ASP.NET Web API is a framework for building HTTP-based services, ideal for applications that need to expose RESTful endpoints for clients to interact with data and functionality over the web. Here are some key features ASP.NET Web API provides:

**1. RESTful API Development**

* ASP.NET Web API is designed to support REST principles, making it easy to create RESTful services. It maps HTTP methods (GET, POST, PUT, DELETE) to controller methods, allowing CRUD operations to be naturally implemented.

**2. Routing and Attribute Routing**

* It provides both conventional and attribute-based routing. With attribute routing, you can directly specify routes at the action level, allowing for more granular control and customization of route structures.

**3. Content Negotiation**

* Content negotiation automatically chooses the most suitable data format (e.g., JSON, XML) based on client preferences or the Accept header. ASP.NET Web API supports multiple formats, which can be extended to add custom media types.

**4. Model Binding and Validation**

* Web API supports model binding, allowing parameters in HTTP requests (query string, URL segments, request body) to be bound to complex types in the action parameters. It also integrates with data annotation attributes to perform server-side validation on models.

**5. Dependency Injection (DI)**

* ASP.NET Web API has built-in support for Dependency Injection, making it easy to inject services and repositories into controllers, promoting modular and testable code.

**6. Exception Handling**

* Provides a structured approach to handle exceptions, with features like Exception Filters, which allow custom handling of unhandled exceptions at the controller or global level. Additionally, IExceptionHandler and IExceptionLogger can be implemented to log and manage errors globally.

**7. Authentication and Authorization**

* ASP.NET Web API supports various authentication and authorization methods, including:
  + **OAuth and OpenID Connect** for token-based authentication.
  + **JWT (JSON Web Tokens)** for stateless authentication.
  + **OAuth 2.0**, **Bearer Tokens**, and **IdentityServer** integration for securing APIs.
  + **Role-based and policy-based authorization** to control access to resources.

**8. Filters (Action, Authorization, Exception)**

* Filters allow you to inject custom logic at different stages of the request pipeline:
  + **Authorization Filters** for authorization checks before executing actions.
  + **Action Filters** for pre- and post-processing logic around action methods.
  + **Exception Filters** for handling exceptions raised by action methods.

**9. OData Support**

* ASP.NET Web API has support for **OData** (Open Data Protocol), which allows clients to query data dynamically with built-in support for features like filtering, paging, sorting, and querying specific fields. This is especially useful for applications needing robust data query capabilities.

**10. Asynchronous Programming**

* It supports asynchronous programming using async and await, allowing developers to build high-performance, scalable APIs that handle large numbers of requests without blocking threads.

**11. Logging and Monitoring**

* ASP.NET integrates with **Microsoft.Extensions.Logging** and supports structured logging and monitoring with tools like Application Insights, making it easier to track, monitor, and debug application behavior.

**12. Caching**

* Supports various caching mechanisms, such as **output caching** (for caching responses) and **in-memory caching** (for storing frequently accessed data). This improves performance and reduces server load.

**13. Cross-Origin Resource Sharing (CORS)**

* ASP.NET Web API has built-in support for **CORS** to control cross-origin HTTP requests, allowing secure API access from different domains and ensuring compliance with browser security policies.

**14. Swagger/OpenAPI Integration**

* Using tools like **Swashbuckle** or **NSwag**, ASP.NET Web API can automatically generate **Swagger** (OpenAPI) documentation, providing an interactive interface for clients to test and understand the API endpoints, request models, and response types.

**15. Self-Hosting and Cloud Readiness**

* ASP.NET Web API can be hosted on IIS, within ASP.NET Core applications, or even **self-hosted** in a custom process, providing flexibility in deployment options. It’s also cloud-ready and works seamlessly with platforms like Azure.

**16. Versioning Support**

* It has built-in support for **API versioning**, which allows you to manage multiple versions of your API (e.g., v1, v2) within the same codebase. This ensures backward compatibility for clients that rely on older versions.

**17. Testing Support**

* ASP.NET Web API is designed with testability in mind. With features like dependency injection, mockable components, and modular services, it's straightforward to write unit and integration tests for your APIs.

**18. Response Formatting and Customization**

* ASP.NET Web API allows customizing HTTP responses, including status codes, headers, and response bodies. You can also define custom formatters if you need to support a unique media type or representation format.