

Creation of Multipurpose API

Kiran M L¹, Mamatha Balipa²

^{1,2}Dept of MCA

^{1,2}NMAMIT, Nitte

Abstract- An application software interface (api) is a fixed of workouts, protocols, and gear for building software program packages. essentially, an api specifies how software additives must engage. moreover, apis are used while programming graphical user interface (gui) components.

A server-side internet api is a programmatic interface consisting of one or greater publicly uncovered endpoints to a defined request–response message system, normally expressed in json or xml, that is exposed via the internet—most usually via way of an http-based internet server.

Enterprise people use apps. Api's are essential for commercial enterprise because they allow programmers to build extremely good tools that assist us do our jobs greater efficiently. A very good instance is this key-word device that accesses google's search api to indicate keywords your enterprise should target.

Keywords- OAuth2, JWT, REST, API

I. INTRODUCTION

The main purpose of the project is to provide API to the end user who are also termed as modern developers who do not spend their time for creating simple and commonly used code for their use; instead they can make use of readily available API Modules created in our project by us, so that they can use those API's for their various purposes. Most of the WEB API's created in the project are relevant to Gamification API Creation project because those API's that are created are mainly developed for Gamification API creation itself. Ex: most commonly in all the login system for any site or project ,same or similar style of login checks are done, almost the validations are also similar and sometimes the same in common. In such case the new generation developer can make use of the ready -made API's for their project just by including our module or the API created by us. Since [1] WCF is Cleanser based, which utilizes standard XML outline over HTTP, it could prompt slower performance. WEB API is a superior decision for easier, light weight services. WEB API can use any content arrangement including XML and is quicker than WCF.WEB API can be utilized to make all out REST Administrations.

Web Services are in view of the Cleanser protocol. Web API is a more current Microsoft structure which causes you to manufacture [2] REST based interfaces. The reaction can be either JSON or XML, yet there is no real way to create customers naturally in light of the fact that Internet Programming interface does not offer a service description like the WSDL from Web Administrations.

Utility program interface (programming interface) is an arrangement of schedules, conventions, and apparatuses for building programming programs. An api specifies how programming segments should cooperate and apis are utilized when programming graphical ui (gui) segments.

Why we choose WEB API?

- It depends on HTTP and simple to characterize, uncover and devour in a REST-ful way.
- It is lightweight design and perfect for gadgets that have restricted transfer speed like cell phones.
- When similarity is worried ,there is no much issue on utilization of web Programming interface.

A Web API controller action could return following values:

- Void – It would return empty content.
- HttpResponseMessage – It would convert the response to an HTTP message.
- IHttpActionResult – Internally calls ExecuteAsync to create an HttpResponseMessage.
- Other types – You can write the serialized return value into the response body.

The present day functions of asp. net api framework v2.0 are as follows:

- characteristic routing
- go-starting place resource sharing
- outside authentication
- open web interface internet
- httpactionresult

- internet api odata

/createNewProduct
/deleteAllProducts

Difference Between WEB API Routing and MVC Routing.

There should be atleast one route defined for MVC and Web API to run MVC and Web API application respectively. In Web API pattern we can find “api/” at the beginning which makes it distinct from MVC routing. In Web API routing “action” parameter is not mandatory but it can be a part of routing.

Since we have more concentration on the Gamification API, the descriptions are also commonly stressed more on Gamification API.

Web API Creation

Naming Conventions for Controllers and Actions

1. Use plural nouns for Controllers

Don’t mix up singular and plural nouns. Keep it simple and use only plural nouns for all resources.

Example:

/products instead of /product
/users instead of /user

2. Use nouns but no verbs for Actions

Don’t use fully described names for actions. It’s a bad practice.

	GET	POST
RESOURCE	Read	Create
/Products	Returns a list of products	Create a new product
/Products/101	Returns a specific product	Method not allowed (405)

Fig. 3 GET, POST operations

Don’t use verbs:

/getAllProducts

Don’t change the state of HTTP Methods

HTTP methods have their own state of behavior. Don’t change or mix their state.

	GET	POST	PUT
RESOURCE	Read	Create	Update

Fig. 2 GET, POST operations

Example for violated HTTP method rules

- Get Method is used to read the data. Don’t use it to alter the state.
- Delete is only used to delete or inactivate the record.

3. Stability and Consistency

Let’s say, Your API is more successful and it has a lot of clients. Now, you have a requirement to rename one of the fields or add a new field for JSON response. What happens if you modify the changes and publish a web API? Everyone that’s already integrated with you is going to break.

In the software world, we can’t blame changes or new feature requests. They are part of the software development. We can’t stop them or refuse them either.

The most common way of handling changes is, to have versions of an API.

4. Versioning:

Changes that don’t break existing code using the API can be handled within one version. If there are breaking changes i.e. it risks breaking the code which is using the API, those changes should be introduced in a new version of the API.

Every developer should do some planning ahead of your web API. Make the API Version mandatory and never release an unversioned API.

Sample of URL version schema:

/api/v1/products

```
/api/products?version=v1
/api/products?api-version=1
/api/v2-Alpha/products
/api/products?api-version=2-Alpha
```

```
/api/v2015-05-01.3.0/products
/api/products?api-version=2015-05-01.3.0
```

5. Flexibility

Flexibility is most important for Web API's. Let's say, if you have a lot of users for your web API, users may need output as JSON format or XML format as per their business needs. So, you can't say that my API will always return output as JSON. If you say that, you'll lose your business.

How will you manage that? How should your API act flexible (about input, output)? Here is the answer; you can handle it through URL resources. Please look at the below samples.

5.1 Flexibility on Input:

By default, Web API will support multiple formats of input types. Those are text/html, JSON, XML, etc.

5.2 Flexibility on Output:

The below formats are giving flexibility to end users to choose their output format.

Example Resources:

```
/api/v1/products.Json
/api/v1/products.XML
/api/v1/products/?format=Json
/api/v1/products?format=XML
```

5.3 Flexibility on Filtering:

Use a unique query parameter for all fields or a query language for filtering.

GET /api/v1/products?price>1000 Returns a list of products(Id, Name, Description, Code, Price), which price is greater than 1000

GET /api/v1/products?code=P123 Returns a list of products(Id, Name, Description, Code, Price) which code is "P123"

If you provide multiple filters, you can only return resources that match all filters.

5.4 Flexibility on Field selection:

The above get request, gives a list of products along with all fields (Id, Name, Description, Code, Price). Let's say, your API will be consumed by multiple users. Those users may be desktop, tablet, or mobile developers. Desktop users need all fields of products, tablet users need some fields and mobile users need only few fields. They don't need all attributes of a resource all the time.

So, give the API consumer the ability to choose returned fields. This will also reduce the network traffic and speed up the usage of the API.

GET /api/v1/products?fields=name,price

It'll give you only list of products which contains name and price, not other fields (Id, description and code)

6. Pagination

Paginate your API requests to limit response results and make them easier to work with. We use offset and count in the URL query string to paginate because it provides greater control over how you view your data.

```
/api/v1/products?offset=0&count=10
```

7. Sorting

Allow ascending and descending sorting over multiple fields.

GET /api/v1/products?sort=-price
- Retrieves a list of products in descending order of price

GET /api/v1/products?sort=+price,name
- Retrieves a list of products in ascending order of price and name

8. Security

Security is one of the most important and crucial parts in web API. Web API provides information about your functionality of business. Here, there is a chance hackers will misuse your API or steal your business information.

Hence you need to make sure, your methods should be accessible to only authenticated users.

There are different types of authentication in web API.

- Token-based authentication, where the token is a random hash assigned to the user and they can reset it at any point if it has been stolen. Allow the token to be passed in through POST or an HTTP header.

OAuth 2

[4] OAuth is a protocol; the current version is OAuth 2.0.DotNetOpenAuth which provides implementations of both OAuth 1 and OAuth 2 and it's used to secure a .NET Web API.

oauth flow starts where the registered consumer will gift username and password to a selected end point, and the api will validate those credentials, and if everything is legitimate, it'll return a token for the consumer where the customer utility used by the user need to keep it securely and regionally a good way to gift this token with every request to any blanketed cease factor.

9. Token Generation

Choose a secure token, not a short numeric identifier or random string. Something irreversible is best.

Generate token by using [5].JWT. JWT is lightweight and it has pay load. You can choose algorithm type. Generate token by using [6] SHA1 algorithm.

10. HTTP Status Codes

We Use the standard HTTP status codes to return controller actions.

Example:

400 – Bad Request – The request was invalid or validation failed.

401 – Unauthorized – The request requires a user authentication.

500 – Internal Server Error - Some exception or error occurred in your action. So use 500 for failures.

200 – OK – Everything is good.

201 – OK – New resources has been created

202 – Accepted – Request is accepted and it's good for processing.

II. CONCLUSION

the usage of of api's within the gift global where extra importance is given to time element, the api which might

be already made available for use at once of their numerous desires makes green use of time in addition to price cost. but, earlier than any sizeable progress and development can be made to the present practices and technology for net apis, we want to reach a deeper knowledge of how apis are developed.

REFERENCES

- [1] WCF Programmer's Reference: Windows Presentation foundation with C# 2010 and .NET 4 1st edition By Rod Stephens.
- [2] RESTful Web Services Cook Book By Subbu Allamarju.
- [3] UDDI ,SOAP and WSDL: The WEB Services By Aaron E. Walsh.
- [4] Mastering Oauth2 By Charles Bhis.
- [5] Introducing the Auth0 JWT Handbook By Sebastian Payrott.
- [6] US Secure Hash1 Algorithm By D. EastLake.