Contents

RADIANT API Versioning Guide

Overview

This document describes the API versioning strategy for the RADIANT platform.

Versioning Strategy

URL Path Versioning

RADIANT uses URL path versioning as the primary versioning mechanism:

```
https://api.radiant.example.com/v2/models
https://api.radiant.example.com/v2/chat/completions
```

Version Lifecycle

Version	Status	Support End	Deprecation
v1	Deprecated	2024-06-01	Sunset
v2	Current	-	-
v3	Planned	-	Q3 2025

Support Policy

- **Current version**: Full support, all new features
- **Previous version**: Security fixes only, 12 months after new version
- **Deprecated**: No fixes, 6-month sunset warning

Breaking vs Non-Breaking Changes

Non-Breaking Changes (No Version Bump)

These changes can be made to the current version:

- Adding new endpoints
- Adding new optional request parameters
- Adding new response fields
- Adding new enum values (with graceful handling)
- Performance improvements
- Bug fixes that don't change behavior

Breaking Changes (Require New Version)

These changes require a new API version:

- Removing endpoints
- Removing request/response fields
- Changing field types
- Changing validation rules
- Changing authentication methods
- Changing error response formats
- Removing enum values
- Changing default values

Version Header Support

Request Headers

```
# Specify API version via header (optional override)
X-API-Version: 2024-12-01

# Request specific features
X-API-Features: beta-orchestration,streaming-v2
```

Response Headers

```
# Current API version
X-API-Version: 2
X-API-Version-Date: 2024-12-01

# Deprecation warning
Deprecation: true
Sunset: Sat, 01 Jun 2024 00:00:00 GMT
Link: <https://api.radiant.example.com/v2>; rel="successor-version"
```

Deprecation Process

Timeline

```
Day 0:    Announce deprecation
          Add Deprecation header
          Update documentation

Month 3:  Send reminder emails
          Log deprecation warnings

Month 6:  Begin returning 299 status for deprecated endpoints
          Increase warning frequency

Month 12: Sunset - Return 410 Gone
          Redirect to new version docs
```

Deprecation Headers

```typescript
// Add deprecation headers to old endpoints
function addDeprecationHeaders(res: Response, sunset: Date): void {
  res.setHeader('Deprecation', 'true');
  res.setHeader('Sunset', sunset.toUTCString());
  res.setHeader('Link', '<https://api.radiant.example.com/v3>; rel="successor-version"');
}
```

Deprecation Warnings

```typescript
// Log deprecation usage for migration tracking
async function logDeprecatedUsage(req: Request): Promise<void> {
  await analytics.track({
    event: 'deprecated_api_usage',
    properties: {
      endpoint: req.path,
      version: extractVersion(req),
      apiKey: extractKeyId(req),
      tenant: extractTenantId(req),
    },
  });
}
```

Migration Guide Template

v1 to v2 Migration

```
# Migrating from API v1 to v2

## Breaking Changes

### 1. Authentication
- v1: API key in query string (`?api_key=xxx`)
- v2: API key in header (`Authorization: Bearer xxx`)

### 2. Response Format
- v1: Flat response (`{ models: [...] }`)
- v2: Wrapped response (`{ data: [...], meta: {...} }`)

### 3. Error Format
- v1: `{ error: "message" }`
- v2: `{ error: { code: "...", message: "...", details: [...] } }`

## Migration Steps

1. Update authentication headers
2. Update response parsing
3. Update error handling
4. Test all endpoints
5. Switch base URL from /v1 to /v2
```

Feature Flags

Beta Features

```
// Enable beta features via header
const betaFeatures = {
  'beta-orchestration': true,
  'beta-streaming-v2': true,
  'beta-function-calling': true,
};
```

```typescript
function isBetaEnabled(req: Request, feature: string): boolean {
  const features = req.headers['x-api-features']?.split(',') || [];
  return features.includes(feature) && betaFeatures[feature];
}
```

Graduated Features

```typescript
// Track feature graduation
const featureGraduation = {
  'function-calling': {
    beta: '2024-06-01',
    stable: '2024-09-01',
    version: 'v2',
  },
  'streaming-v2': {
    beta: '2024-09-01',
    stable: null, // Still in beta
    version: 'v2',
  },
};
```

SDK Versioning

SDK Version Matrix

SDK	Latest	Min API Version	Max API Version
JavaScript	2.5.0	v2	v2
Python	2.3.0	v2	v2
Go	1.2.0	v2	v2
Ruby	1.1.0	v2	v2

SDK Version Headers

```
# SDKs include version info
User-Agent: radiant-js/2.5.0 node/20.10.0
X-Radiant-SDK: js
X-Radiant-SDK-Version: 2.5.0
```

OpenAPI Specification

Versioned Specs

```
/docs/openapi/v2.yaml      # Current version
/docs/openapi/v2-beta.yaml # With beta features
/docs/openapi/v1.yaml      # Deprecated version
```

Schema Versioning

```yaml
# openapi.yaml
openapi: 3.1.0
info:
  title: RADIANT API
  version: 2.0.0
  x-api-version: v2
  x-version-date: '2024-12-01'
  x-deprecation-date: null
```

Testing Versions

Version Compatibility Tests

```javascript
describe('API Version Compatibility', () => {
  it('should support v2 endpoints', async () => {
    const res = await fetch('/v2/health');
    expect(res.status).toBe(200);
  });

  it('should return 410 for sunset v1 endpoints', async () => {
    const res = await fetch('/v1/health');
    expect(res.status).toBe(410);
    expect(res.headers.get('Link')).toContain('/v2');
  });

  it('should include deprecation headers for deprecated endpoints', async () => {
    const res = await fetch('/v2/deprecated-endpoint');
    expect(res.headers.get('Deprecation')).toBe('true');
    expect(res.headers.get('Sunset')).toBeDefined();
  });
});
```

Client Communication

Changelog

Maintain a public changelog:

```markdown
# API Changelog

## 2024-12-24 (v2.5)
- Added: Orchestration patterns endpoint
- Added: Workflow proposals endpoint
- Changed: Increased rate limits for Professional tier

## 2024-12-01 (v2.4)
- Added: AI translation for localization
- Deprecated: Legacy /translate endpoint (sunset 2025-06-01)
```

Email Notifications

```
// Notify developers of breaking changes
async function notifyApiChanges(change: ApiChange): Promise<void> {
  const affectedKeys = await getApiKeysUsingEndpoint(change.endpoint);

  for (const key of affectedKeys) {
    await sendEmail({
      to: key.ownerEmail,
      subject: `RADIANT API: ${change.type} - ${change.endpoint}`,
      template: 'api-change-notification',
      data: {
        change,
        migrationGuide: change.migrationGuideUrl,
        deadline: change.sunsetDate,
      },
    });
  }
}
```

Best Practices

For API Developers

1. **Plan for change**: Design APIs to be extensible
2. **Use optional fields**: Make new fields optional with defaults
3. **Version from day one**: Include version in all endpoints
4. **Document everything**: Keep OpenAPI specs updated
5. **Communicate early**: 12-month deprecation notice minimum

For API Consumers

1. **Pin versions**: Don't use unversioned endpoints
2. **Handle unknown fields**: Ignore unexpected response fields
3. **Monitor deprecation headers**: Set up alerts
4. **Test regularly**: Run integration tests against current version
5. **Subscribe to updates**: Follow changelog and email updates

Contact

Role	Contact	Purpose
API Support	api-support@radiant.example.com	Usage questions
Developer Relations	devrel@radiant.example.com	SDKs, docs
Engineering	engineering@radiant.example.com	Bug reports