# **REST API Frameworks – C++**

## Comparison Table

| **Framework** | **Use Cases** | **Relevance** | **Modernity** |
| --- | --- | --- | --- |
| C++ REST SDK | Cloud-based apps, client-server communication | Widely used, industry standard | Actively maintained, modern |
| Crow | Web apps, microservices | Gaining popularity | Supports modern C++, less active |
| HttpLib | RESTful APIs, HTTP clients | Popular for small projects | Actively maintained, modern |
| Pistache | Scalable web services | Well-regarded | Actively maintained, modern |
| Restbed | Lightweight, high-performance web services | Widely used | Actively maintained, modern |
| Restinio | High-performance, low-latency web services | Known for performance | Actively maintained, modern |
| Oat++ | Scalable web apps, microservices, IoT | Gaining popularity | Actively maintained, modern, zero dependencies |
| POCO | Network apps, REST APIs, IoT | Widely used, industry standard | Actively maintained, modern, comprehensive |

You can find more detailed comparisons and benchmarks [here](https://github.com/guteksan/REST-CPP-benchmark).

**Here are some popular C++ REST API frameworks, along with a comparison based on use cases, relevance, and modernity:**

### 1. C++ REST SDK (cpprestsdk)

* **Use Cases**: Suitable for building cloud-based applications, client-server communication, and microservices.
* **Relevance**: Developed by Microsoft, widely used in the industry.
* **Modernity**: Actively maintained, supports modern C++ standards.

### 2. Crow

* **Use Cases**: Ideal for building web applications and microservices with a focus on simplicity and performance.
* **Relevance**: Gaining popularity among developers for its ease of use.
* **Modernity**: Supports modern C++ standards, but not as actively maintained as some other frameworks.

### 3. HttpLib (cpp-httplib)

* **Use Cases**: Lightweight and easy-to-use library for building RESTful APIs and HTTP clients.
* **Relevance**: Popular for small to medium-sized projects.
* **Modernity**: Actively maintained, supports modern C++ standards.

### 4. Pistache

* **Use Cases**: High-performance REST toolkit suitable for building scalable web services.
* **Relevance**: Well-regarded in the developer community.
* **Modernity**: Actively maintained, supports modern C++ standards.

### 5. Restbed

* **Use Cases**: Asynchronous RESTful framework for building lightweight and high-performance web services.
* **Relevance**: Widely used in the industry.
* **Modernity**: Actively maintained, supports modern C++ standards.

### 6. Restinio

* **Use Cases**: Suitable for building high-performance and low-latency web services.
* **Relevance**: Known for its performance and efficiency.
* **Modernity**: Actively maintained, supports modern C++ standards.

### 7. Oat++

Oat++ is another excellent C++ REST API framework. Here's how it compares with the others:

* **Use Cases**: Suitable for building highly scalable and resource-efficient web applications, microservices, and IoT applications.
* **Relevance**: Gaining popularity for its performance and ease of use.
* **Modernity**: Actively maintained, supports modern C++ standards, and has zero dependencies.

Oat++ stands out for its zero dependencies and high performance, making it a great choice for resource-efficient applications. You can find more information about Oat++

### 8. POCO Libraries

POCO (Portable Components) is a powerful and versatile C++ library that provides a wide range of functionalities, including building REST APIs. Here's how POCO compares with other C++ REST API frameworks:

* **Use Cases**: Suitable for building network-based applications, REST APIs, web services, IoT applications, and more.
* **Relevance**: Widely used in the industry, trusted by leading companies, and deployed in millions of devices.
* **Modernity**: Actively maintained, supports modern C++ standards, and offers a comprehensive set of features.

POCO stands out for its versatility and comprehensive set of features, making it a great choice for a wide range of applications, from web services to IoT.

You can find more information about POCO