ONE PAGE SUMMARY

IT & Software › API Development › API Versioning

Introduction to API Versioning

Master API Versioning for Seamless Software Evolution

DEFINITION

> API versioning is the practice of managing changes to an API while maintaining compatibility. It allows developers to introduce new features without disrupting existing users.

KEY CONCEPTS

Semantic Versioning

A versioning scheme that uses three numbers (major.minor.patch) to indicate changes in functionality.

URI Versioning

Involves including the version number in the API endpoint URL, making it clear which version is being accessed.

Header Versioning

Uses HTTP headers to specify the API version, keeping the URL clean and user-friendly.

Backward Compatibility

Ensures that new API versions do not break existing client applications that rely on older versions.

Deprecation Policy

A strategy for phasing out older versions of an API, giving users time to transition to newer versions.

EXAMPLES

→ GET /api/v1/users

→ GET /api/v2/users

→ POST /api/v1/orders

MEMORY TIPS

★ Think 'Version Control' like software updates on your phone.

★ Remember 'URI' as 'Unique Resource Identifier' for clear versioning.

★ Use 'Major, Minor, Patch' to recall Semantic Versioning structure.

COMMON MISTAKES

✗ Neglecting to document version changes for users.

✗ Forgetting to maintain backward compatibility.

✗ Using inconsistent versioning methods across APIs.

QUICK RECAP

API versioning is essential for managing changes in software without disrupting users. Key methods include URI and header versioning, with a focus on backward compatibility and clear documentation.