



## API Readiness Checklist

How do you know when an API is ready to launch? As your program matures, the number of requirements for an endpoint to be considered ready for deployment can rapidly explode. This checklist covers common requirements that must be fulfilled before an API endpoint can be published. Since each API program often has its own quirks and unique approach to service delivery, you should feel free to adapt this list to the needs of your organization.

### Testing

- Automated code and regression tests** - Has the underlying code passed standard software quality tests?
- Security tests** - Has the code been screened for known security issues, such as insecure library dependencies and common misconfigurations?
- Design Review** - Does the endpoint, request data model, and response data model adhere to your internal style guide?
- Acceptance testing** - Does the requesting customer or product manager agree the services work as they expect?
- API endpoint functionality tests and assertions** - Does the interface respond to the right requests with the correct data and fail elegantly when fed the wrong data?

### Management

- Access controls** - Who should be allowed to have access to the data represented by the endpoint? Should any of the response data be filtered based on the role of the requesting application?
- API packaging** - Is the endpoint part of a larger API package? Has it been properly added to that package?
- Traffic and throttling** - Have you placed limits on the maximum number of calls each user can make per second, hour, day, month, etc. to ensure your services remain performant?
- Monitoring** - Have you added the endpoint to your existing monitoring solution and configured any relevant alerts?

## Support

- Customer Support Training** - Does the team supporting your developer customers have the information they need to troubleshoot issues with the endpoint?
- Endpoint documentation** - Is the endpoint's capabilities, limitations, and instructions for use fully documented, preferably using an open format such as the OpenAPI or Async API specifications?
- Common use case tutorials** - Is there material that demonstrates clearly through the use of sample code, diagrams, or a clear narrative how the endpoint is expected to be used?
- SDKs and Supporting Libraries** - Have you built and deployed updated versions of your prebuilt library code for each supported language to make it easier for developers to integrate?

## Service Discovery

- Content Pushed to Portal** - Have you published all relevant documentation - tutorials, service descriptions, SDKs, etc. - to your developer site and protected it with access controls as needed?
- Services Published To Your Marketplace** - Can your target audiences find the services in your marketplace and access them readily through your gateway?

## Promotion

- Update All Relevant Channels** - Have you posted the endpoint's availability to your public changelogs, developer-targeted social media accounts, internal and external Slack channels and wherever your target developer audiences turn to receive updates?
- Publish to External Marketplaces** - Have you shared your endpoint's details and documentation with other marketplaces where your services are listed?
- Update all Customers Waiting For Endpoint** - Who in your CRM is waiting for this functionality to launch? Have you communicated its availability to them?

# Tools That Can Help

## Testing

- For Automated Security Testing
  - [Soluble.ai](#) - Deploy a variety of open source security software applications to automatically scan your infrastructure as code for known vulnerabilities and common misconfigurations without disrupting your DevOps flows.
  - [Dependabot](#) - GitHub's native dependency scanner helps assure your code is not affected by security flaws in the code libraries your code relies upon.
  - [Snyk](#) - Developer-first automated vulnerability scanning.
- For Design Review
  - [Spectral](#) - A JSON/YAML Linter that can be used to automatically ensure new endpoints follow your service style guidelines.
- For API Endpoint Functionality Tests and Assertions
  - [Postman](#) - Test your HTTP APIs directly from your browser or desktop, and use Postman Collections and other tooling to automate testing prior to deployment.
  - [Karate by Intuit](#) - A complete testing framework written in Java that can also provide automated UI and API endpoint testing.
  - [Serenity](#) - A Behaviour Driven Development (BDD) framework designed to make writing automated tests easier.

## Management

- [The Gartner Magic Quadrant for Full Life Cycle API Management report](#) (Google it to receive a free copy from a participating vendor) is a consistently reliable source for the most popular and capable API management products on the market.

## Support

- For endpoint documentation
  - [Swagger Tools](#) - Swagger is the original name for the OpenAPI Specification. The Swagger team supports a number of both open source and commercial tools that make documenting your RESTful APIs easier and more consistent.
  - [AsyncAPI Tooling](#) - The AsyncAPI team provides a number of tools to support documenting your real time and event driven API endpoints and services.
  - [GraphQL Tools](#) - A number of tools supported by a group called "The Guild" to ease the management of GraphQL schemas.
- For SDKs and Supporting Libraries
  - [APIMatic](#) - Automagically create fully functional SDK libraries and other tooling from your API specification documents with little effort and without bloated code.