

Chain of Responsibility Design Pattern

Chain of responsibility pattern is used to achieve loose coupling in software design where a request from the client is passed to a chain of objects to process them. Later, object in the chain will decide themselves who will be processing the request and whether the request is required to be sent to the next object in the chain or not.

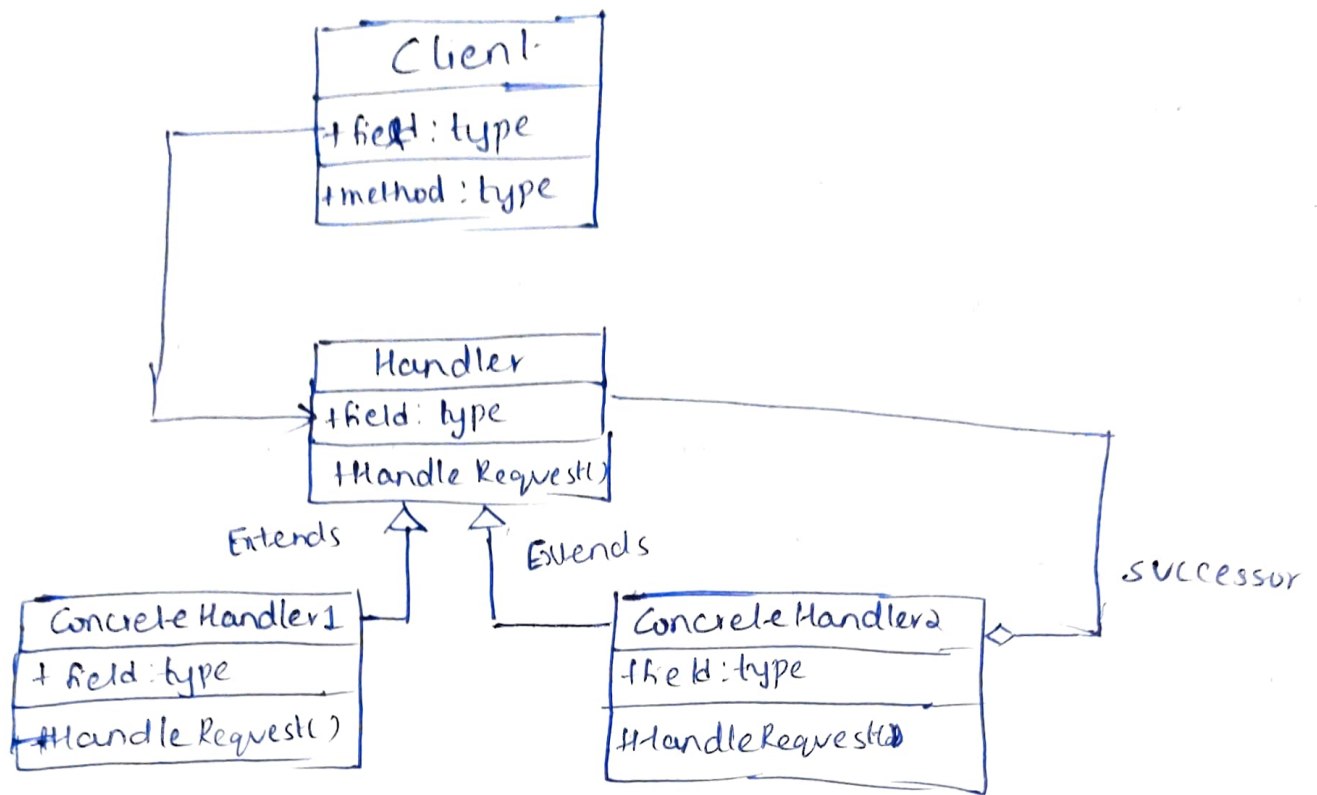
→ Where and When Chain of Responsibility pattern is applicable

When you want to decouple a request's sender and receiver.

Multiple objects, determined at runtime are candidates to handle a request.

When you don't want to specify handler's explicitly in your code. When you want to issue a request to one of several objects without specifying the receiver explicitly.

→ This pattern recommended when multiple objects can handle a request and the handler doesn't have to be specific object. Handler determined at runtime.



Handler: This interface which will primarily receive the request and dispatches request to chain of handlers.

Concrete Handlers: Actual handlers of request chained in some sequential order

Client: Originator of request and this will access handler to handle it.

→ the client in need of a request to be handled sends it to chain of handlers which are classes that extend Handler class. Each of handlers in chain takes its turn at trying to handle request it receives from client.