

API connect is the one of the integration tool of dataPower.

It's runtime is DP.

While we send some request get some response.

EX:- When you click on the button something (or) something happened at backend is called API.

The action is nothing but an API

To develop the APIs we use IBM API Connect.

IBM DataPower Gateway is a single gateway platform that provides security, control, integration.

In DP we need to have all the requirements ready before we start the development.

DP is at service level we can change the whole service.

But in API Connect it is different.

→ API Connect is customer level which means we can change our service acc to need of customer.

EX:- hotStar /

API Monetization :-

Generating the revenue through
the use of APIs.

Clients use services through API
client sends 1000 request we can charge
like 1\$ for 1000 requests.

API components :-

→ API Manager = It manages all APIs and Products & Analytics. A product, is the collection of API's. member, catalog & space =

→ Cloud Manager = It manages multiple API Manager Admins (PAP, SSL/TLS, Gateway, security) and Analytics. (graphs & stats)

→ Developer Toolkit = Running commands for APIs. like in Java we use command prompt.

→ Developer Portal = We expose socialize the API's to clients in order to consume. EX: Ecommerce websites.

→ Gateway Service → Runtime overview which is DP.

Instant developer
feedback
monitoring

75208

SOAP \Rightarrow Simple Object Access Protocol

Rest \Rightarrow Representation State Transfer.

SOAP API

Transports data
in standard
XML format

Because it is XML based
it works
with

WSDL \rightarrow Web Service Description Language
it is XML format for describing
network.

Designed with large
enterprise application
in mind.

Rest API

Generally Transports
data in JSON

It is based on URL
Because Rest follows

Stateless model.
XML (or) JSON

It works with
Get, Post, Put
Delete

Designed with
mobile device in
mind

API design

Soap, Rest.

→

Soap (It is for the last version we are using)
Used for older Application.

Rest => We are using now.

It is a Json, XML, but is nothing but
a plain XML.

It uses HTTP methods protocols for
data communication.

→ GET → Provides read only access (list user to get all data)
 used to update / replace a resource → PUT → update when used to change existing one.
 used to update an existing one → POST → create we need to give all create.
 → PATCH → Partially update the data.
 → DELETE → used to remove an resource.

(Keep : takes any method based on the user input.

Swagger :-

Section :-

Info = It contains title, name, version, contact

consume = It is nothing but request format like json or xml. We have stored.

produce = It is like response (like json or xml) same.

Path = In order to hit a particular one.

Ex :- It is like like an url.

To add the value of the catalog level.
 operation in get - /api/users?page=2
 Path
 API It contains more one
 in Post - /api/register.

definit - It is nothing but structure of results and response.

Parameter = They are unique key which we will use across operation. In DB we have primary key

Path Parameter = If we want to fetch a details with in w
 query Parameter { /get NOT details /vm 3328

= If we want to filter the details.

Schemes = They are nothing but Protocols
http, https (default)

basePath = uniquely path for each api.
/ngts-training.

host = It is nothing but an API url.

Syntax (catalog.host) whatever gateway
using in API url.

tags = It is like a keywords

For ex (mytagname tag).

(When we entering customer details)
We need (customer Tag).

External doc = addition external documentation.

Properties = Fixed values.

before hitting backend

We use credentials

Assigning
a value
to a variable.

Storing something in this
like any number repeatedly
Not typing we using a one.

Swagger

Universal Format -

↳ YAML

↳ RAML

→ open API 2.0 & 3.0 ^{not} → Fully Stable yet

API'S : 3 different API'S we have

→ New API (Rest)

→ OAuth API (Security API)

→ Soap API.

New API (Rest) :

Sections

Info = Name, Title, Version, Contact.

Description ⇒ What API giving

Version = 1.0.0

⇒ Main · Major · minor

Schemes = What type of Protocol

http

https (default) common use.

WebService, WPS.

Host = By default \$ (catalog.Host)

Syntax which states whatever gateway is using on API.

Base Path = unique URI which we use to identify an API from the rest of APIs.

Basical API Connect supports 2 media
Consume & Types JSON, XML

What is the type of request and response.
Product

Life cycle :- 3 Phases We have ^{completely} ~~end to end~~.
identified → Identify the requirement
specified → ~~where~~
Realized

Identified → identify the requirement completely
end to end

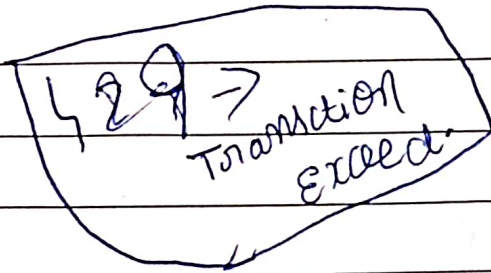
specified → When user in a stage user waiting
for feedback we have requirement

Realized → Actual development starts here.

Security → selection of security.

Security definition & =

1. api Key - client id, key
2. basic - username Pwd etc app.
3. oauth - it is a Token mechanism.



Properties : Fixed value before
storing something in the value
ex: Assigning a value to a variable

Paths : operation Path What kind of operation
we can perform in the API.

Parameter
operations → Get, Put, Post

Parameter : It is a unique which we are used a unique
operation

Path = If we want to fetch details within
uri

query = If we want to filter the
details.

Defination :- Structure of The request
and API is going to consume & produce.

Response :- 404 - Error Not found
403 - Forbidden
401 - Unauthorized error

Service :- An other Additional Policy
If You want to add any features to API.

Tag :- It is ^{to define} a keyword in APIs.
ex: Instagram
any person we need to Tag

Categories :- Where we are classify
in APIs