Service-based APIs

By jingyi zhang u26578499

1. Google Maps API

Syntax: REST-based, offering rich URL request options.
Status: Widely used, stable.
Documentation: Very comprehensive, with numerous examples.
Versioning: Version specified through URL path.

2. YouTube API

Syntax: REST-based, supports client libraries in multiple languages.
Status: Widely used, stable.
Documentation: Detailed, easy to understand.
Versioning: API version specified in the URL.

3. Twitter API

Syntax: RESTful, supports various data formats.
Status: Widely used, continually updated.
Documentation: Very detailed, includes developer guides and tutorials.
Versioning: Clear version number management.

Library-based APIs

1. TensorFlow API

Syntax: Python-first but also supports C++, Java, and other languages.
Status: Widely used, continuously updated.
Documentation: Very comprehensive, including API reference, tutorials, and guides.
Versioning: Follows semantic versioning.

2. Numpy API

Syntax: Designed for Python, easy to use.
Status: Stable, the de facto standard for scientific computing.
Documentation: Detailed, includes tutorials and reference manuals.
Versioning: Follows semantic versioning.

3. Keras API

Syntax: Python, designed to be simple and easy to learn.
Status: Stable, tightly integrated with TensorFlow.
Documentation: Exhaustive, with rich examples.
Versioning: Follows semantic versioning.

Hardware-based APIs

1. OpenGL

Syntax: Cross-language, most widely used with C/C++.
Status: Mature and stable, but replaced by newer APIs like Vulkan on some new platforms.
Documentation: Comprehensive but complex, may be difficult for beginners.
Versioning: Clear version numbers, good backward compatibility.

Browser APIs

1. Mozilla Web APIs

Syntax: JavaScript, closely integrated with web standards.
Status: Continuously updated, following the evolution of web technologies.
Documentation: MDN provides extremely detailed documentation and examples.
Versioning: Typically not a concern for developers, managed by browser vendors.

2. Chrome APIs

Syntax: JavaScript, designed for Chrome extension development.
Status: Stable, but only available in the Chrome environment.
Documentation: Google provides detailed development documentation and guides.
Versioning: Closely related to Chrome versions.

Summary

Service-based APIs are primarily oriented towards web services, with syntax and usage modes relatively unified, emphasizing cross-platform compatibility and ease of use.
Library-based APIs focus on providing rich functionality for specific programming languages.