APIs and Restful APIs

Mithilesh Tarkar[1], Ameya Parker[2]
[1]Student, [2]Professor
Swami Vivekanand Institute of Technology (VESIT),
Chembur, Mumbai, India

ABSTRACT

This Research Paper is about Restful API and Rest APIs. This Paper Presents Description about how APIS Restful APIs work and how they are implemented and why they are used in the modern world.

Keywords: Restful APIs, APIs

INTRODUCTION

Restful APIs and APIs have become an essential tool of backend development. Without APIs , there will be a lot of security issues in the system. The Modern Mobile and Web Development world essential rely on Restful APIs and APIs for authentication, Fetching and Submitting Data.

A Good API makes it easy for the developer to work with, and helps to create really powerful applications. Design of an API has to be in such a way that, the user should not have anything visible in terms of the complex structure of the API.

The Entire Research Paper explains 1) What are APIs2) How are they used. 3) What are the required technologies to create APIs. 4) Why are they important and must use.
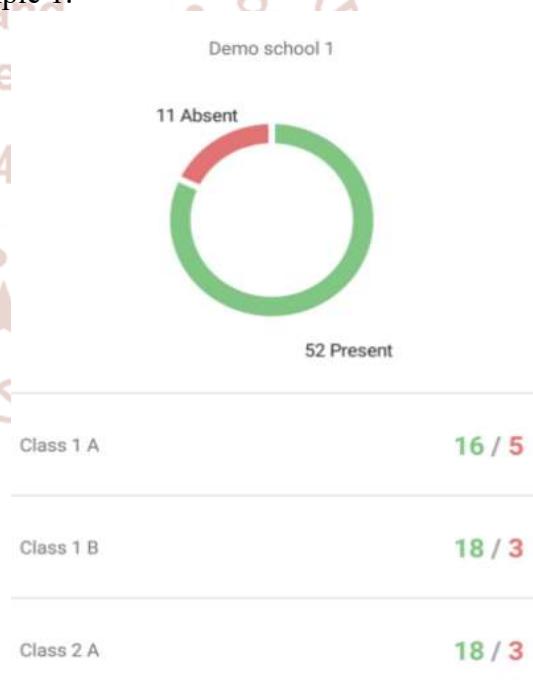
RESEARCH ELABORATIONS

After doing a lot of research work. I can represent the following elaborations. API - standings for Application Programming interfaces. APIs are helpful to have a centralized backend system for a mobile and a web application. For E.g.: if a college has website which registers students in its database. And the college wants to develop a mobile app , to register the students in their database. However the college is already using a Database for the website. So they in

order to keep using the same database for the mobile application, as well as for the website. An API implementation can be used to solve this problem.
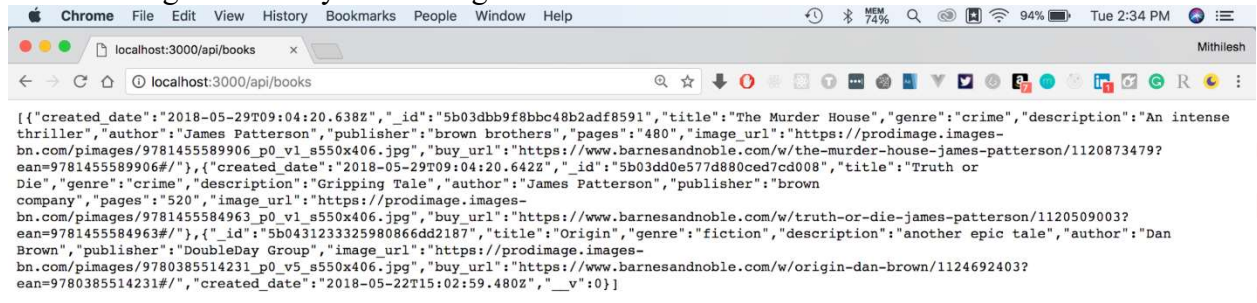
With an API, the Developer only has to design the front end of the mobile application. The backend of the application will be the same as that of its website. The Developer has to do API Calls to fetch the data, display the data, push the data and to delete the data. In My Research i also found out that, API are also used for live data updates. Where there is a pie chart of attendance.

Example 1:

*The Above Figure shows the attendance of Class 1 A, Class 1 B and Class 2 A   , where the data is updated whenever the attendance is taken through a live API.
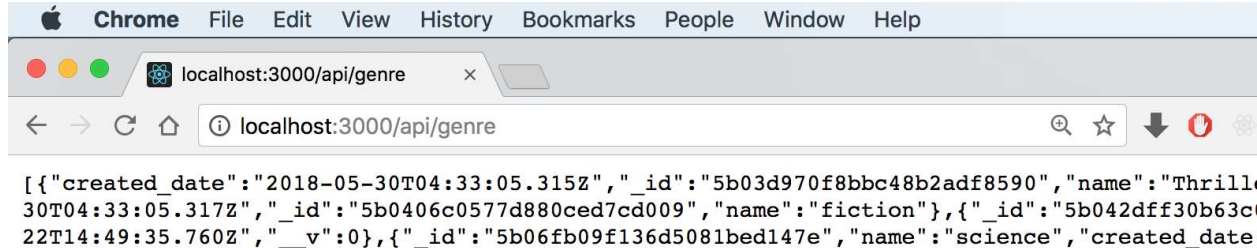
Example 2: Fetching JSON array from mongodb Database.

* The above image , shows a database collection of books being fetched from mongodb database , by calling the above URL.
* This data can then be used to display on the front end of the website/mobile application.

Code Snippets :

Creating a MongoDB Schema -

```javascript
var mongoose = require('mongoose');

// Create Genre Schema..
var genreSchema = mongoose.Schema({

    name:{
        type: String,
        required:true
    },
    created_date:{
        type:Date,
        default:Date.now
    }

});
```

Writing the Functions :

```javascript
var Genre = module.exports = mongoose.model('Genre',genreSchema)


// Get Genres Function..

module.exports.getGenres =   function(callback,limit){

        Genre.find(callback).limit(limit);



}

// Post a Genre...
module.exports.addGenre = function(genre,callback){

        Genre.create(genre,callback);

}
```

Calling the Function from the Server main file:

```javascript
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
var mongoose = require('mongoose');

app.use(bodyParser.json());
Genre = require('./models/genre');
Book  = require('./models/book');

//connect to mongoose.
mongoose.connect('mongodb://localhost:/bookstore');
var db = mongoose.connection;

app.get('/',function(req,res){


        res.send('Please use /api/books or /api/genres');

});
app.get('/api/genre',function(req,res){

        Genre.getGenres(function(err,genres){

                if(err){

                        throw err;
                }
                res.json(genres);

        });

});
app.post('/api/genre/add',function(req,res){
        var genre = req.body;
        Genre.addGenre(genre,function(err,genre){

                if(err){
                        throw err;
                }
```

Simulation Softwares Used :
- RestEasy Plugin , for Google Chrome.
- Postman IDE

RESULTS OR FINDING

For research purposes we modelled a books api and did testing, where we are trying to put, push and delete the data in the database. If we did not query the API correctly, it showed error code. If the query was written correctly, then it showed status ok, and inserted the data into the database. There was no rollback, or dead transactions in this. Either the Transaction was accepted or it was rejected.

There is this solid security in the API calls, where if even a single part of the query is wrong, nothing will work properly.

CONCLUSIONS

Thus, we can conclude the by using API Technology, we can make a very strong back end , and it can be scalable and the data can be used to render on a mobile device or even a Smart TV. Thus, helping to separate the front end environment and backend environment completely.

BIOGRAPHY

Mr Mithilesh Tarkar l is currently a final year M.C.A student of Vivekanand Education Society's Institute of Technology (V.E.S.I.T), Mumbai. He has completed his B.COM graduation in the year 2013 from R. A Podar College of Commerce and Economics, Mumbai, Maharashtra, India .He has an abiding interest in Full Stack Development, Graphic Designing, UI/UX Designing.