

Summer Internship 2016 - Hasura.io

Status of Work Done - 10 May 2016 to 24 May 2016



Web Development

Web development broadly refers to the tasks associated with developing websites (with a specified functionality) for hosting via internet or intranet.

The tasks which follow are

- Client-side coding
- Server-side coding
- Database technology

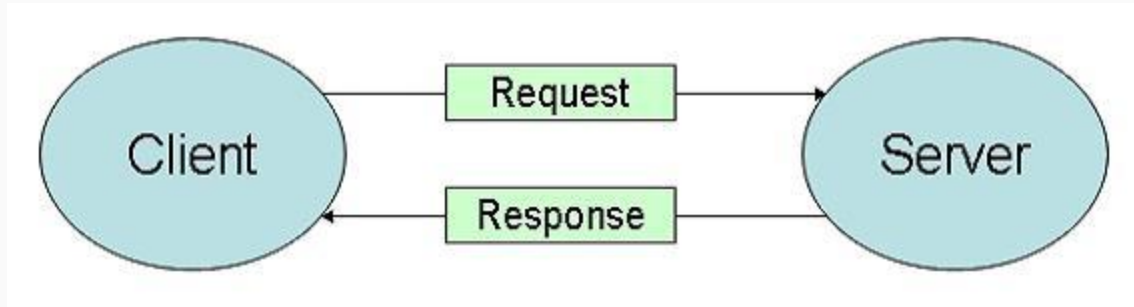
Client-Server Architecture

Client : Service / Resource Requesters.

Server : Service / Resource Providers.

The client and the server exchange messages in a request-response messaging pattern. The client sends a request and the server sends a response.

Client-Server Architecture



To understand the messages, they have to communicate in a common language. Thus the client does not have to be concerned with how the server performs while fulfilling the request and delivering the response. It just should be able to understand the response received.

LAMP Stack

LAMP is a combination of Apache Server, MySQL Database Service, and PHP as the server-side scripting language on Linux machine.

Apache also supports perl, python etc.

Worked with **forms (HTML)**, and understood PHP and MySQL Database.

Flask

Flask is a Python framework for creating web applications.

We used/understood the following while working with Flask.

1. Different functionalities at different end-points.
2. Redirect / Render templates.
3. AJAX calls
4. Session Management.

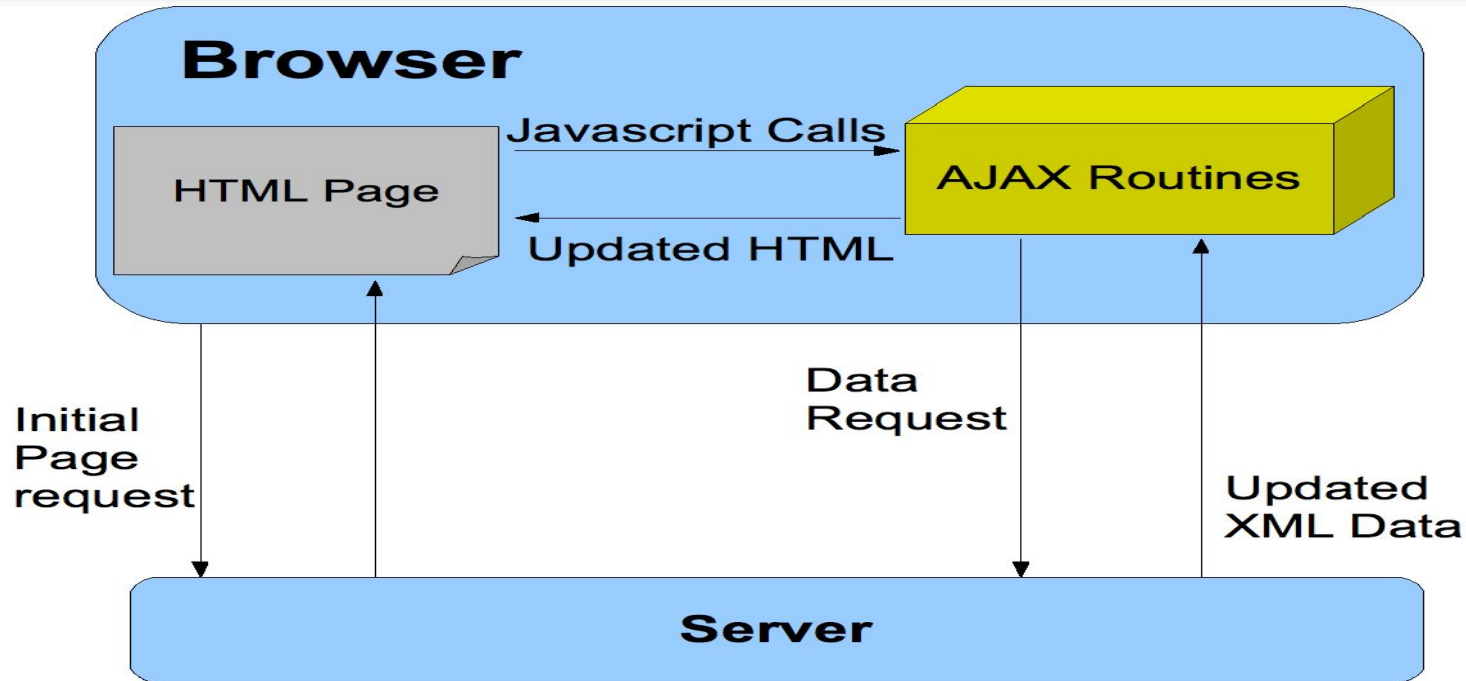
AJAX

AJAX is a short form for **A**synchronous **Ja**vascript and **X**ML.

It uses web-technologies on the client-side to create asynchronous web-applications.

With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and the behavior of the existing page..

AJAX - flow chart



AJAX

Technologies used

1. **HTML** and **CSS** for presentation
2. **Document Object Model (DOM)** for dynamic display of and interaction with data.
3. **JSON** for the interchange of data
4. The **XMLHttpRequest** object for asynchronous communication
5. **Javascript** to bring all these together.

AJAX - Request Types

GET

- Fast
- Parameters remain in browser history
- There is a restriction on data length
- Data is visible to everyone in the URL
- Less secured as compared to POST since the data is sent as the part of the URL, and also the parameters are stored in the browser history

POST

- Relatively slow
- Parameters are not saved in the browser history
- No restriction on the data length
- Data is not displayed in the URL
- More safer than POST since the data is revealed no-where and is neither stored in browser history.

Session Management - HTTP

Ah !

HTTP is a stateless protocol. And hence it does not remember anything between invocations.

But we want the server to remember some information across some specific set of requests. How can it be done ?

Session Management - HTTP

One way is to store the data in the form of cookies. Or encode it into the URL. But then this can be dangerous, as it is vulnerable to manipulated changes from the client.

What happens is. Once an authentication happens. A unique “session id” is generated and passed into a cookie. And the data corresponding to the user is stored with this session id. Thus if tampered, there won't be any leak in the information being released.

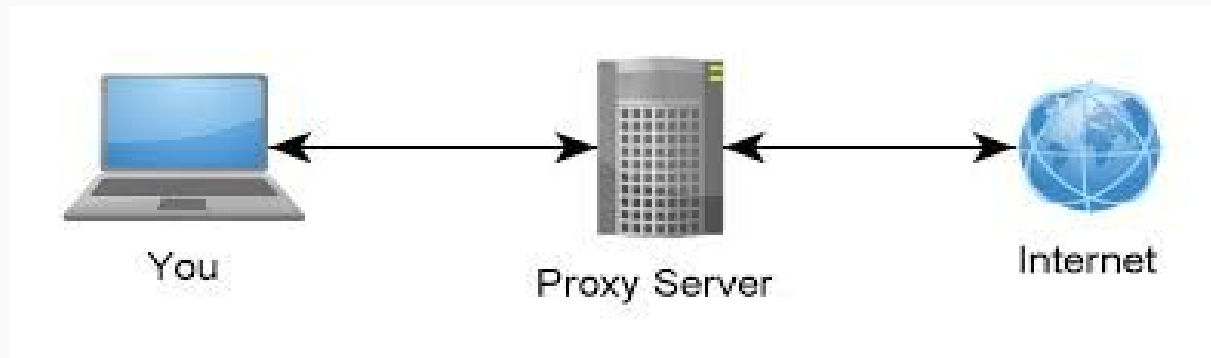
NGINX - A Reverse Proxy Server

A Proxy Server ?

A proxy server is a go-between or intermediary server that forwards requests for content from multiple clients to different servers across the internet.

NGINX - A Reverse Proxy Server

A Proxy Server ?



NGINX - A Reverse Proxy Server

A Proxy Server - Advantages

1. Act as caching server and loads the required web-page faster.
2. Masks the real IP address of the client (anonymity).

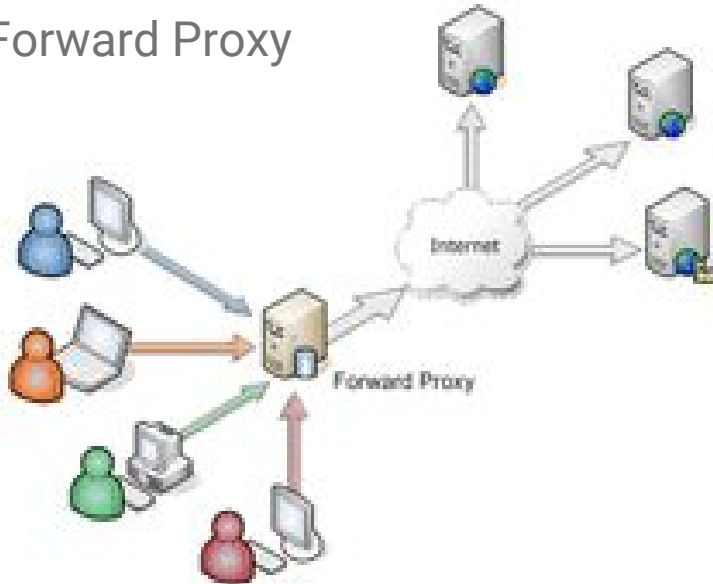
NGINX - A Reverse Proxy Server

But then Reverse Proxy ?

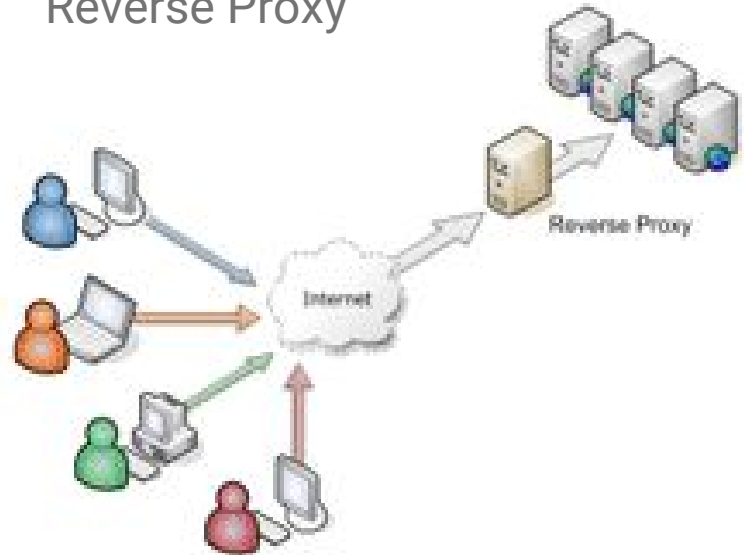
A reverse proxy server is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server. A reverse proxy ensures the smooth flow of network traffic between clients and servers.

NGINX - A Reverse Proxy Server

Forward Proxy

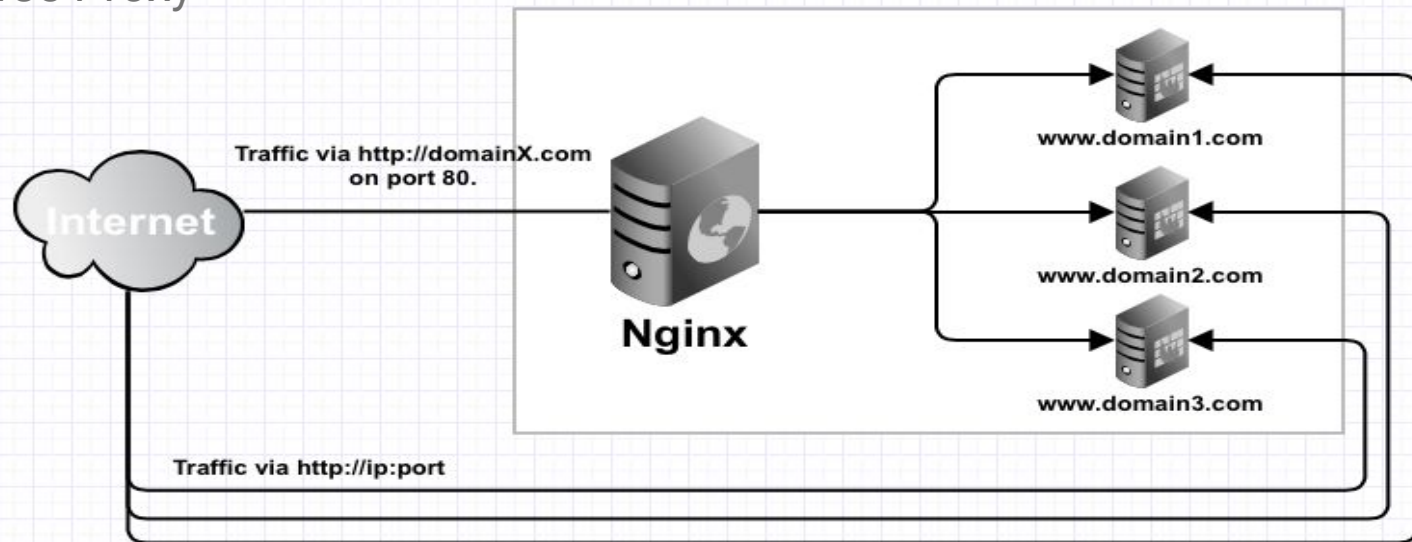


Reverse Proxy



NGINX - A Reverse Proxy Server

Reverse Proxy



NGINX - A Reverse Proxy Server

Reverse Proxy - Advantages

1. Web Acceleration
2. Security and Anonymity
3. Load Balancing



Obvious ! Since a proxy server

Acts as a traffic cop. In the case a few servers go down (off-line) or a few more servers are added, it accordingly redistributes the traffic.

Node.js

Node.js is a JavaScript runtime built on Chrome's V8 Javascript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Node.js

Advantages over Flask :

- Asynchronous
- Non-blocking I/O - leads to faster response.
- Easy to get started with because we have already used javascript in the client side.
- Lots of frameworks available for a specific task.
Ex : express, socket.io

