



# REST API *vs* RESTful

---

# REST API

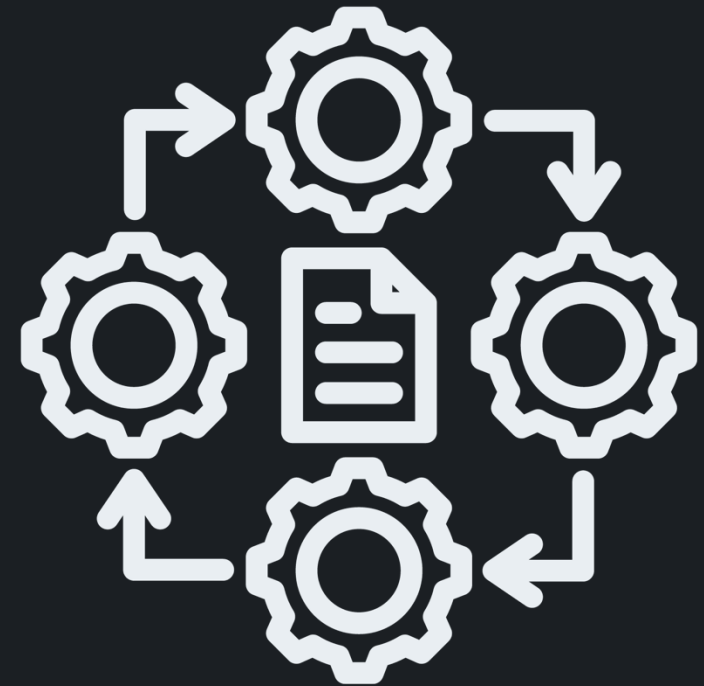
REST API is a client/server application that sends and receives data through the HTTP protocol using communication patterns such as XML and JSON and can be implemented in the desired language, allowing interoperability between services.



# REST vs RESTful

REST is an architectural model, that is, a set of patterns that bring good practices for the development of these APIs.

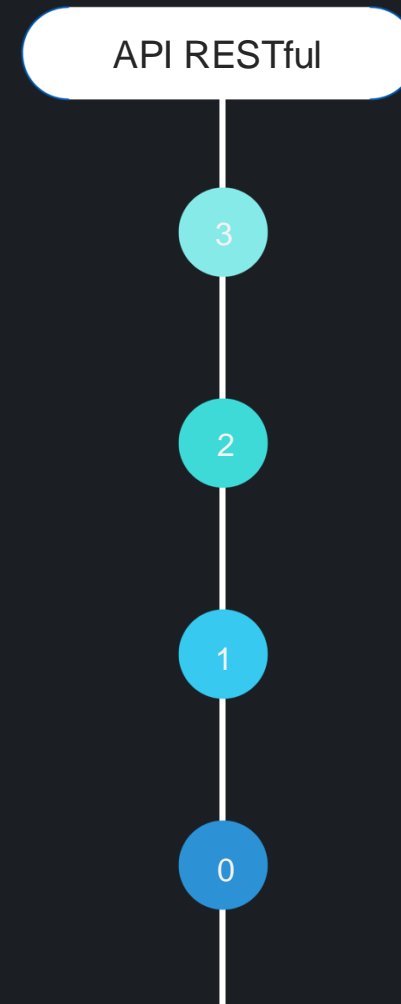
RESTful is the implementation of the REST architectural model in practice in some API. And to create an API that is considered RESTful, you need to know the REST architecture and also apply it correctly.



# Richardson Maturity Model

Leonard Richardson proposed a maturity model consisting of four levels for an API to be considered RESTful:

- Level 0
- Level 1
- Level 2
- Level 3



# API RESTful

3

LEVEL 3 - the API makes use of HATEOAS (hypermedia) showing its current state and also the relationship with other API resources

2

LEVEL 2 - the API uses the HTTP protocol semantically with its verbs, **GET POST DELETE PUT...**, and also the correct definition of returns according to each response

1xx: Information codes  
2xx: Success Codes  
3xx: Redirection  
4xx: Client error  
5xx: Server error

1

LEVEL 1 - the API has a well-defined resource mapping, with nouns and the correct use of URIs

0

LEVEL 0 or POX - the API uses the HTTP protocol only as a communication mechanism

# RESTful API in EAD Architecture

- RESTful API creation;
- Spring projects: Boot, Web, Data JPA, Validations, HATEOAS...
- Yaml files;
- Records;
- JsonView;
- Validations;
- Pagination;
- Exception handling;
- CORS;
- Dynamic filters with specifications;
- HATEOAS
- ...

