

## REST API for E-Commerce Site

Sneha Jadhav<sup>1</sup>, Anjali Singh<sup>2</sup>, Apurva Upadhyay<sup>3</sup>, Smita Avhad<sup>4</sup>, Prof. Pravin Hole<sup>5</sup>

<sup>1,2,3,4</sup>B.E. Computer Engineering, Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India

<sup>5</sup>Professor, Dept. of Computer Engineering, Terna Engineering College, Maharashtra, India

**Abstract** - In today's world with the help of ecommerce web design, you get an opportunity to have your products and services available to the customers 24 hours. It is a good exposure to your business. Most of the people prefer to shop online due to shortage of time, you can easily make more revenue and more profit. With the help of an e-commerce website people can select and buy desired products anytime anywhere. Payments can be done easily through credit cards or other payment options available in website. REST (Representational State Transfer) [1], an architectural style for web services, is getting more and more popular in recent years. Many vendors have opened the doors of their services to developers, providing them with restful accesses to different web services. A traditional e-Commerce website is growing in popularity. Large amount of product selection, competitive pricing and excellent customer care were drive more customers to the site.

**Key Words:** REST, e-Commerce

### 1. INTRODUCTION

REST (Representational State Transfer) is architectural style for designing loosely coupled web services. It is mainly used to develop lightweight, fast, scalable and easy to maintain web services that often use HTTP as means of communication. REST [2] is not connected to any particular platform or technology. REST defines Web as a distributed hypermedia application

RESTful applications use HTTP request to post data (create/update), read data (making queries) and delete data. Therefore, REST uses HTTP for all four **CRUD (Create/Read/Update/Delete)**.

The main concepts used in REST API are resources, verb and representation.

- Resources – Resources are fundamental interconnected elements on web platform. Every resource has a unique identifier known as URI (Universal Resource Identifier).
- Verbs are HTTP actions like GET, POST, PUT and DELETE.
- Representation determines a way to showcase these re-sources to clients. REST supports all formats like

JSON, XML or any other format without any restrictions.

The different components of REST API are:

- To support Routes for forwarding the supported requests (and any information encoded in request URLs) to the appropriate controller functions.
- Controller functions allow to get the requested data from the models, create an HTML page displaying the data, and return it to the user to view in the browser. Views is used by the controllers to render the data.

### 1.1 The MEAN Stack

A. Node.js: Node.js is an open source server environment and involves using JavaScript on the server. It uses asynchronous programming[3]. It can generate dynamic page content and runs single threaded.

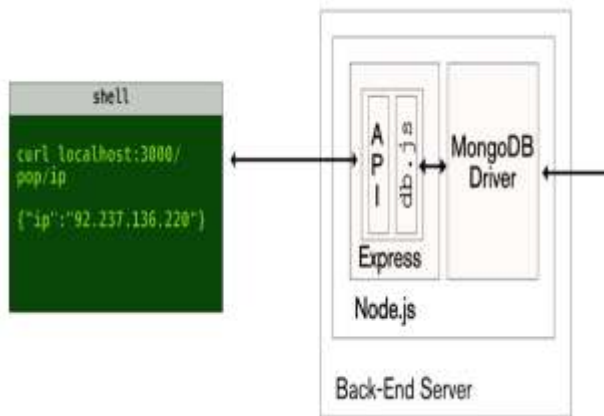
Features of Node.js:

- 1) Asynchronous and Event Driven: It is non-blocking
- 2) Very Fast
- 3) Single Threaded but Highly Scalable: Same program can provide many services to a much larger number of requests.
- 4) No Buffering: Never buffers any data and simply output the data in chunks.
- 5) License: Released under the MIT license.

B. Express.js: It builds on the underlying capability of Node. Allows HTTP operations such as GET and POST. It facilitates a simplified solution. Express application always uses a call back function whose parameters are request and response objects.

C. Mongo DB: It is a high performance NoSQL database [3]. It is built around the JSON data format. It provides dynamic schema. A record in MongoDB is a document, which is actually a data structure composed of field and value pairs. MongoDB documents are almost similar to JSON objects. The values of fields may include other documents, arrays, and combination of both.

D. Angular.js: It is an open source web application framework. For maximizing the performance it can be used with Yeoman and Bootstrap. It is a client-server framework which works for MVC. AngularJS expressions are always written inside double braces: `{{expression}}`.



### 1.2 Creating a REST API using PHP

We can actually create a simple REST API in PHP and call them in JavaScript using various technologies. For example, using **HttpRequest**, **jQuery AJAX** and **AngularJS AJAX**. Data is exchanged in JSON (JavaScript Object Notation) format. MySQL is the most used database system in the PHP world.

You can represent the database information in JSON format by using PHP. We can also provide security to the data by using various algorithms such as SHA.

### 2. Security

All the database data need to be safe. Security and authentication is necessary for a good ecommerce site. HTTP carries some inherited authentication mechanisms and it allows Basic, Digest, and Custom Authentication. For high security, one should implement HMAC or SHA.

### 3. MODEL VIEW CONTROLLER

MVC (Model View Controller) [4] is pretty straightforward. After creating a website, the following steps are followed by MVC:

- 1) Client sends REST keyword request to the server.
- 2) Server matches and reacts to the requested URL to the controller action.
- 3) Returns the result back to the client for processing in JSON format.

The generated JSON in the fourth step can be thought of as a 'view'. It is a representation of state, suitable for crossing process boundaries.

MVC is a way of structuring server side code. REST is concerned with the communication of clients and servers.

### 4. Architectural Constraint

A. Uniform interface: Decide interface for resources inside the system which are exposed to API consumers.

B. Client-server: Client application and server application MUST be able to evolve separately without any dependency on each other.

C. Stateless: Make all client-server interaction stateless. Server will not store any data about the latest HTTP request client made. It will treat each and every request fresh. There should be No session, no history.

D. Caching of data and responses is of utmost important. It brings performance improvement for client side, and better scope for scalability for a server because the load has reduced.

E. Layered system: It allows you to use a layered system architecture.

F. Code on demand (optional): You will be sending the static representations of resources in form of XML or JSON but when you need to, you are free to return executable code to support a specific part of your application.

### 5. Soap Vs REST

REST[5] operates through a consistent interface to access named resources. It's most commonly used when you are exposing a public API over the Internet. SOAP exposes components of application logic as services. It operates through various interfaces. REST accesses data and SOAP performs operations through a more standardized set of messaging patterns.

SOAP was created by Microsoft, and it's been a lot longer than REST. REST helps to access web services in a much simpler way than possible with SOAP by using HTTP. REST allows more variety of data formats. REST offers better support for clients.

REST provides efficient performance. REST is used most often for major services such as Yahoo, Amazon, etc. REST is faster and uses less bandwidth.

### 6. Scope of REST API

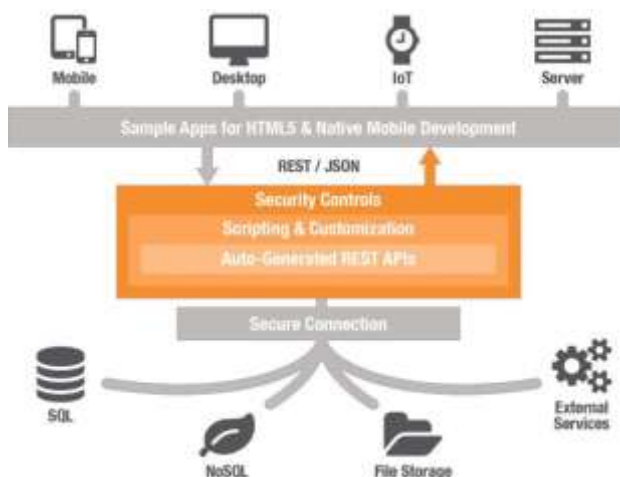
With can develop modifiable and system to accommodate growing business and need of people. The system built will be resistant to failure. Useful in cloud application.

Stateless components can be freely redeployed if something fail [6].

### 7. Architecture of REST API

Architecture considered RESTful or REST-style is characterized by:

- 1) State and functionality are divided into distributed resources.
- 2) Each and every resource is actually uniquely addressable using a uniform and of course minimal set of commands (typically using HTTP commands of GET, POST, PUT, or DELETE over the Internet)
- 3) Protocol is client/server, layered, and supports caching.



### 8. Advantages of REST

- a. REST is any interface between systems using HTTP to obtain data and generate various operations on the data in all possible formats, such as XML and JSON.
- b. Popular alternative to other standard data exchange protocols such as SOAP.
- c. Uses a simpler data-processing solution.

### 9. REST API features

1. Derives functionality for retrieving files from File fields in (RDOs) [7].
2. Supports create, read, update, delete, and query working.
3. Provides secure access to resources using flexible authentication methods, such as HTTP(S).
4. It is Language independent made possible by adherence to RESTful architectural standards, providing you with the

option to choose a programming language based on application requirements.

5. It includes implementation of standard HTTP methods and JSON resource representations.

### 10. Restful Concept

REST API follows standard architectural principles defined by Representational State Transfer [9]. It supports all these key architectural concepts:

1. **Stateless** – The REST service doesn't use login sessions or store other state information on the server as well.
2. **Communicates using HTTP** –Uses common HTTP methods:
  - a. GET: reads data
  - b. POST: creates resources
  - c. PUT: updates resources
  - d. DELETE: removes resources

3. **Uses HTTP status codes:** Status codes represent many results of operations that you can perform against the REST service.

4. **Manipulation of resources:** Rest API service represents the objects. A unique URL identifies each resource. Other entities are not generally identified as objects are also represented as resources.

5. **Providing a hypermedia-driven API** – The REST API helps to return responses that include links to resources available in Relativity. Considering the example: response for a workspace query returns a list of each matching workspaces. It helps in representing each workspace in a simplified format that contains a link to further details about the resource.

### 11. Unique implementation details

REST API [10] accommodates the data model for processing and scaling, while this also supports common RESTful principles. Consuming the REST API [11], consider the following requirements that are unique:

1. **Using paged lists of data:** REST API requires clients to search and page through all search results. Initially make a request, the number of resources that will be returned can't be determined immediately, so all lists of data need to be paged to support large result sets.

2. **Specify additional data fields to return** – REST API returns the only default set of fields for a resource unless

and until you provide a query string that specifies additional fields or all fields of the specific object.

You can improve performance by retrieving only default.

**1. Use queries to help for filtering data** – You are enabled to filter a list of resources by performing a query, which is a request and that uses the POST method and contains conditions specified specially in JSON representation. Then, REST API returns the search result resource, you can even page through the list of resources returned. You usually can't use a query string for filtering a collection of resources.

**2. Providing required headers** – You should include all the authentication credentials and an empty CSRF header specially in every request.

The following diagram defines the interconnections between REST & Services APIs.

## 12. Authentication of REST API

1. Relativity REST API provides you the ability for choosing an authentication method and that best fits your environment including application requirements. For providing secure communication between a client & the Relativity service endpoint, it supports basic authentication over HTTPS.

2. Authentication header is usually required for all calls to the REST endpoint. Authorization field in the HTTP header is better used to pass user credentials.

3. REST API uses JSON as a firm format for representing DTOs returned by HTTP methods [12] from the Services API. A JSON representation includes properties that describe a resource or collection of resources, a self link to the resource (specified in the `_Location` property), and links for pagination as required.

4. Several HTTP operations on a single resource return its JSON representation. Whenever multiple objects are returned by an HTTP operation the JSON representation includes Results property along with an array of resources.

5. Services API always supports the development of customized end-user applications. It usually simplifies the development process and provides object classes and other data structures that are considered as the building blocks for custom applications.

## 13. CONCLUSION

RESTful engine is mainly used to develop lightweight, fast, scalable and easy to maintain web services that often use http as means of communication. We have seen implementation of RESTful [13] web services by

developing an Ecommerce site. Internet is the most popular platform for business, many companies are setting their businesses online and Ecommerce is the most popular among them. Therefore, the system is developed to build a scalable REST API for Ecommerce to serve any frontend client.

## References

[1]<https://www.techiediaries.com/php-rest-api/>

[2][https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp)

[3]<https://docs.mongodb.com/manual/introduction/>

[4]<https://www.c-sharpcorner.com/blogs/how-to-make-rest-api-in-net-framework-mvc>

[5]<https://softwareengineering.stackexchange.com/questions/324730/mvc-and-restful-api-service>

[6][https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)

[7]<https://restfulapi.net/rest-architectural-constraints/>

[8]<https://dzone.com/articles/differences-in-performance-apis-amp-more>

[9][https://www.google.com/search?rlz=1C1CHZL\\_enIN755IN755&biw=1366&bih=608&tbm=isch&sa=1&ei=Az1HXLvKbIL28QWLubaQBw&q=REST+api+concepts+using+mongo+DB&oq=REST+api+concepts+using+mongo+DB&gs\\_l=img.3...1449973.1455834..1456261...0.0..0.286.2950.0j7j8.....0....1..gws-wiz-img.....0i24.5PcJmKI3fdE#imgrc=1W2zKqYPBeTErM:](https://www.google.com/search?rlz=1C1CHZL_enIN755IN755&biw=1366&bih=608&tbm=isch&sa=1&ei=Az1HXLvKbIL28QWLubaQBw&q=REST+api+concepts+using+mongo+DB&oq=REST+api+concepts+using+mongo+DB&gs_l=img.3...1449973.1455834..1456261...0.0..0.286.2950.0j7j8.....0....1..gws-wiz-img.....0i24.5PcJmKI3fdE#imgrc=1W2zKqYPBeTErM:)

[10][https://www.google.com/search?q=architecture+of+rest+api&rlz=1C1CHZL\\_enIN755IN755&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjzh8HkyoHgAhXKuY8KHb2qAJIQ\\_AUIDigB&biw=1366&bih=608#imgrc=d1IU9HsnHv8HfM:](https://www.google.com/search?q=architecture+of+rest+api&rlz=1C1CHZL_enIN755IN755&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjzh8HkyoHgAhXKuY8KHb2qAJIQ_AUIDigB&biw=1366&bih=608#imgrc=d1IU9HsnHv8HfM:)

[11] Research paper: On Composition of SOAP based and RESTful Services. 2016 IEEE 6th International Conference on advance Computing.

[12] Research paper: Using Mean Stack to implement a RESTful Service for an Internet of things application.

[13] Infosys white paper: Best practices for building restful web services.