**What is Web API?**

API is an abbreviation for Application Programming Interface. A [Web API](https://www.scholarhat.com/tutorial/webapi/what-is-web-api-and-why-to-use-it) is a Web application programming interface. A Browser API can be used to enhance a web browser's functionality. A Server API can be used to enhance a web server's functionality.

**Why is Web API required?**

A web API is a service that retrieves information or data from a server. It is critical for business growth. Here are some of the reasons why Web API is so important:

* Provides a data access interface for both web pages and client applications.
* Supports a variety of text formats, including XML and JSON.
* Works nicely on low-bandwidth devices.
* Commonly used in the creation of UI/UX, resulting in increased visitors to the website.
* Compatibility with any browser and device.
* It is used to develop simple HTTP Services that are not SOAP-based.
* It is also a simple way to create with Web API. REST Services with WCF
* It is HTTP-based and simple to define, expose, and consume RESTful.
* It is a lightweight architecture that is excellent for devices with limited bandwidth, such as smartphones.

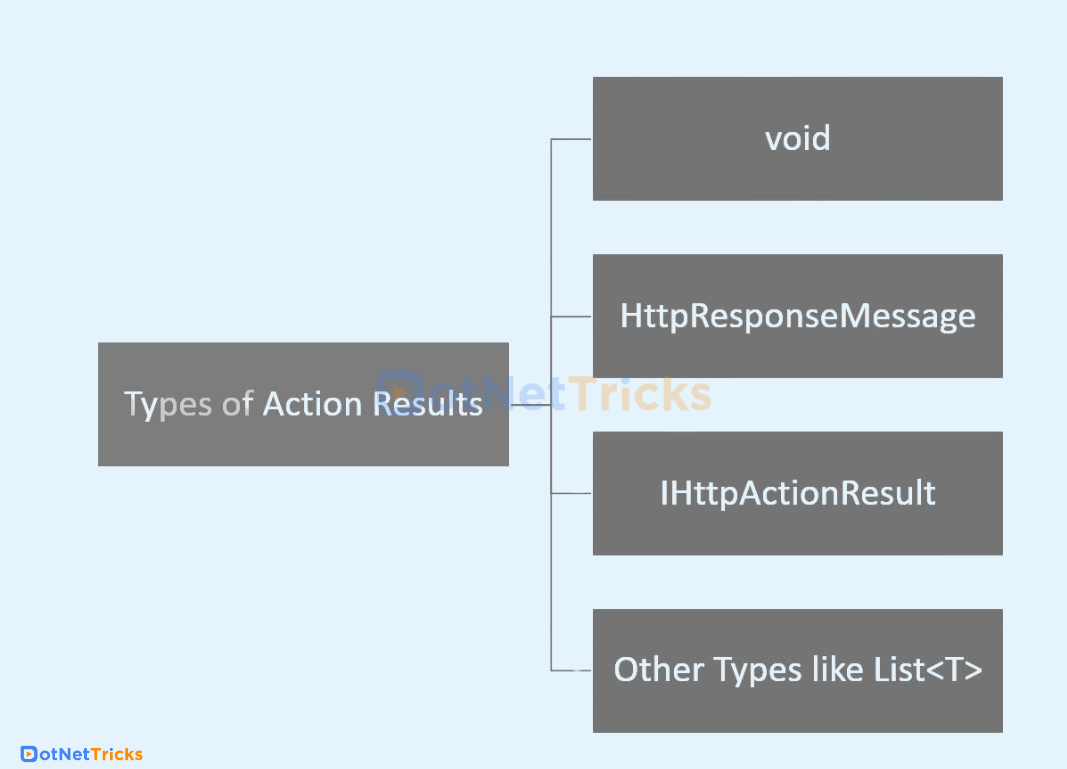
**What are the advantages of Web API?**

* **RESTful magic:** Uses conventional HTTP methods and lightweight data formats to simplify API design.
* **Lightweight champion:** Runs quickly with little overhead, making him excellent for mobile and web clients.
* **Flexible companion:** Works well with a variety of formats (JSON, XML) and hosts (IIS, self-hosted).
* **Developer delight:** It is simple to learn, configure, and test, which reduces development time.
* **Modern Muse:** Aligns with current online trends, allowing for easy future integration.
* **Scalability superstar:** seamlessly handles escalating needs, from modest projects to massive endeavours.
* **Community cheer:** Support and resources are provided by a thriving development community.

**What are the main return types supported in Web API?**

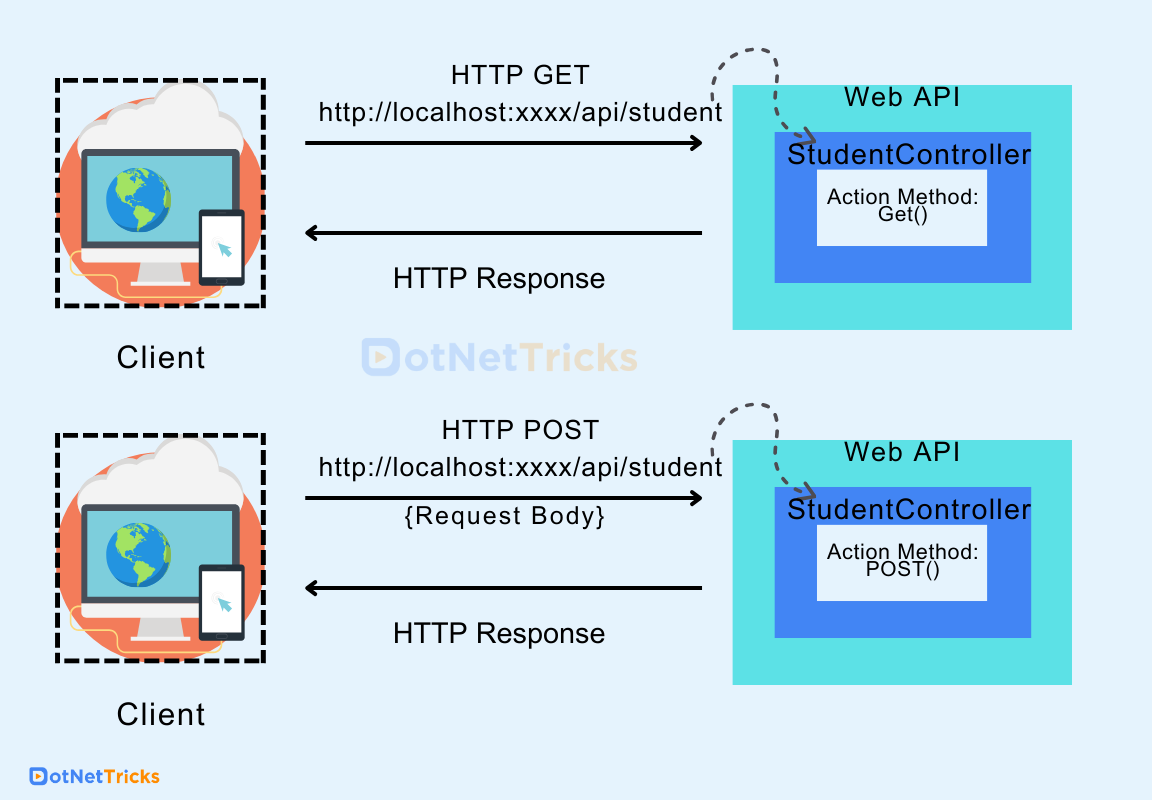
The following values can be returned by a Web API controller action:

* **Void** - It will return an empty string.
* **HttpResponseMessage** - The response will be converted to an HTTP message.
* **IHttpActionResult** - invokes ExecuteAsync internally to generate a HttpResponseMessage.
* **Other types** - The serialized return value can be written into the response body.



**What is Web API Routing?**

[Web API Routing](https://www.scholarhat.com/tutorial/webapi/comparing-aspnet-web-api-routing-and-aspnet-mvc-routing) is the routing of incoming HTTP requests to the correct controller activities within a Web API application. It serves as a dispatcher, ensuring each request is routed to its proper destination.



**How Can we assign an alias name for ASP.NET Web API Action?**

To give an ASP.NET Web API action an alias name, use the [ActionName("alias")] attribute on the action method. If you're using custom routes, make sure your route template matches the alias.

Example

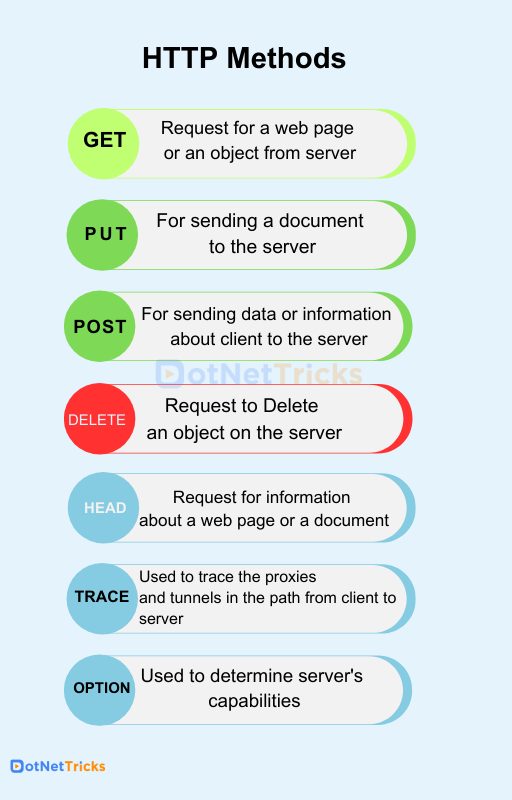
ActionName("GetAllProducts")]

**public** IEnumerable<Product> **GetProducts**() { ... }

**Explain the different HTTP methods.**

There are 8 HTTP methods:

* **GET:** Uses a specific URI to retrieve data from the server.
* **HEAD:** Like GET, but just sends the status line and header section.
* **PUT:** Replace all existing resources with the uploaded content and update them.
* **POST:** Send data to the appropriate server.
* **DELETE:**Deletes all current resources specified by a URI.
* **OPTIONS:** Specifies the target resource's communication options.
* **CONNECT:** Constructs a tunnel to the server based on a URI.
* **TRACE:** Runs a message loop-back test along the path to the destination resource.



**How are different HTTP Status Codes categorized?**

Here, All HTTP status codes are categorized into five classes. These include –

* 1xx (Informational) – It shows that the server has received a certain request and the process is continuing.
* 2xx (Successful)–It shows that the request was successful and accepted.
* 3xx (Redirection)–It shows that the request has been redirected and its completion will require further action or steps.
* 4xx (Client Error)–It shows that the request for the web page cannot be reached as either it is unavailable or has bad syntax.
* 5xx (Server Error)–It shows that the server was unable to complete a certain request even though the request seems valid.