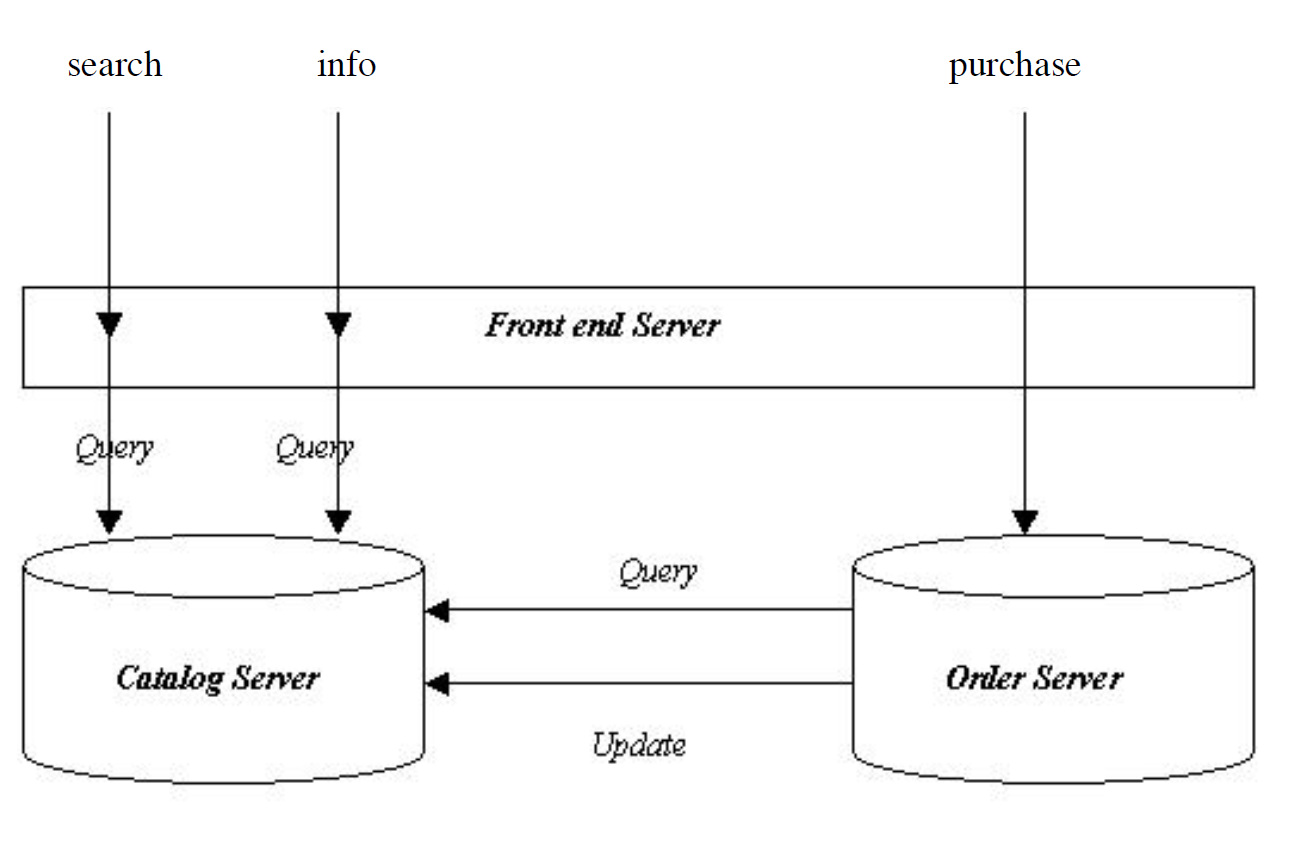
Name: محمود حسين باشا Id: 11840238

Proxy server 7/22/22

Proxy server

This is a proxy server code created using node js using express and http module to handle a request from client then send it to a micro service and send the response to the client.



This part implements the front-end server in the figure above where is receive requests from clients

And forwarding them to order and catalog server.

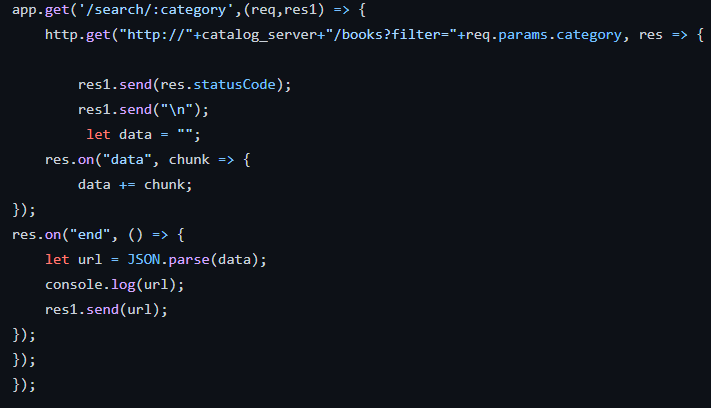
This method is for transparency, hiding the complexity of multiple servers.

This code handle three requests:

1. Search(GET): where it send request to search for a catalog
2. Info(GET): where is search for details about a book by given id
3. Purchase(post): which send request to purchase

Code explains: the code used express to handle client request where http used to send requests

Using this implements a forwarding system.



Here node receives a request at “search/category” and then send the request to catalog server then getting the JSON response from service and print the status code and the retrieved data in the client browser. Same on for the last two parts.

In the end I determine the port the client must use to send data for the proxy(front-end):



Scalability: it’s easy to scale due to the front work is to control traffic between client and servers, it also works as a load balancer.

Reliability: currently the system is not reliable but with some node the system can be reliable.

Single point of failure: there is only one front end server if it fails the whole project fail but it can be solved by redundancy.

Dependency consideration: if catalog server fails the whole system fail but of purchase fail the whole system doesn’t fail.

Data integrity: one node for each functionality so there is no need for data integrity (no data loss).

Security and privacy: no security and privacy.

The system can improve by adding user’s database and payment third party using payable or visa cards,

It also can improve by adding more features:

We can add number of books to be purchased in the api url for purchase server

We can add search for a specific detail like search for a specific genus

Where the program fail:

1. If the service send xml response since this server only handle json response.

This can be solved by writing a handler for xml of use third party handler

1. If the catalog server fails : the catalog is the main server where if it fails the whole system fails.

This program can be runed using (npm run start) which runs the server, then send the request using any browser by putting the ip and port number for the front server and then the api desired .