Dell - Data Engineering Training

Uday Kumar – Data Platform Architect

Day 2

Data Engineering Training

Agenda

1. API Fundamentals
2. What is REST API
3. Introduction to FAST API
4. Role of FAST API in Data Engineering Tasks

API Fundamentals

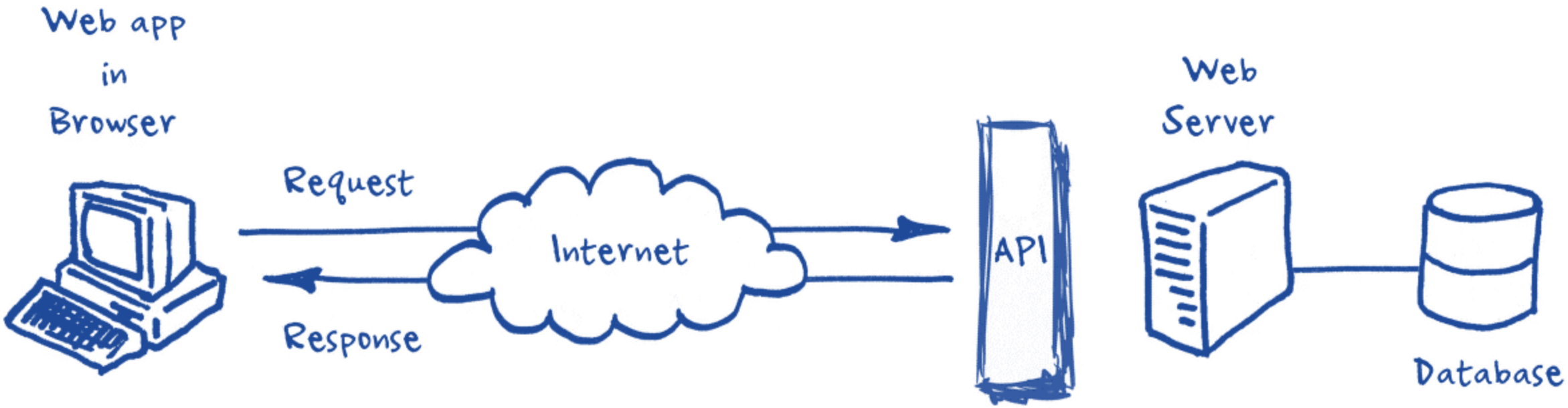
API Fundamentals

The growth of the **API Market in the US** continues to climb year over year, with an expected increase of **34% - totaling a projected $7.5B market size in 2026**.

From massive global corporations to local businesses, the widespread use of APIs has permanently changed the face of all major industries.

API stands for **Application Programming Interface**. An API is a software intermediary that allows **two applications to talk to each other**. When you use an application on your phone, the application **connects to the Internet and sends data to a server**. The server then processes the data and sends it back to your phone. The application on your phone then **interprets the data and presents it to you in a readable way.**

API Fundamentals



API Fundamentals

Remote Procedure Call, RPC.

Clients can call functions on the server.

Remote Method Invocation, RMI.

Clients can call methods on objects on the server.

Representational State Transfer, REST.

Clients can apply CRUD operations on resources on the server.

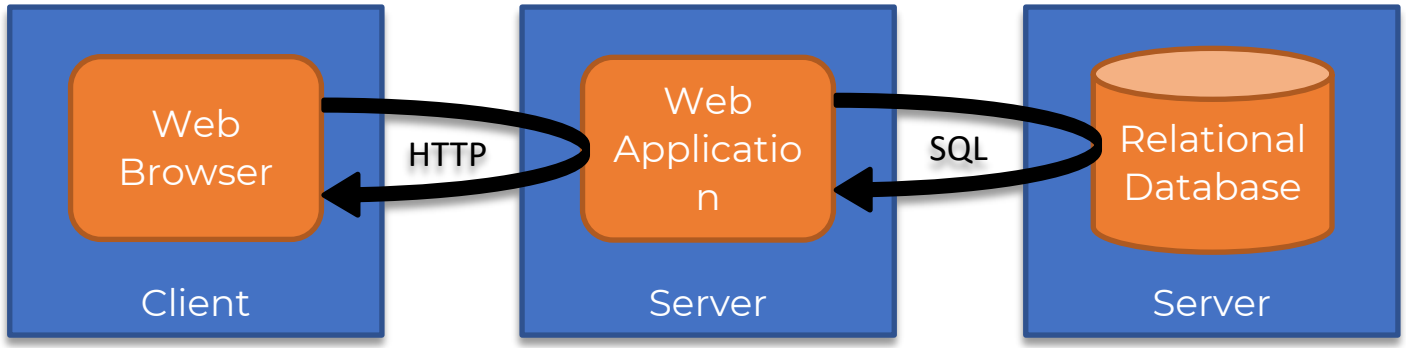
What is REST API ?

WHAT IS REST API ?

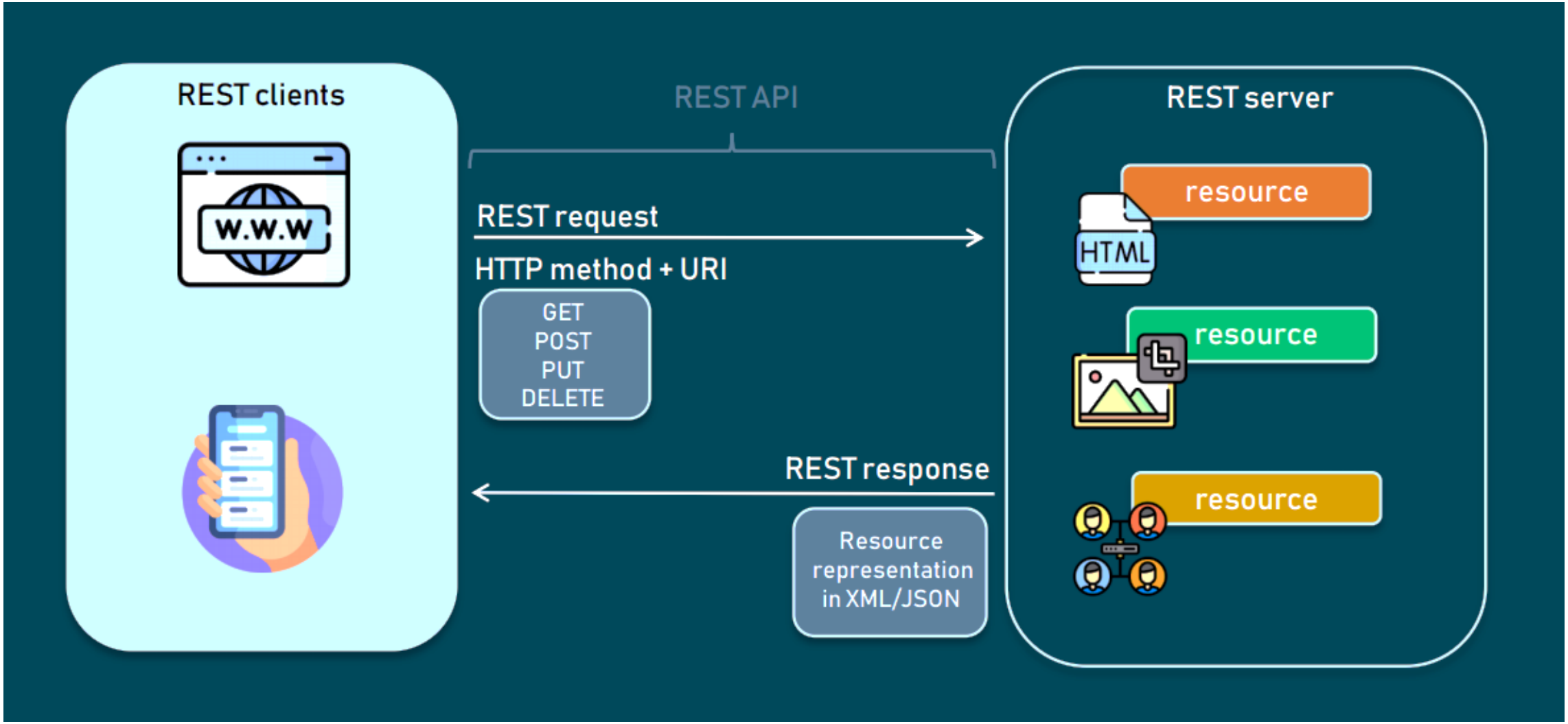
An architectural style for distributed hypermedia systems described by Roy Thomas Fielding in his doctoral dissertation 2000.

Consists of constraints:

- Client - Server
- Stateless
- Cache
- Uniform Interface
- Layered System
- Code-On-Demand



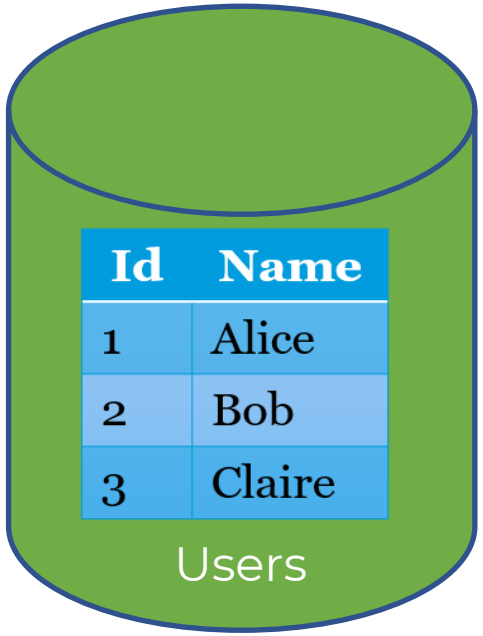
WHAT IS REST API ?



WHAT IS REST API ?

Client

Server



GET /users/2

...

{"id": 2, "name": "Bob"}

Changes state.
{"id": 2, "name": "John"}

PUT /users/2
{"id": 2, "name": "John"}

WHAT IS REST API ?

Use URIs to identify resources.

Use HTTP methods to specify operation:

Create: POST (or PUT)

Retrieve: GET

Update: PUT (or PATCH)

Delete: DELETE

Use HTTP headers

Content-Type and **Accept**

to specify data format for the resources.

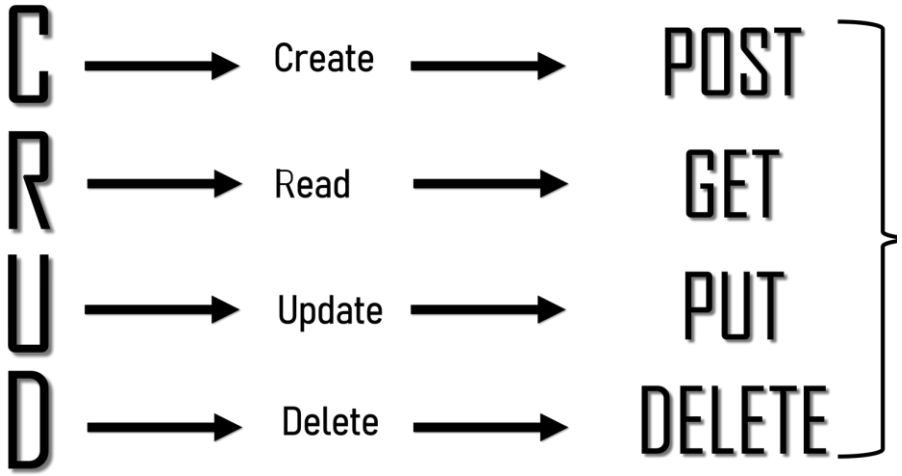
Use **HTTP status code** to indicate success/failure.

Bad

POST /login
POST /create-product
GET /get-top-10-products

Good

POST /login-sessions
POST /products
GET /top-10-products



WHAT IS REST API ?

Use URIs to identify resources.

Use HTTP methods to specify operation:

Create: POST (or PUT)

Retrieve: GET

Update: PUT (or PATCH)

Delete: DELETE

Bad	**Good**
POST /login	POST /login-sessions
POST /create-product	POST /products
GET /get-top-10-products	GET /top-10-products

Use HTTP headers

Content-Type and **Accept**

to specify data format for the resources.

Use **HTTP status code** to indicate success/failure.

WHAT IS REST API ?

REST API Security: Best Practices

There are a myriad of ways to break into your API and cause significant damage to your company.

The average cost of a data breach is estimated to be **$8.64 million for US-based companies**.

Considering that **83 percent of consumers will stop the relationship with a company** that became a victim of a successful cyberattack, securing API has never been more critical.

Protect your API against simple cyberattacks:

Always use HTTPS: You should always use SSL to ensure higher security standards.
Use password hashing: Password hashes add another layer of security, protecting the integrity of sensitive data even if a password was compromised.
Avoid exposing sensitive data in URL strings: Any data that hackers can potentially use to break into your system, from usernames to session tokens, must not be included in the URL string.
Implement OAuth: OAuth is a widely recognized authorization framework allowing data exchange without exposing sensitive information

Introduction To FASTAPI

Introduction To FASTAPI

FastAPI is a high-performing **web framework for building APIs with Python 3.7+** based on standard Python type hints. It helps developers build applications quickly and efficiently. **FastAPI** is built on top of the **Starlette web server** and includes features that make building web applications easier, such as **automatic data validation, error handling**, and **interactive API docs**.

FASTAPI Features

Performance: On par with NodeJS and the Go language.

Speed: Increase the development speed 2-3X.

Easy: Great editor support. Completion everywhere. Easy to learn and use.

Robust: Production-ready code with automatic interactive documentation.

OpenAPI based: Fully compatible with OpenAPI and JSON Schema.

FASTAPI Comparison

	Django	Flask	FastAPI
Community	Big. 66K GitHub stars	Big. 61K GitHub stars	Big. 50K GitHub stars
Performance	Django is massive. It isn't the best in terms of performance.	Flask is a micro web framework. It performs better than Django.	FastAPI is one of the fastest web frameworks with native async support that adds to the efficiency of the framework.
Async support	Yes, with limited latency.	No. Needs Asyncio	FastAPI provides native async support.
Ease of use	Django is massive and hence a bit complicated to learn.	Flask is easy to learn and pretty straightforward in use.	FastAPI is the simplest of all three.
Interactive Documentation	Not interactive	No	Yes (OpenAI, Redoc)
Data Verification	No	No	Yes

FASTAPI Installation

Installing FastAPI

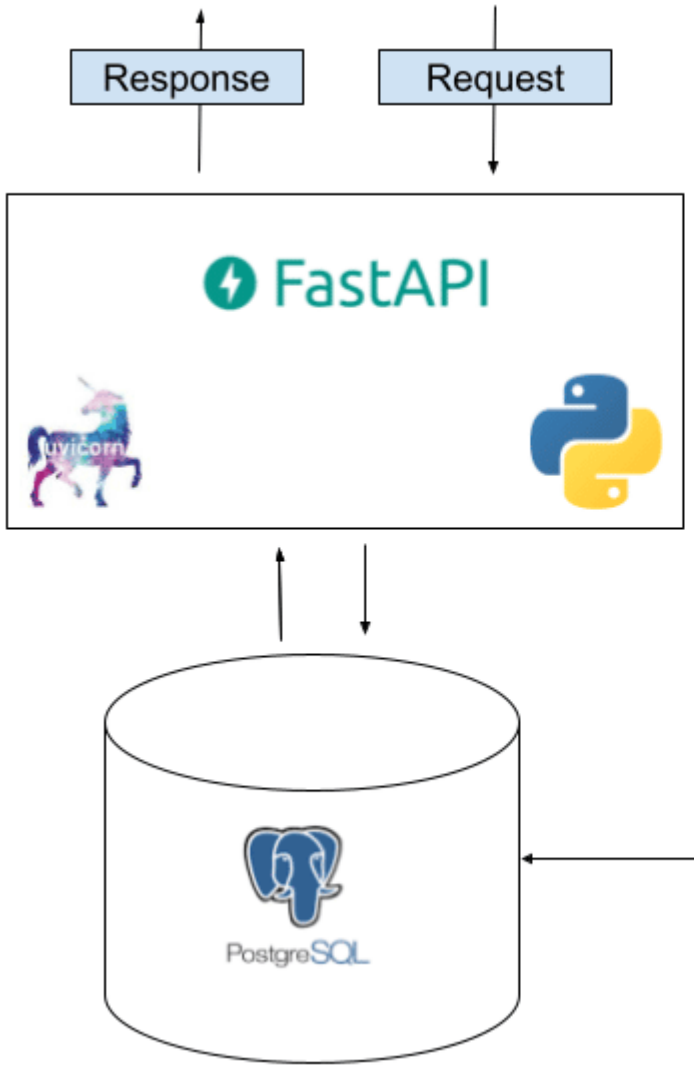
FastAPI requires Python 3.7+. It can be installed using pip. You will need to install FastAPI and the ASGI server `uvicorn`.

```
# install fastapi
pip install fastapi

# install uvicorn
pip install uvicorn
```

Role Of FAST API in Data Engineering

ROLE OF FASTAPI IN DATA ENGINEERING



User		
pk	id	UUID
not null	first_name	string
not null	last_name	string
not null	gender	gender
not null	roles	list strings

Contact Us

080-4524-9465

support@intellipaat.com