Back end components

Thursday, January 18, 2024 3:40 PM

Back end Servers

hardware and OS that hosts all apps necessary to run the web app

data access layer

Software

the backend server contains the other 3 backend components

- web server
- DB
- Development framework

other software components may be hypervisors, containers, WAFs

common back end server stacks :

common back end server stacks:

- LAMP = Linux Apache MySQL PHP
- WAMP = Windows " "
- WINS = Windows IIS .NET SQL server
- MAMP = macOS ...
- XAMPP = cross-platform Apachy MySQL PHP/PERL

Hardware

Many large web apps distribute load over many back end servers

Web servers

apps that run on the back end server that handles HTTP requests from browser, roots it to requested pages, and responds to browser and responds to

run on port 80 or 443

connect end users to parts of web app

accept user input within HTTP requests

responsible for routing request, run prousses needed, and return response

pages that servers route traffic to are web app core files

could create our own web servers w/ python, JS, or PHP but there are many apps to help w/

Apache

httpd

most common web server

more than 40% of all sites

more than 40% of all sites

all OS

Usually PHP but can be .NET, Python, Perl

NGINX

Second most common

30% of all sites

focus on serving many concurrent requests
with low memory and CPU load
with async architecture

60% of top traffic sites use it

IIS

internet information services

3rd most common

15%

ft developed; mainly runs on

microsoft developed; mainly runs on
ms windows servers

.NET mostly but can be for others

well optimized for AD and includes
Windows Auth to auto sign
users in to web apps

other popular servers can be Apache
Tomcat for Java and Node.JS
for JS

Databases

store content and info related to
web app

assets, content, user data

web apps can easily and quickly
d enable

web apps can easily [illegible]
store/ retrieve data and enable
dynamic content that is diff
for each user

Relational (SQL)

Store data in tables
each table has unique keys which
can link tables together

ex: users table has <u>id</u> col which
can be used as table key

another table can use this col
inside their col to link to
it

Schema - relationship between tables

Non-relational (NoSQL)

Non-relational (NoSQL)

no tables, keys, schemas

stores data in different storage models

good for dealing w/ datasets that are not well defined or structured

4 common storage models

- key-value
- document-based
- Wide column
- graph

key-value

usualy JSON or XML

key for each pair, storing all data as its value

```
{
    "1": {
        ....
    },
    "2": {
        ...
    },
    ...
}
```

Document-based model
  Store data in JSON objects
  each object has meta-data while
    Storing rest of data like
    key-value

Use in web apps
                    . Used on back end

Use in web apps

db needs to be istalled on back end server, then web apps can start using it by connecting with the programming language of choice

Development Frameworks and APIs

frameworks are to streamline and help build apps by providing functionalities like user regisration

laravel, express, Django, Rails

APIs

web APIs and HTTP request params to connect frontend and back end

less [illegible] is

end

Query parameters

default method of sending args is
GET and POST

Web APIs

usually accessed over HTTP and
handled/translated through web
servers

SOAP

simple objects access

share data through XML

XML request through HTTP, and
response is also XML

very good for transferring structured
data (class object) or binary

data (class)

data

<u>complex data</u>

also good for stateful objects; change
or store current state of page

may be difficult for beginners and
require long requests for simple
tasks

REST

representational state transfer

shares data through URL path
and usually returns output
in JSON

expect one type

focus on pages that expect one type of input passed through URL path w/out specifying name or type

good for queries like search, sort, filter