API-BACKEND DEVELOPMENT ARCHITECTURES, TECHNIQUE

In backend design and development we need to create programs, functions that can be utilized by any frontend programs by just using the backend endpoints which actually fires the code of the backend and send the results to frontend to show to the user

An API type (rest or graphql or etc.) Is architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other.

Tha API (application programming interface) are the functions that runs behind every endpoints when fired from frontend. The APIs are the design by various types of protocols, architectural structures, techniques that are well suited and best for the needs of the programs or project.

There are various types of API’s design standards or API implementation practices, which are below:-

1. REST – representational state transfer
2. GraphQL – graph query language
3. gRPc – google remote procedure call
4. SOAP – simple object access protocol
5. WebSocket
6. WebHook
7. Many more

General terminology while studying API’s

1. State – information of the request or user , if stateless then api does not know the request from where it is coming , if statefull then api know from where it is coming
2. Restfull API – means a API which follow rest architecture or a rest api
3. Microservice architecture – a development architecture where main project is divided into multiple micro functions connected to each other , independent from each other
4. Monolithic architecture – a development architecture where main project as a single function or a single program , useful for smaller projects
5. HTTP Status code – when an api return some response while answering the request , there is status code attached to it which defines the response if it is success , failure , success with exceptions or failure with some success , majorly only five types of status codes are used which are below:-
6. Success – 200 ok (2xx)
7. Failure – 400 bad request(4xx)
8. Fatal failure - 500 internal server error (5xx)
9. Informational – 100 continue (1xx) , 101 switching protocol
10. Redirectional – 301 moved permanently (3xx)
11. Caching – caching is the technique, approach to store the data temporary in a storage which is accessible faster than database storage or regular storage. Types of cache – browser cache(web browser stores the current page, last page static files such as html , css, js for temporary to local storage for faster reload rather downloading it again) ,server cache – data is stored on server ram or server disc , CDN cache – network of servers that cache the data close to the user’s location for faster travel of data over internet , database cache – database can cache the results or data , application cache – application or running programs or functions can cache their own data for a period of time in example an object present in memory .
12. Endpoint – an endpoint is a URL, a link following a protocol generally http by which a client can access resources.
13. Payload – the data or actual data sent with the request which will be used by api to generate appropriate response.
14. Headers – additional data sent with request which tells the information about request , authentication , content type , in key –value pairs.
15. Authentication and authorization – authentication is verifying the user identity , if a user present in the database, and authorization is the access permission if a user has the permission to access the data or has the permission to perform certain actions and functions.
16. Token – a token is a string of characters which Is generally used to identify the user identification over internet , a token is like a digital key by which a api can identify the user identity by validating token , a token is created when a user logged in first time and after that whenever user visit the site again if token is not expired then user will not have to login again , browser store the token generated till the expired date .we can store the token in multiple ways.
    * 1. In server side- while user logged in first time we can store the token in server side in databases.
      2. In client side – in cookies , local storage, session storage
      3. Using jwt – json web token , self contained tokens , stored on client side , scales well
      4. Using third party softwares
17. Rate limiting – techniques used to control the no of requests a client can make to server
18. API versioning – creating different versions as per the development cycle of the project.
19. CORS – when a web browser interact with application first time then it send a request to the frontend code , but if frontend code and backend code are hosted on different domains then the origin of request changes for example , a request from user goes to [www.google.com](http://www.google.com) (actual ip address can be like 198.23.54.45:portno) and google.com interact with backend code on [www.backendgoogle.com](http://www.backendgoogle.com) (actual ip address can be like 198.23.54.55:portno) then the origin of the request is different , and here comes the security feaures provided by web browser called Cross origin resource sharing . if in case there are need to implement this then we need to follow the instructions to implement CORS from the official CORS documentation of how to implement CORS.
20. API gateway – a point of entry of requests coming from client , a api consumer . for big projects we need to implement api gateway which implement features like rate limting , authentication and authorization , load balancing , protocol translation , service discovering , monitoring of api request , logging system , analysing of request , billing of request , implemting caching .
21. API Documentations - OpenAPI/Swagger – a specification and tool for defining and interactive documentation of API’s
22. Pagination – a technique of defining large data (querysets) into smaller data which are more manageable and easy to transfer between request and response cycle.
23. Middlewares – a function or a software acts as an intermediary between request and response cycle. Often performs functions like logging, authentication, authorizations and other functions as per the needs of projects.
24. Data Serialization – process of converting data into a format which can be understand by the web or client, generally frontend. When creating backend the database send the data in format of object which is not understandable or readable so we need to convert it into general format which is json format and then sends the data to frontend.
25. API Testing – when a api is created then we need to create different test cases and to create different test cases we define different functions which will perform the testing of the api and generally it test the efficiency and speed and per request handling time and etc.
26. Dependency Injection – this is a advance concept used in oops. Easiest explanation is belo-

When you go and get things out of the refrigerator for yourself, you can cause problems.

You might leave the door open, you might get something Mommy or Daddy doesn't want

you to have. You might even be looking for something we don't even have or which has

expired.

What you should be doing is stating a need, "l need something to drink with lunch," and

then we will make sure you have something when you sit down to eat.

