REST API Fundamentals

learn more

Testleaf
Always Ahead

Why API Automation is growing in demand?

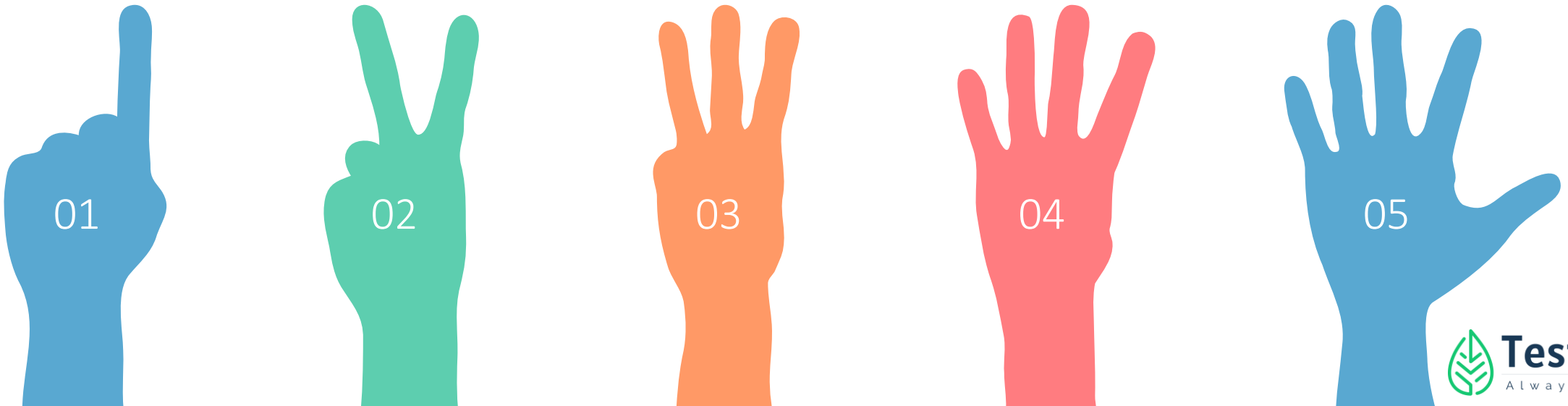
API Based Apps

Faster

Easier & Isolated

Data Generation

Less Failures



Testleaf
Always Ahead

Why API Automation is growing in demand?

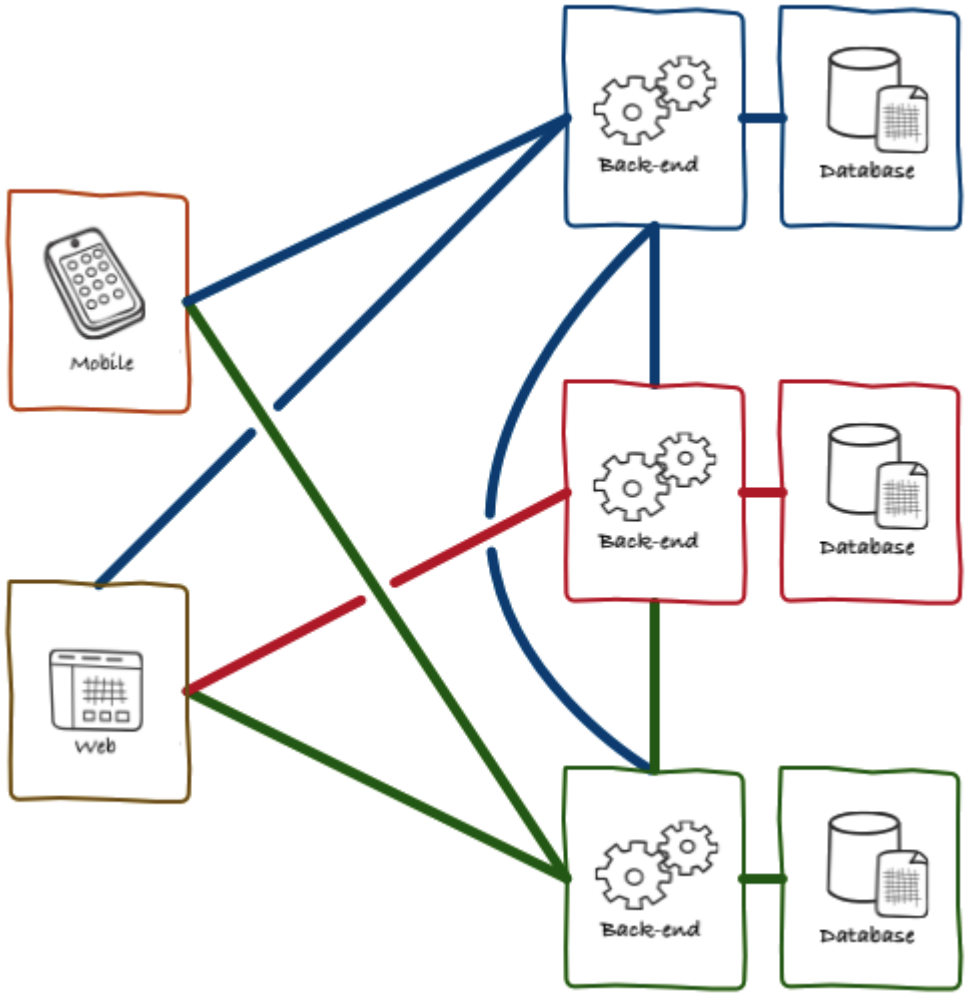
API Based Apps

Testleaf
Always Ahead

API First Design Strategy – Leading Italian Bank

Testleaf
Always Ahead

API First Design Strategy – Leading Italian Bank



Testleaf
Always Ahead

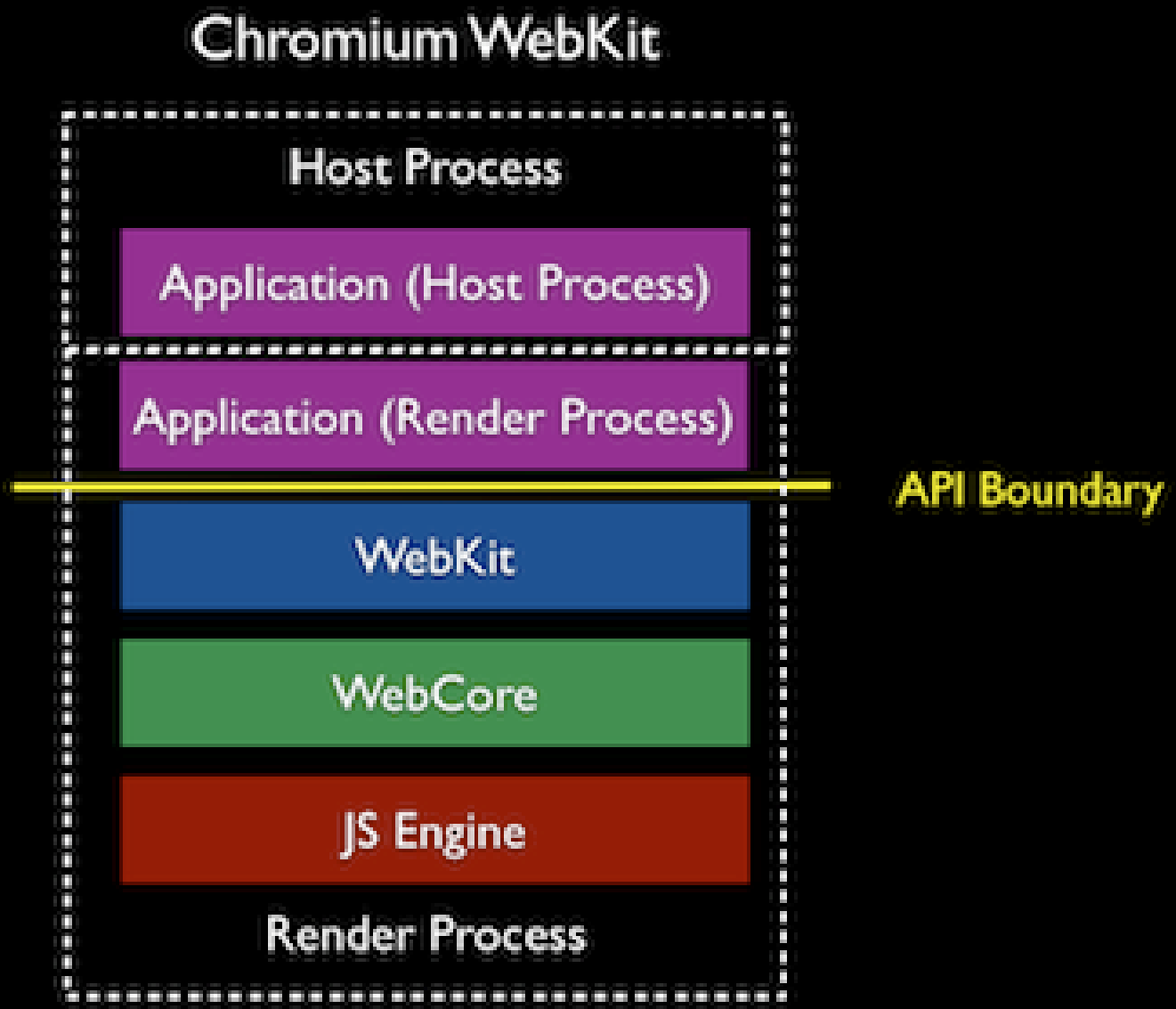
Why API Automation is growing in demand?

Faster

Testleaf
Always Ahead

Rendering Engine

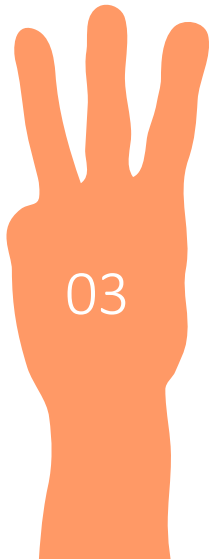
Rendering Engine



Testleaf
Always Ahead

Why API Automation is growing in demand?

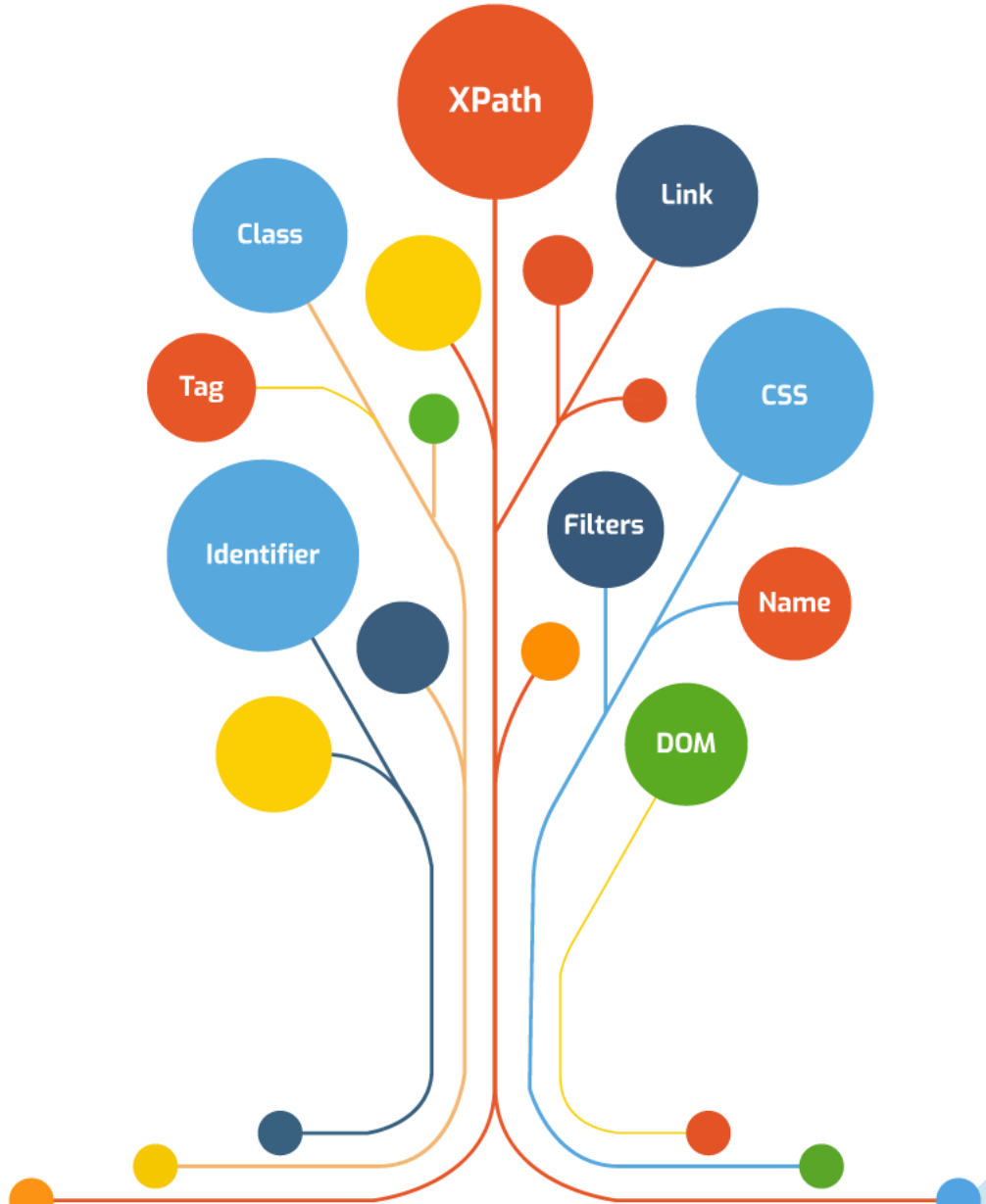
Easier & Isolated



Testleaf
Always Ahead

Browser Specific

Element Challenges



Why API Automation is growing in demand?

Data Generation

Testleaf
Always Ahead

Why API Automation is growing in demand?

Less Failures

Testleaf
Always Ahead

Element Not Found Challenges

Time Out (Behavioral) Failures

Summary !!

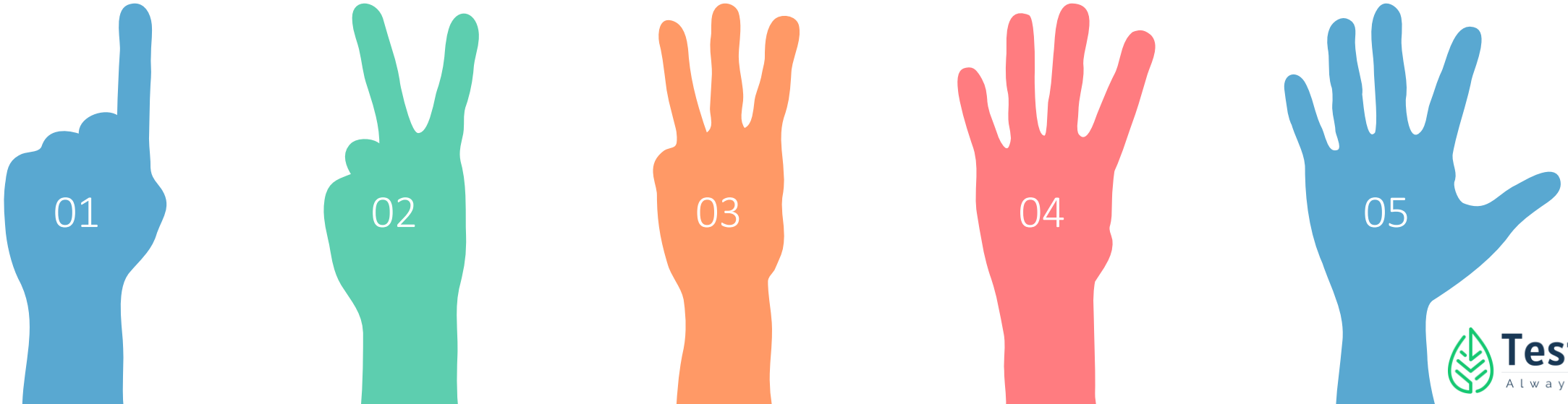
API Based Apps

Faster

Easier & Isolated

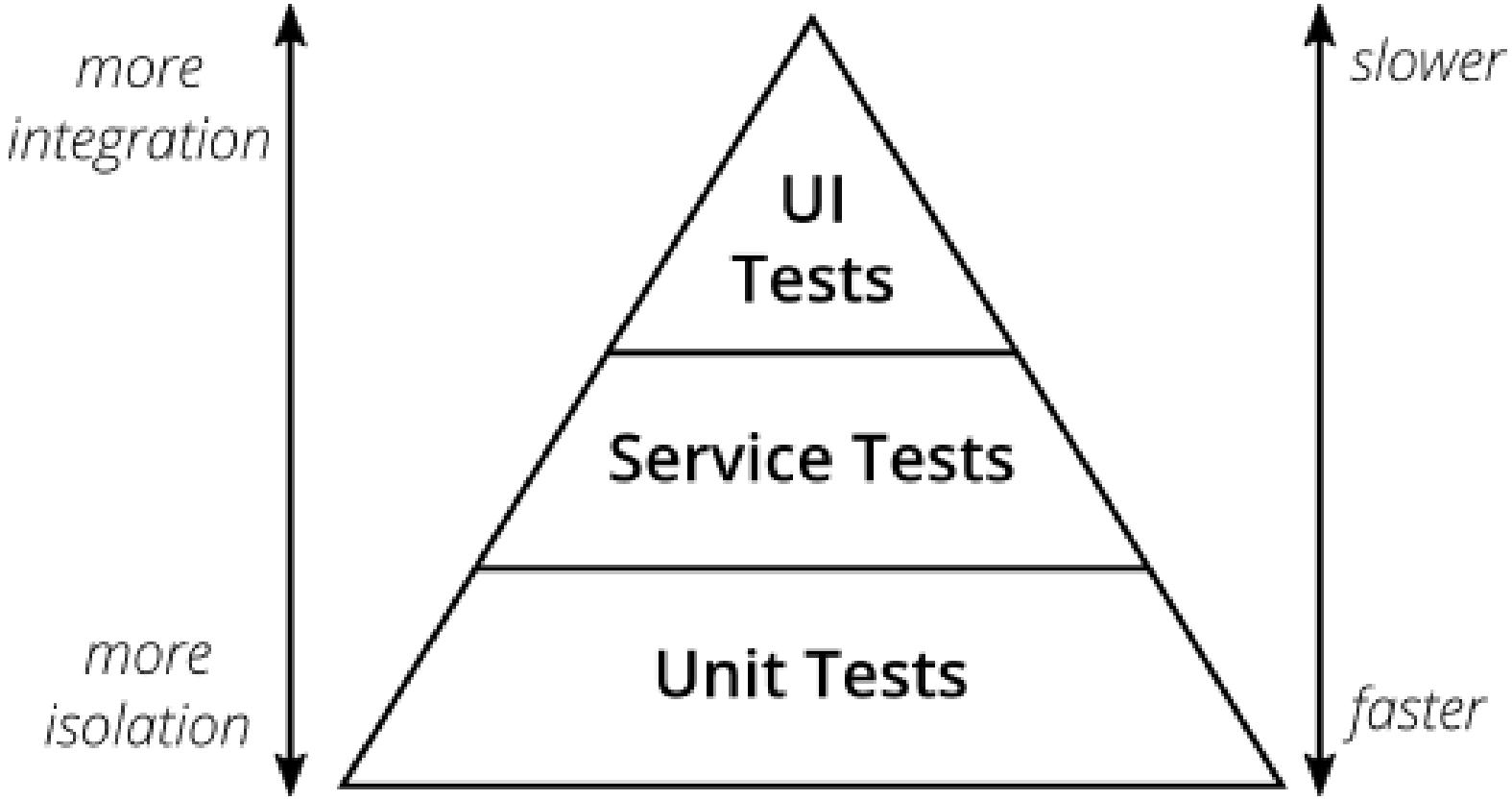
Data Generation

Less Failures



Testleaf
Always Ahead

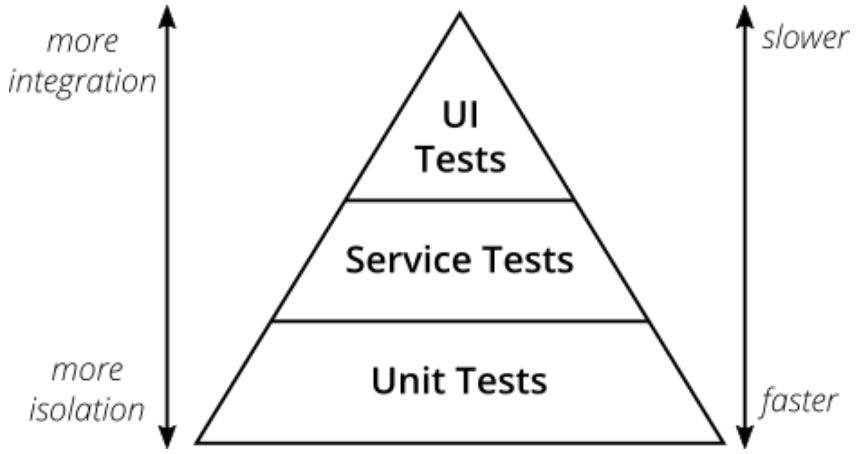
Mike Cohn' Test Pyramid



Testleaf
Always Ahead

Test Pyramid - Take Away

1. Write tests with different granularity (End to End tests in UI)
2. The more high-level you get the fewer tests you should have



Testleaf
Always Ahead

SOAP vs REST vs GraphQL vs gRPC

SOAP - Simple Object Access Protocol (1998)

REST - Representational State Transfer (2000)

GraphQL – Graphical Query Language (2015)

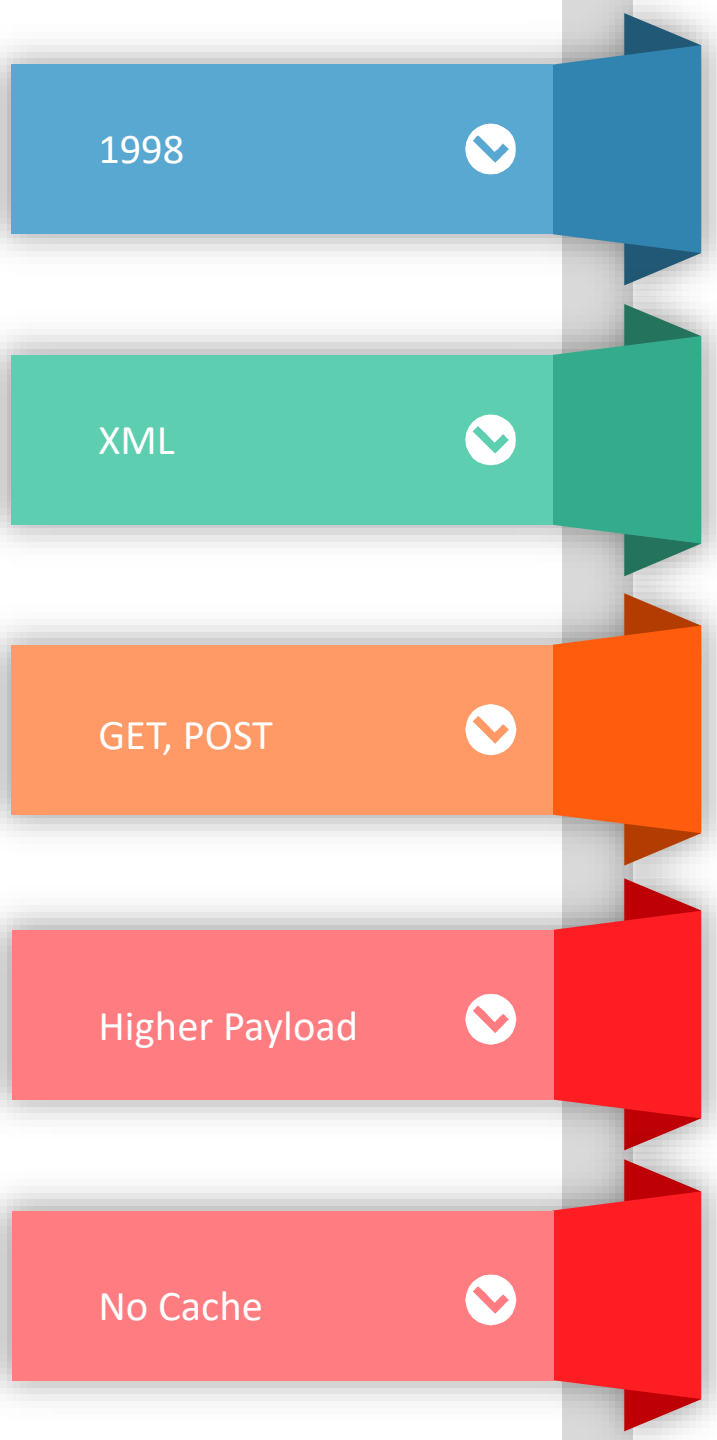
gRPC – Google Remote Procedural Call (2016)

Testleaf
Always Ahead

SOAP

SOAP (Simple Object Access Protocol) is a messaging protocol specification for exchanging structured information in the implementation of Web Services.

Element	Description	Required
Envelope	Identifies the XML document as a SOAP message.	Yes
Header	Contains header information.	No
Body	Contains call and response information.	Yes
Fault	Provides information about errors that occurred while processing the message.	No

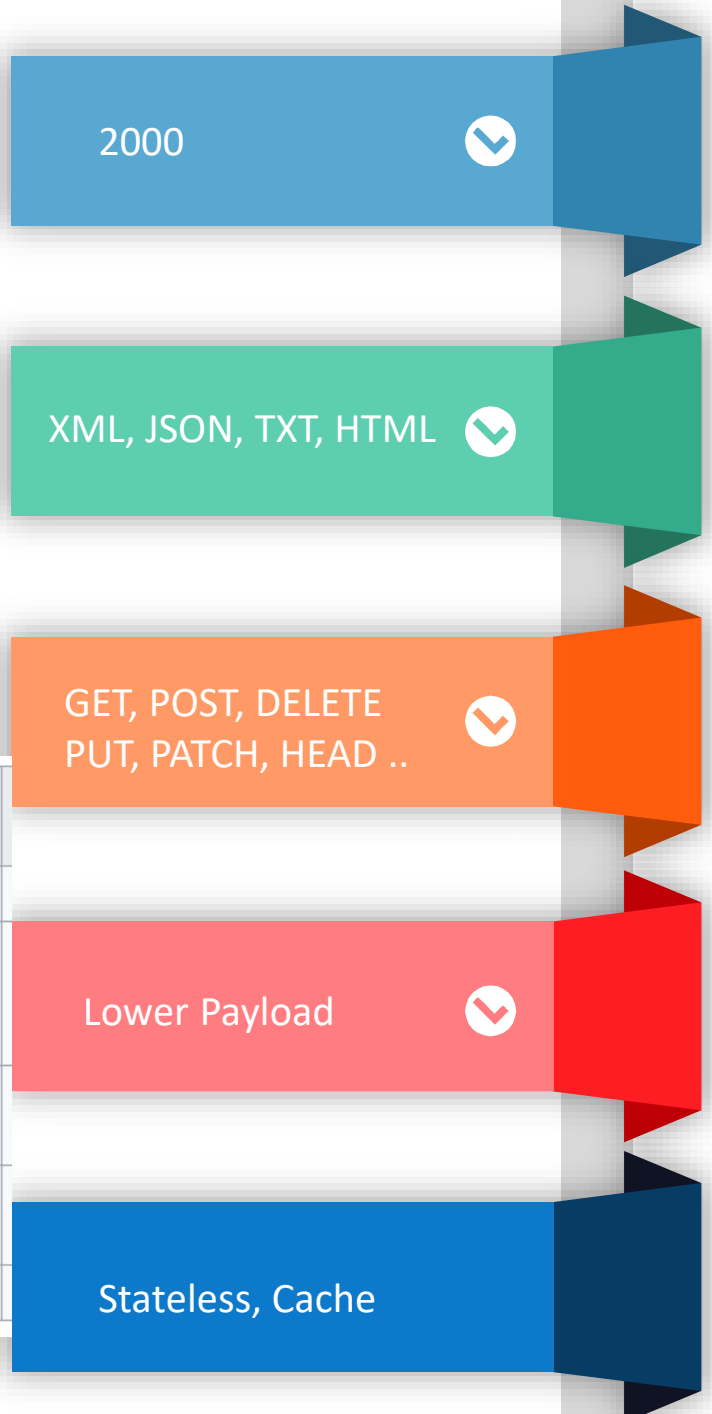


Testleaf
Always Ahead

REST

Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web Services

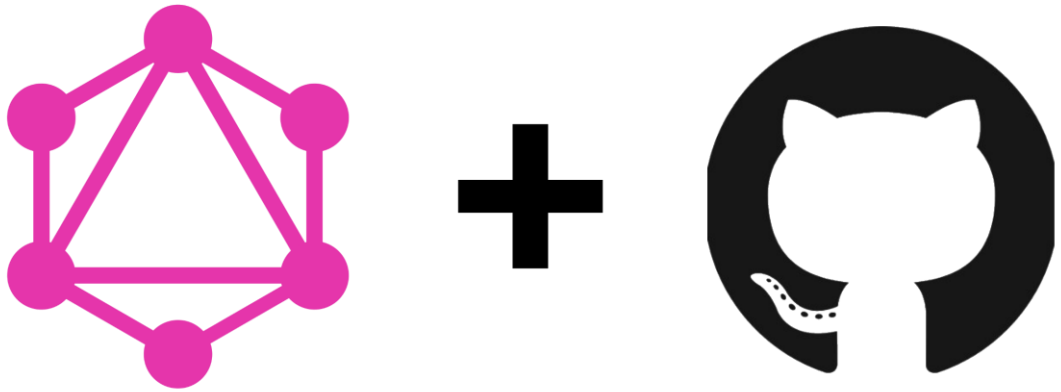
HTTP METHODS	Collection resource, such as `https://api.example.com/collection/`
GET	*Retrieve* the URIs of the member resources of the collection resource in the response body.
POST	*Create* a member resource in the collection resource using the instructions in the request body. The URI of the created member resource is *automatically assigned* and returned in the response *Location* header field.
PUT	*Replace* all the representations of the member resources of the collection resource with the representation in the request body, or *create* the collection resource if it does not exist.
PATCH	*Update* all the representations of the member resources of the collection resource using the instructions in the request body, or *may create* the collection resource if it does not exist.
DELETE	*Delete* all the representations of the member resources of the collection resource.



Testleaf
Always Ahead

GraphQL

GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data

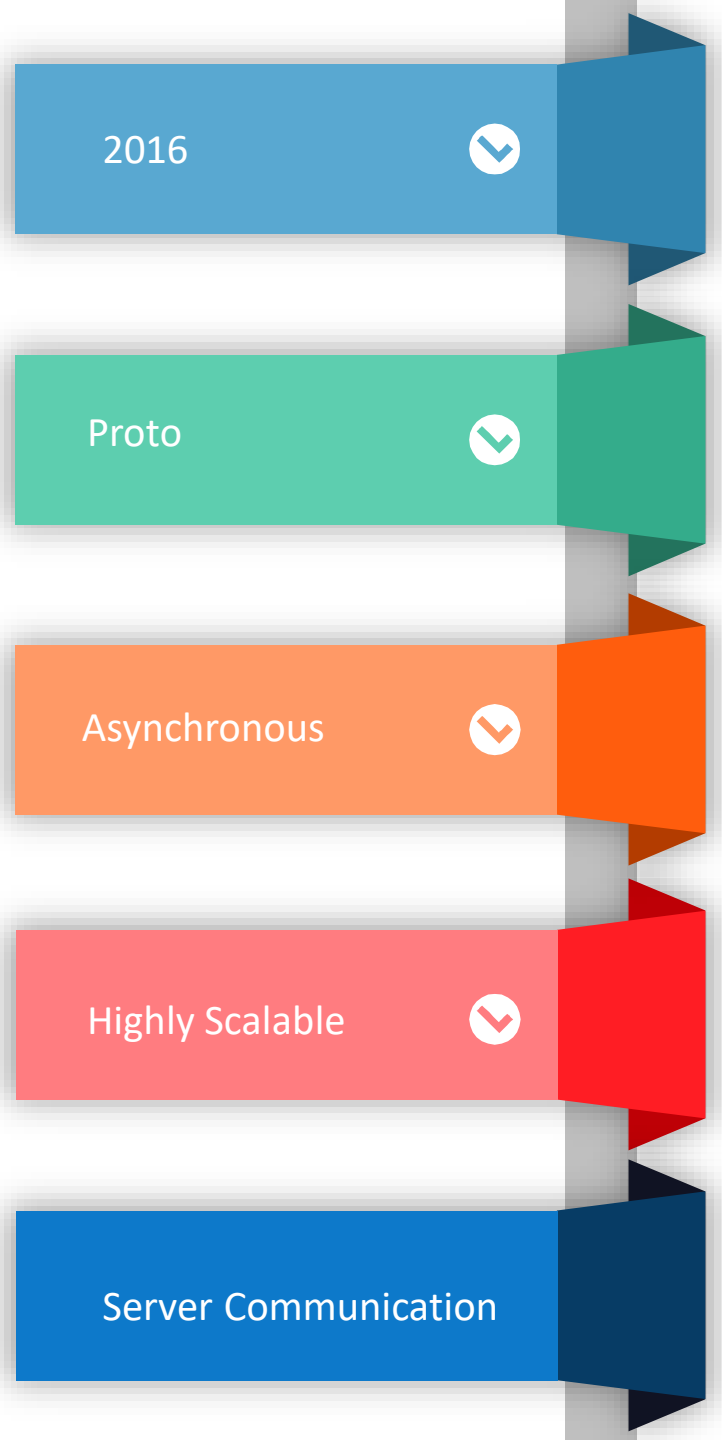


- 2015
- JSON
- Asynchronous
- HTTP, AMQP, MQTT
- Send Only required data

Testleaf
Always Ahead

gRPC

A nimble and lightweight system for requesting data. gRPC, on the other hand, is best used when a system requires a set amount of data or processing routinely



Testleaf
Always Ahead

HTTP Methods (CRUD)

1

GET

Retrieve resource information (READ)

2

POST

Create new subordinate resources (CREATE)

3

DELETE

delete resources (DELETE)

4

PUT

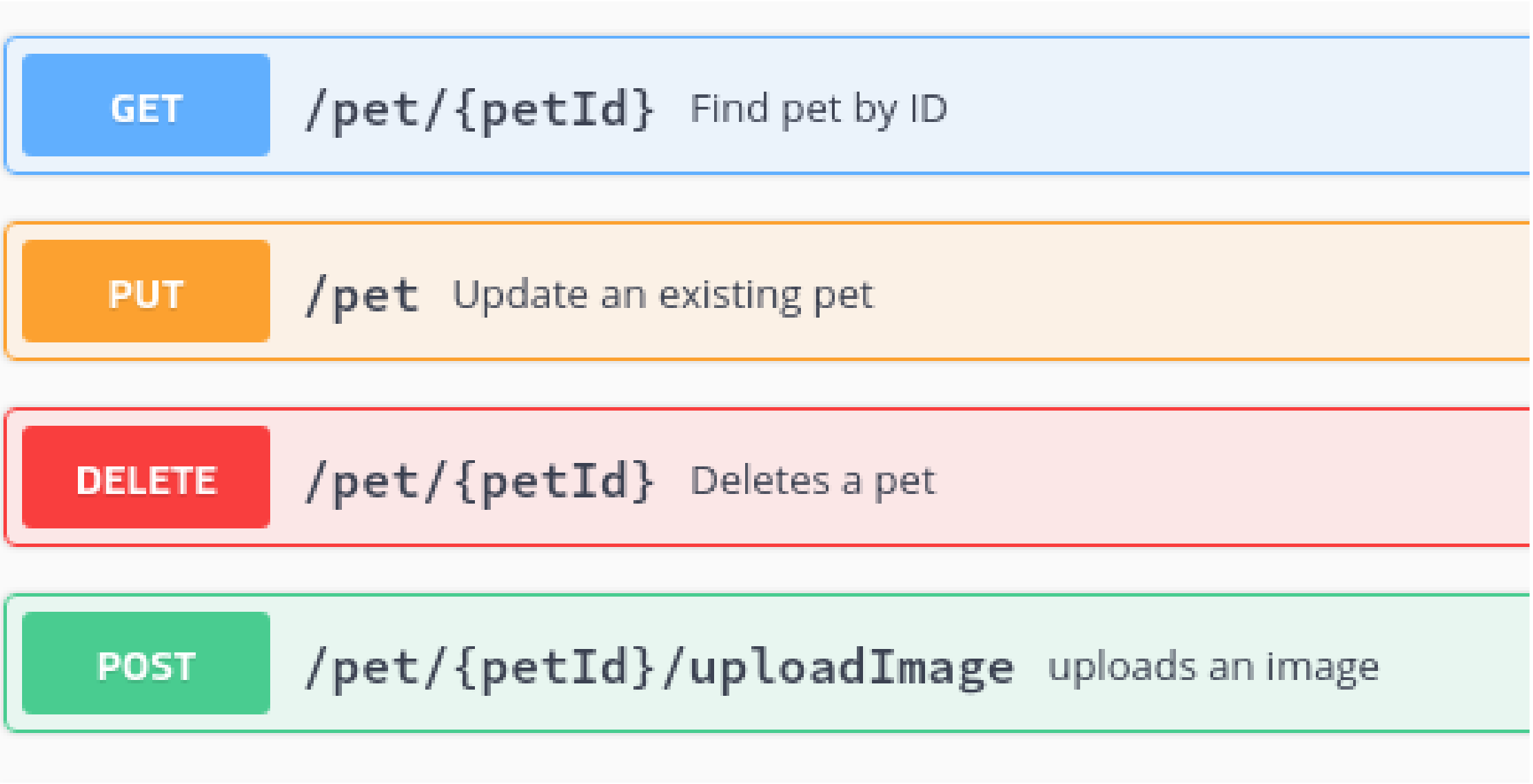
primarily to update existing resource (REPLACE)

5

PATCH

make partial update on a resource (UPDATE)

Testleaf
Always Ahead


GET
/pet/{petId}
Find pet by ID
PUT
/pet
Update an existing pet
DELETE
/pet/{petId}
Deletes a pet
POST
/pet/{petId}/uploadImage
uploads an image

Authentication

No Authentication

Examples: Weather Forecast, Google Map

Basic

Examples: ServiceNow, JIRA

OAuth 2.0

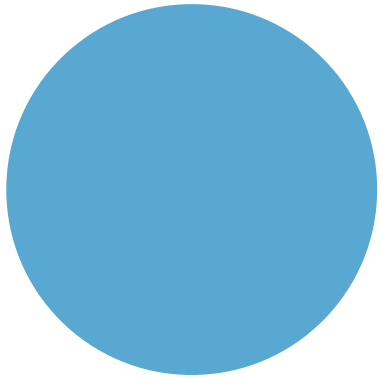
Examples: Payment Gateways, ServiceNow

Token

Examples: Oracle WMS, JWT

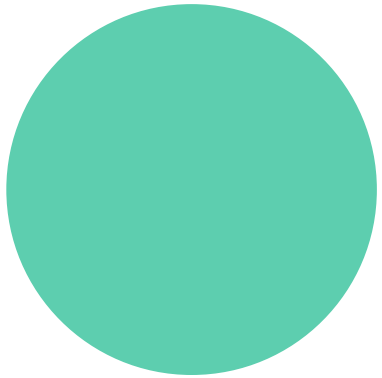
Testleaf
Always Ahead

HTTP Headers



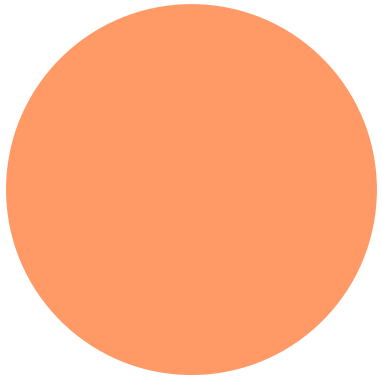
Content-Type

The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient. **(request)**



Accept

The Accept request-header field can be used to specify certain media types which are acceptable for the **response**.



Cookie

The Cookie request-header field value contains a name/value pair of information stored for that URL

Testleaf
Always Ahead

Documentation Tool

VS

https://covid-19-apis.postman.com/

Testleaf
Always Ahead

Request Parameters

Header

Request header, usually related to authorization.

Query

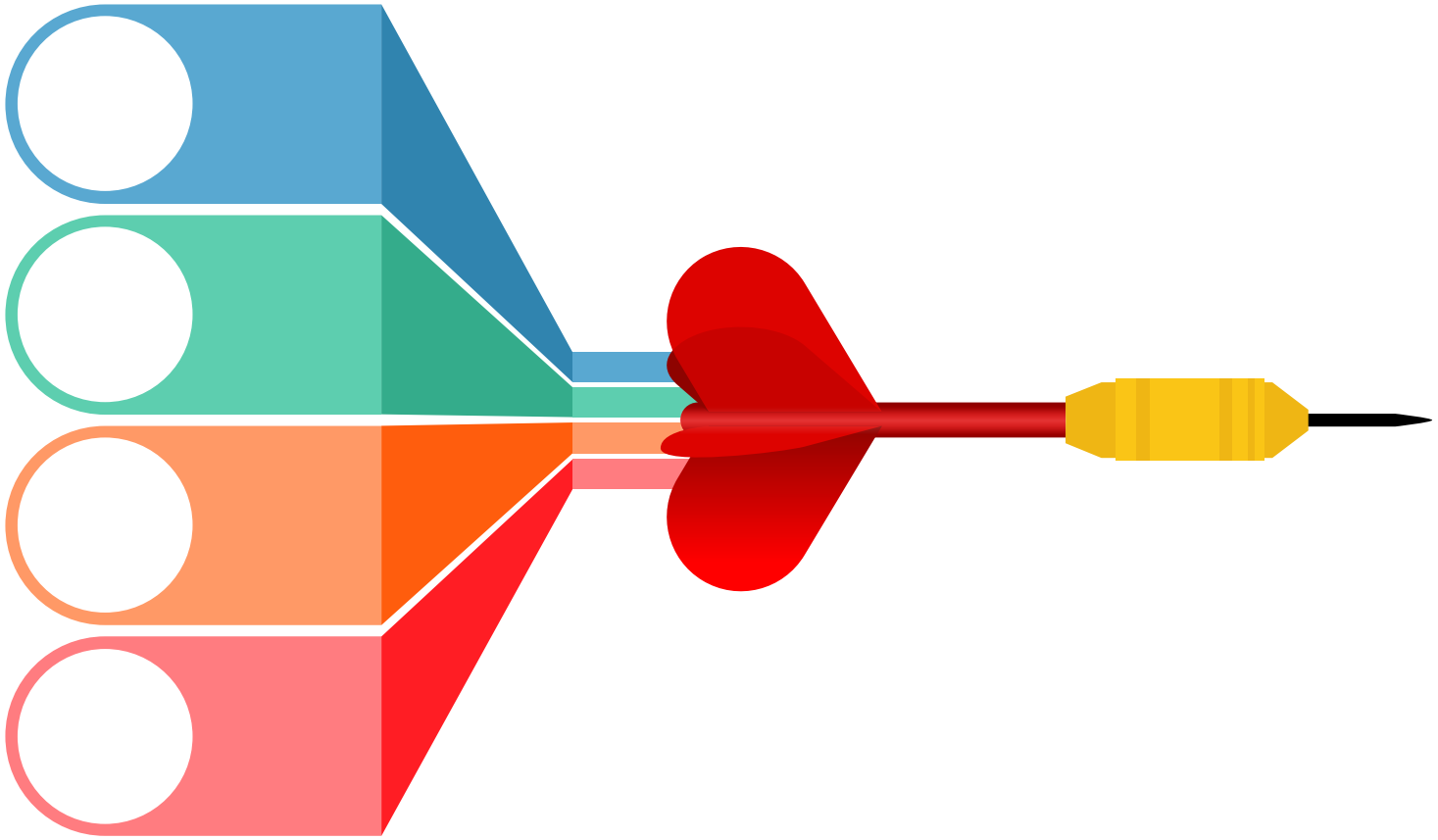
Parameters in the query string of the endpoint, after the ?.

Form

Parameters included in the request body. Usually submitted as JSON.

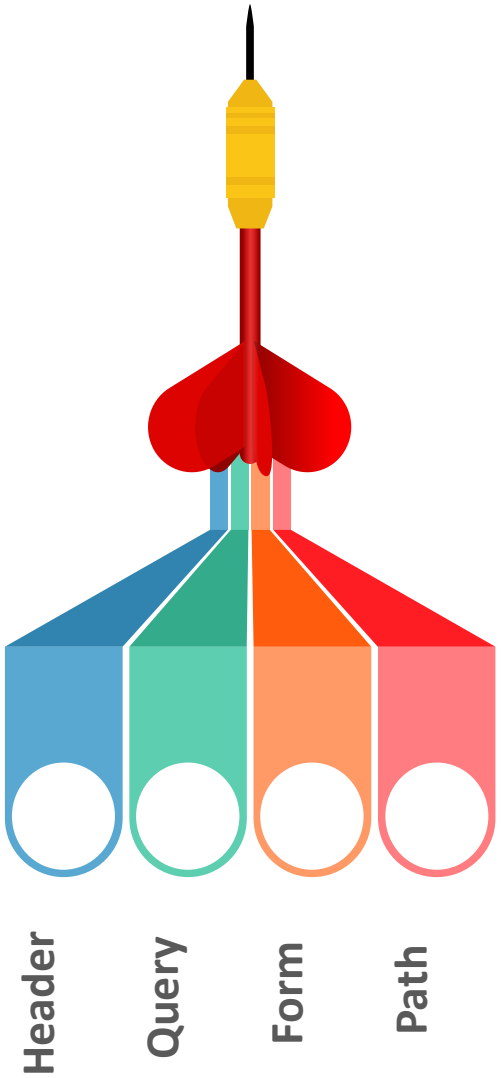
Path

Parameters within the path of the endpoint, before the query string (?). These are usually set off within curly braces.



Testleaf
Always Ahead

Request Parameters – Data Types



These data types are the most common with REST APIs:

- **string**: An alphanumeric sequence of letters and/or numbers
- **integer**: A whole number – can be positive or negative
- **boolean**: true or false value
- **object**: Key-value pairs in JSON format
- **array**: A list of values

Testleaf
Always Ahead

Request Body (Data Formats)

Form Data

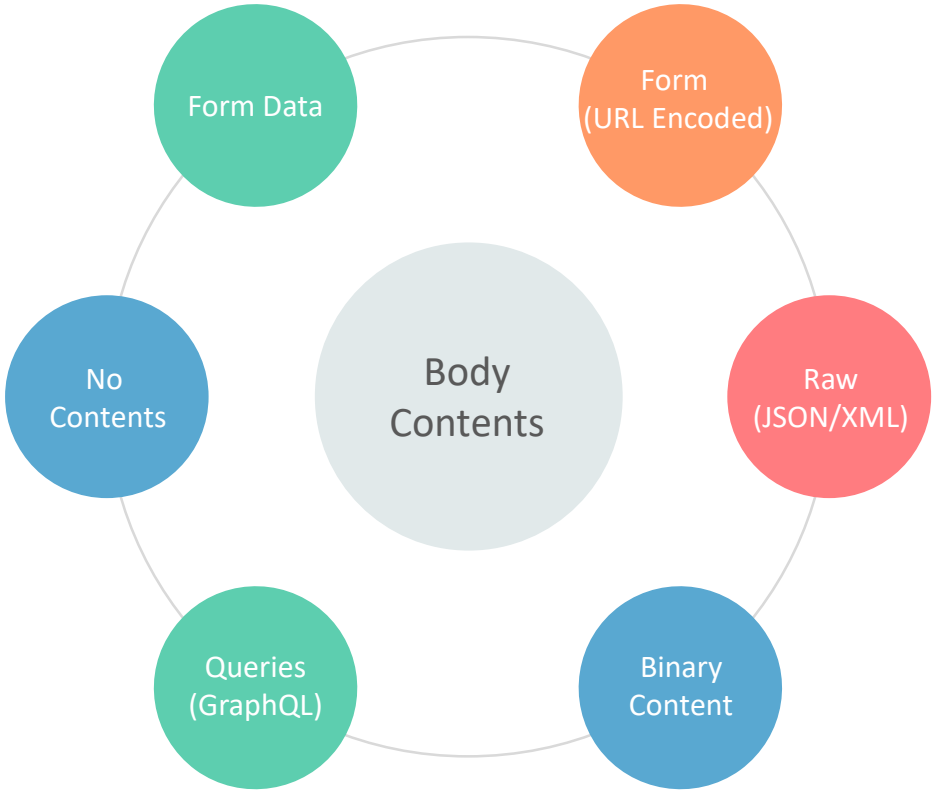
In Key-value format (Text / File) including multi-part

None

No data passed in body

Queries & Variables

This is specially for the GraphQL Scenarios



Form (URL Encoded)

Data is encoded in key-value format

Raw Data

This can be JSON/ XML or TXT

Binary

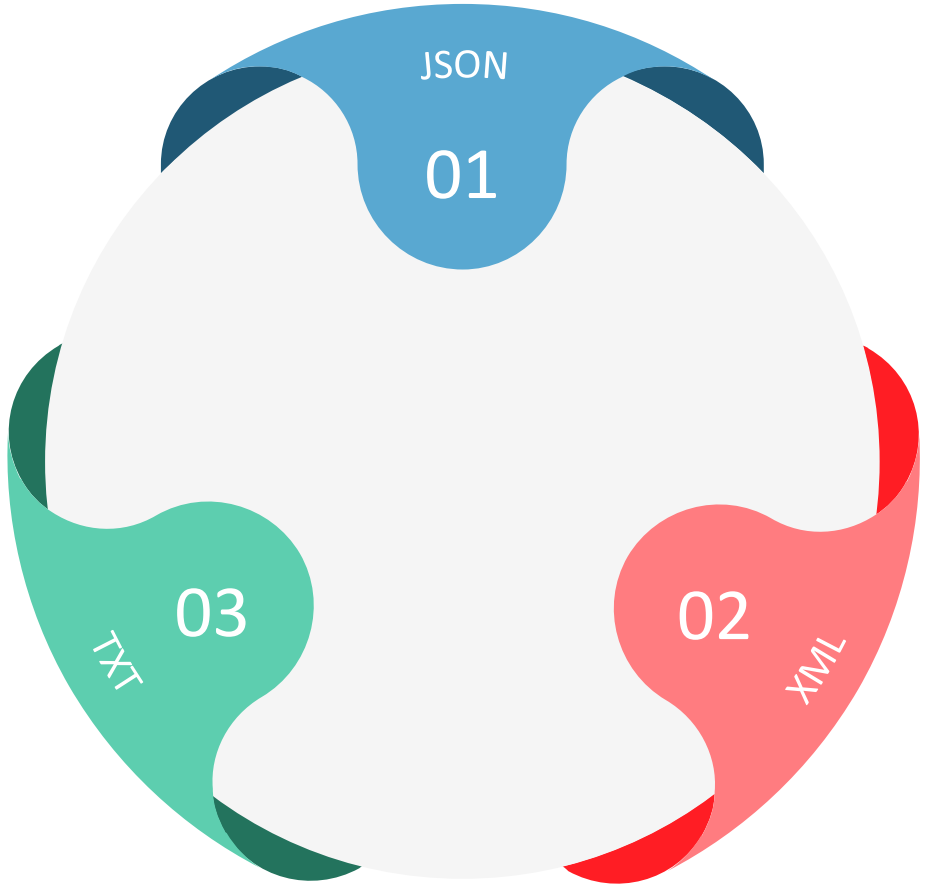
The files or executables that can be attached

Testleaf
Always Ahead

Body (Contents)

The body can have different formats like below:

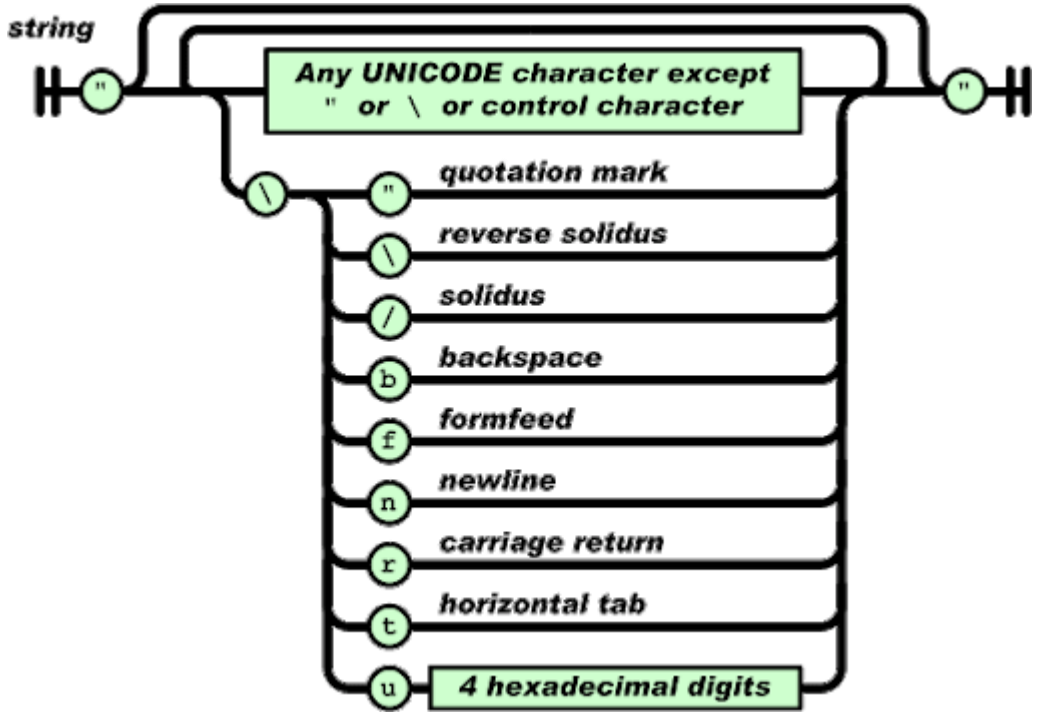
- JSON - JavaScript Object Notation
- XML - Extensible Markup Language
- TXT – Plain text format



Testleaf
Always Ahead

JSON – A quick run through

- JavaScript Object Notation
- Lightweight
- Easy to understand
- Not specific to languages



Testleaf
Always Ahead

JSON Structure

Simple JSON format

```
{
  “name”: "Babu",
  “age”: 45,
  “city”: "Chennai"
}
```

Simple JSON Array

```
{
  “name”: "Babu",
  “age”: 45,
  “companies”: ["Syntel", "HCL", "HP", "TL"]
}
```

Testleaf
Always Ahead

JSON Nested Arrays

```
{
  "name":"Babu",
  "age":45,
  "companies":[
     {"name" : "syntel", "years" : "3.1", "customers" : 4, "country" : ["india", "usa"]},
     {"name" : "hcl", "years" : "1", "customers" : 1, "country" : ["india"]},
     {"name" : "hp", "years" : "13.2", "customers" : 18, "country" : ["india",
"usa","uk","sg"]},
     {"name" : "tl", "years" : "4.3", "customers" : 8, "country" : ["india", "usa"]}
   ]
}
```

Testleaf
Always Ahead

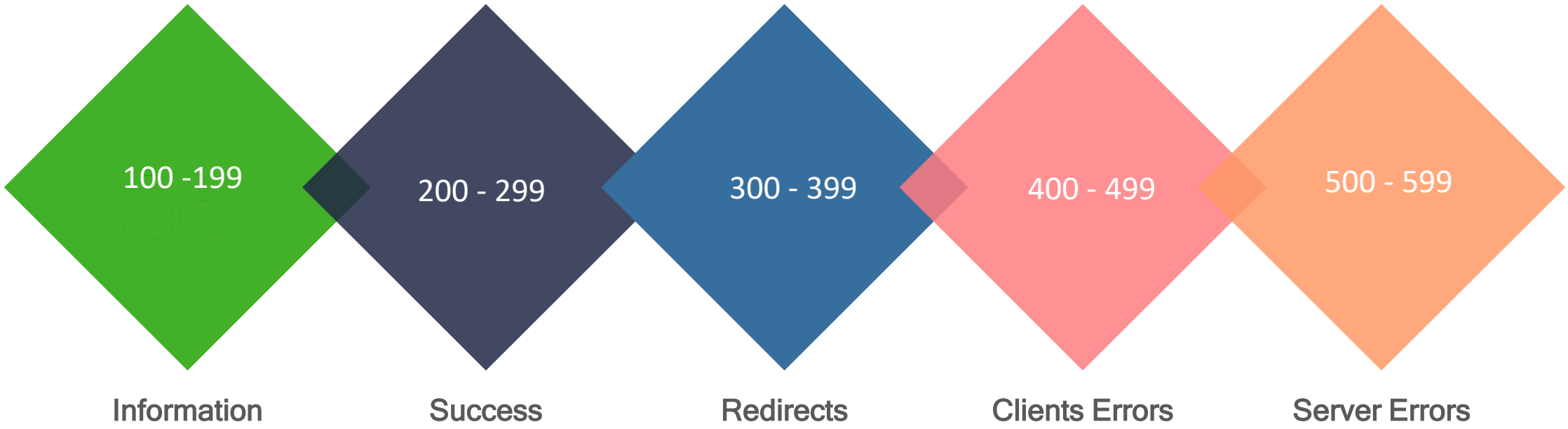
XML Structure

Simple XML

```
<?xml version="1.0" encoding="UTF-8"?>
<details>
    <person>
        <name>Babu</name>
        <age>45</age>
        <city>chennai</city>
     </person>
</details>
```

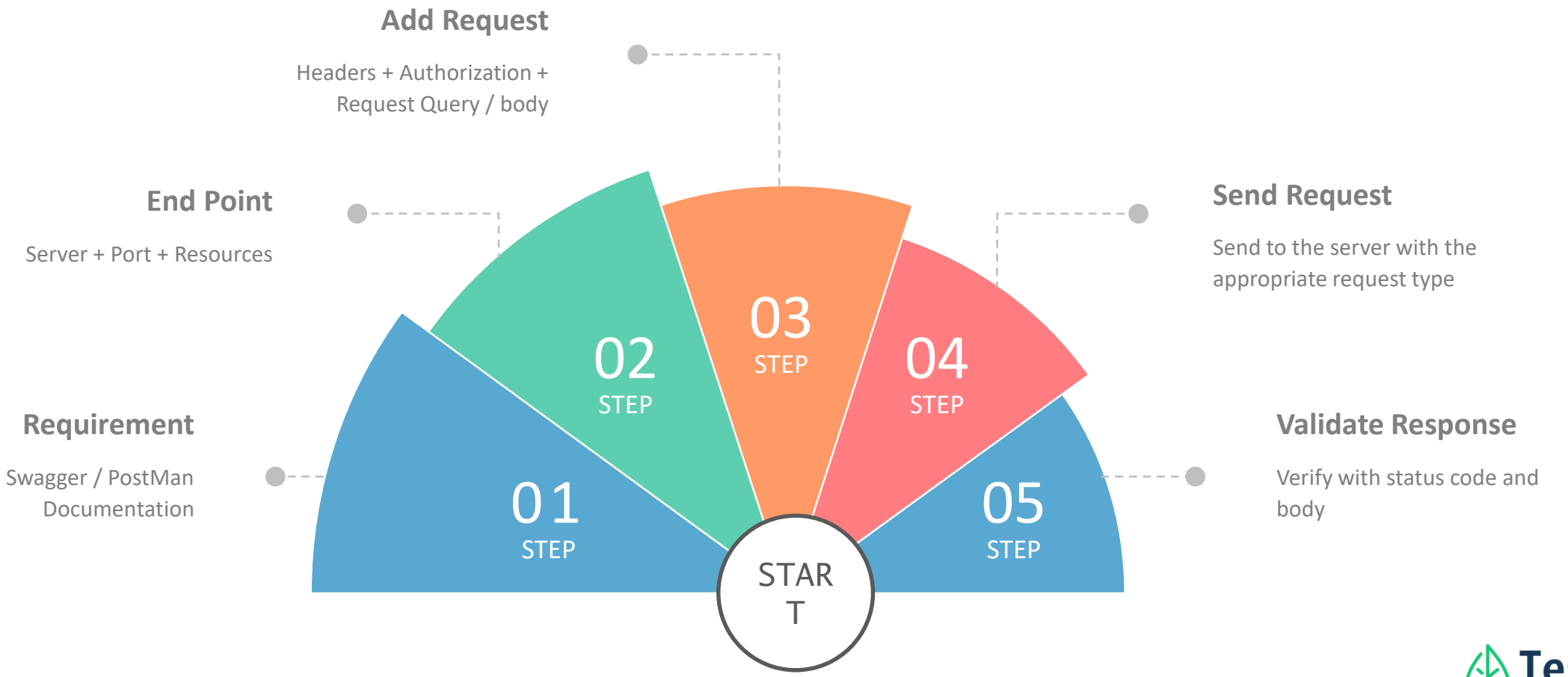
Testleaf
Always Ahead

Status Codes



Testleaf
Always Ahead

5 Step Process



Testleaf
Always Ahead

Thank You!

Testleaf
Always Ahead