_start",

    "businessKey":"320",

    "processVariables":{

        "question":{"value" :"what is cockpit? ", "type":"String"}

    }

}

**Timer as Start Event:** Only after a certain period of time the process will start

**Signal event:** intermediate event. Could be throw or catch, interrupting or non interrupting. All the running process instances whoever has this signal all those instances will act on this signal.

<http://localhost:8080/engine-rest/signal> (POST)

{

    "name":"GoHomeAlert",

    "variables":{

        "alertType":{

            "value":"High"

        }

    }

}

**Signal interrupting:**  will follow 1 path after signal arrived

**Signal non interrupting:**  will follow both paths after signal arrived

**Signal start end event**

**Condition event:** An event-based gateway allows the process to wait for one or more events to occur before deciding which path to take.

**Escalation event:** Interrupting and non interrupting (eg file: escalationEvent.bpmn) where we have used shipment of orders as an example. If the delivery takes a long time we will send a message to the customer

**Escalation as a start event:** it will catch the escalation thrown by the intermediate escalation event. The start event cant be a escalation event

**Tasks in Camunda:**

**Types:** Service task, User task, Script task, Send task, Receive task, Business rule task

**Script task:** Inline(within the camunda bpmn file), External(external and adding it in the bpmn as external resource)

**User task, Service task**

**Business rule task:**