ePlanning Program

ePlanning API Versioning Policy

Final V2.3

March 2022

NSW Department of Planning and Environment | DPIE.nsw.gov.au

Published by NSW Department of Planning, Industry and Environment

dpie.nsw.gov.au

Title: ePlanning API Versioning Policy

Subtitle: Final V2.3

First published: 13-Dec-19

1 Document change history

Date	Version	Change	Author(s)
21-Jan-2020	0.1	Draft version	Andrew Seymour
20-Apr-2020	1.0	Release version	Andrew Seymour
28-Mar-2021	2.0	Draft version	Phani Chilukuri
05-Apr-2021	2.1	Final version updates - Section 6.1 - Inbound API URL aligned with outbound URL pattern.	Phani Chilukuri
21-July-2021	2.2	Deprecation cycle increased from 12 to 24 months based on vendor and council feedback	Phani Chilukuri
22-Mar-2022	2.3	Updates made to the text in section 5 to clarify API Deprecation and Decommissioning	Dipendra Adhikari

1.1 Related documents

Related document title	Author(s)

2 Table of Contents

Contents

3 Versioning Strategy

3.1 Overview

API versioning follows a 3-number version number format, with each number separated by a period.

The version number consists of MAJOR.MINOR.PATCH information where:

MAJOR version update which includes breaking API changes,

MINOR version when functionality is added in a backwards-compatible manner,

PATCH version is used for backwards-compatible bug fixes.

Example release cycle:

API	Version	Notes
DAMgmt	1.0.0	Initial release
DAMgmt	1.0.1	Bug fix
DAMgmt	1.0.2	Bug fix
DAMgmt	1.1.0	Feature enhancement
DAMgmt	2.0.0	New Major version

3.2 Version notation in APIs

Only the Major version number is present in the URL for the API to denote the API version as Minor and Patch versions are considered transparent to the consumer. In the example below, the v1 denotes the major version of v1 for the DA Management API.

https://api.apps1.nsw.gov.au/planning/DAMgmt/v1/AcceptReturn/PAN-XXXX

when version v2 is released the URL will change to

https://api.apps1.nsw.gov.au/planning/DAMgmt/v2/AcceptReturn/PAN-XXXX

4 Backwards Compatibility

4.1 Backwards compatible (non-breaking/minor) changes

Non breaking changes are defined as changes to the API interface or functionality that do not have an adverse effect on the consuming application. These changes may be bug fixes or functional enhancements that do not impact the usability of the API for existing consumers. Consumers wishing to take advantage of new functionality in a new minor version will need to upgrade their system but they are not required to do so.

Examples of non-breaking changes include (but are not limited to):

- Adding an API interface to an API service
- Adding a method to an API interface
- Adding an HTTP binding to a method
- Adding a field to a request message
- Adding a field to a response message
- Adding a value to an enumeration (inbound API only)
- Adding an output-only resource field

4.2 Backwards incompatible (breaking/major) changes

Breaking changes are changes that will make the API unusable for existing consuming systems, requiring a change to the consuming system, including a change to the API endpoint.

Examples of breaking changes include (but are not limited to):

- Removing or renaming a service, interface, field, method or enumeration value
- Changing an HTTP binding
- Changing the data type of a field
- Changing a resource name format
- Changing visible behaviour of existing requests
- Changing the URL format in the HTTP definition
- Adding a read/write field to a resource message

5 API Lifecycle

5.1 Concurrent Versions

DPE will maintain two concurrent major versions of each API. These are defined as version ‘n’ and ‘n-1’. Each major version will have one, and only one, minor/patch version.

When a new version is released, it will become the current version (n), replacing the existing current version will be demoted to a deprecated version (n-1).

5.2 Major version

New major versions will be dictated by legislative changes and business requirements. The Department will communicate these changes to the API customer well in advance of deployment.

5.3 Minor versions

New minor versions may be released at any time in line with the Department’s development timeline and maintenance release plans.

5.4 Deprecation and Decommissioning

When an API version is superseded and it becomes the ‘n-1’ version, it is considered deprecated.

Existing consumers of the deprecated version will have up to 24 months to upgrade to the newer version of the API. During that period, the deprecated API version will continue to be supported with minor enhancements.

At the end of 24 months, it will be decommissioned, meaning that the deprecated APIs will no longer be operational.

New API customers should always start using the latest version of an API, rather than a deprecated version.

5.5 Versioning Granularity

The API versioning will apply at a Digital Service level for e.g. Online DA API V1 & Online DA API V2, which means the version updates will apply to all operations that come under each service whether there is any change or not.

All the common API operations, such as Document Management, Case Management and Payment Management, are grouped under Common APIs and will be versioned together.

Following is the list of digital services which have APIs and the current version of those APIs as at February 2021:

- Online DA V1
- Online DA V2 (in UAT)
- Online Post-consent Certificate V1
- Online Certificate Registration V1
- Online CDC API v1
- Online Section 10.7 Planning Certificate API
- Common API V1

6 Versioning Implementation

6.1 API Service URL Patterns

The following Inbound and Outbound Service URL Patterns will be adapted starting with Version 2 of the digital services that are already in Production as at February 2021. For the newly digital services, this approach will be implemented from Version 1.

Inbound API Service URL Pattern:
{DPE End Point}/ {Digital Service}/{vX}/ API Operations/{CaseID}.

Digital Service	Example Inbound service URL Patterns (API Consumers to ePlanning)
DA API	https://api.apps1.nsw.gov.au/planning/OnlineDA/v2/API Operation/{CaseID}
Post-consent Certificate API	https://api.apps1.nsw.gov.au/planning/OnlinePCC/v2/API Operation/{CaseID}
Certificate Registration API	https://api.apps1.nsw.gov.au/planning/OnlineCR/ v2/API Operation/{CaseID}
CDC API	https://api.apps1.nsw.gov.au/planning/OnlineCDC/ v2/API Operation/{CaseID}
Common API	https://api.apps1.nsw.gov.au/planning/Common/v2/API Operation/{CaseID}

Outbound API Service URL Pattern:

{External_Sys_API_Endpoint}/{Digital Service Name}/vX/API Operations/{CaseID}

Digital Service	Example Outbound service URL Patterns (ePlanning to API Consumers)
DA API	https://testcouncil.nsw.gov.au/OnlineDA/v2/API Operation/{CaseID}
Post-consent Certificate API	https://testcouncil.nsw.gov.au/OnlinePCC/v2/API Operation/{CaseID}
Certificate Registration API	https://testcouncil.nsw.gov.au/OnlineCR/v2/API Operation/{CaseID}
CDC API	https://testcouncil.nsw.gov.au/OnlineCDC/v2/API Operation/{CaseID}
Common API	https://testcouncil.nsw.gov.au/Common/v2/API Operation/{CaseID}

6.2 Guidance for Handling Breaking Changes

Inbound Breaking Changes

When a new major version is introduced due to the enhancements to the Inbound API calls, the existing production version may require the API consumers to manually provide the data elements corresponding to the breaking changes.

For example:

1. In the current state, V1 of an API is in production for a digital service.
2. Due to new requirements, several mandatory fields were introduced into the Inbound API calls for this service. ePlanning delivers the requirements by releasing a new major version: V2.
3. For all V1 requests where additional V2 data elements are required, the Portal will save all the V1 data and return a HTTP 206 response to the user with a request to provide the missing V2 data. The user will then need to login into the Portal to provide this information.

   Until the missing data corresponding to the V2 fields are supplied, the relevant process cannot progress further.

   Please note, API consumers will need to enhance their connectors to detect the newly introduced HTTP 206 code and advise their users appropriately.

   A sample response sent to the user is shown below

```
"StatusCode": 206,
"Message": "Your request has only partially succeeded as it was sent
from the V1 API version which is now deprecated.
Please login to the Portal to supply the missing information shown below"
"errorCode": "DPE_E100",
"Missing Fields":[
{
    "YAML TagName": "certificateType",
    "Portal Field": "Select the Certificate Type"
}
{
    "YAML TagName": "otherCertificateType",
    "Portal Field": "Enter Certificate Type"
}
]
```

Outbound API Breaking Change

When breaking changes are introduced in the outbound API calls, the API consumers will need to enhance their current production version to detect and ignore the placeholder values shown in the table below:

Breaking Scenario	Breaking Field Type: Placeholder Value Supplied
• Change field from conditional mandatory to optional • Change Field from mandatory to conditional mandatory • Change Field from mandatory to optional • Remove field	Integer: -32768 Double: -32768 String: E_PLACEHOLDER Enum: ENUM_PLACEHOLDER

Where the breaking field type is boolean, the Portal will inject “false” as the default value.

For example:

- The V1 version of an outbound API has a field “Property Value” as a double.
- The field was retired from the V2 version of the outbound API.
- Since it is a breaking change for V1 users, the Portal injects a placeholder value: -32768.
- Once the API consumers have enhanced their connectors to handle the placeholder values, the V1 version of API connector will ignore the -32768 received for the “Property Value” attribute.

7 Glossary

Term	Definition
Inbound API	The Inbound APIs enable a council/certifier/agency to send the data entered into their IT systems to the respective digital services in the Portal. The API flow direction is from the perspective of the Portal.
Outbound API	The Outbound APIs enable a council/certifier/agency to receive the data entered into the Portal directly into their IT system in real-time. The API flow direction is from the perspective of the Portal.