



Service Layer API for Interfaces and Hardware Component Features

This chapter provides information about the service layer API for Interface and Hardware Component features.

- [Service Layer API, on page 1](#)

Service Layer API

The Service Layer API is a model-driven API that

- leverages Google's remote procedure call (gRPC) to provide a high-performance interface
- offers direct access to the routing infrastructure layer
- provides flexibility to integrate custom protocols, agents, or controllers while IOS XR manages low-level tasks like resource conflict resolution, route preference, transactional notifications, and data plane abstractions
- supports integration of applications, routing protocols, and controllers in languages such as C++, Python, and Go without requiring additional packages, and
- enables clients to subscribe to required state event notifications, facilitating real-time monitoring and management of network devices.

Activating the Service Layer API server on your router enables it to process API requests. You must also develop a Service Layer API client to interact with the server for retrieving data, configuring settings, or managing the network. See the *Telemetry Configuration Guide for Cisco 8000 Series Routers* for details on activating the Service Layer API server and developing the Service Layer API client.

Creating the Service Layer API provides access to system resources, enabling external applications to interact with and configure the router programmatically. See [Cisco IOS-XR Service Layer](#) for details on creating the Service Layer API.

Table 1: Feature History Table

Feature name	Release Information	Description
Service Layer API for Interfaces	Release 24.4.1	<p>The Service Layer API enhances the system's performance by directly accessing the routing infrastructure layer using gRPC. It also allows flexible integration of custom protocols.</p> <p>For detailed information on Service Layer API for <code>Interfaces</code>, see Service Layer APIs for Interfaces.</p>