NSwag is used with an API to simplify and enhance the development, documentation, and consumption of the API. It provides tools for generating client code, interactive documentation, and various features that streamline the process of building and using APIs. Here are some reasons why NSwag is used with an API:

1. \*\*Client Code Generation:\*\* NSwag allows you to generate client code in different programming languages (such as C#, TypeScript, etc.) from your API's OpenAPI (Swagger) specification. This generated client code provides a structured and easy-to-use way for developers to interact with the API without manually writing HTTP requests or handling serialization/deserialization.

2. \*\*Interactive Documentation:\*\* NSwag generates interactive API documentation using the Swagger UI or Redoc. This documentation makes it easier for developers to understand the API's endpoints, request/response structures, available parameters, and response codes. It provides an interactive sandbox for testing API calls, which can greatly assist during development and integration.

3. \*\*API Versioning and Changes:\*\* When the API evolves and changes over time, NSwag helps ensure that the generated client code remains up-to-date. If you update the API's OpenAPI specification to reflect changes, NSwag can regenerate the client code, minimizing manual code changes and reducing the chances of errors.

4. \*\*Standardization:\*\* NSwag enforces consistency and adherence to the API specification. It helps ensure that client code and documentation align with the API's intended behavior, reducing confusion and misunderstandings between API producers and consumers.

5. \*\*Development Acceleration:\*\* NSwag can expedite the development process by providing a clear contract between the API and its consumers. Developers can quickly integrate with the API using the generated client code, reducing development time and effort.

6. \*\*Code Maintenance:\*\* The generated client code can handle tasks such as request construction, parameter serialization, and error handling. This minimizes the amount of boilerplate code that developers need to write, simplifying code maintenance and reducing the potential for bugs.

7. \*\*Validation and Type Safety:\*\* By using generated client code, you benefit from compile-time validation and type safety. This means that potential errors related to incorrect API requests or responses can be caught at compile time rather than at runtime.

8. \*\*Cross-Platform Compatibility:\*\* The client code generated by NSwag can be used across different platforms and environments. This enables consistent API interactions across various client applications, including web applications, mobile apps, desktop applications, and more.

In summary, NSwag is a valuable tool that enhances the development experience for both API producers and consumers. It simplifies client code generation, improves API documentation, and helps ensure that APIs are used correctly and effectively.