

GlobalLogic[®]

A Hitachi Group Company

Uvod do REST API v jazyku C++

Zdenko Pavlik

Software developer

9.11.2021

What is REST api?

- “**Representational state transfer (REST)** is a software **architectural style** that was created to guide the design and development of the architecture for the **World Wide Web**.”
- “RESTful web APIs are typically loosely **based on HTTP methods** to access resources via URL-encoded parameters.”
- https://en.wikipedia.org/wiki/Representational_state_transfer

- **GET** - Retrieves data from a remote server. It can be a single resource or a list of resources. (“Download”)
 - GET on www.database.com/api/students
 - Returns 200 + list of students
- **POST** - Creates a new resource on the remote server. (“Upload”)
 - POST on www.database.com/api/students + body of request
 - Returns 201
- **PUT** - Updates the data on the remote server. (“Modify existing”)
 - PUT on www.database.com/api/students/1 + body of request
 - Returns 200
- **DELETE** - Deletes data from the remote server. (“Remove”)
 - DELETE on www.database.com/api/students/1
 - Returns 200
 - Another DELETE request on the same endpoint will return 404, Not found

SOAP approach - REST alternative

- Only POST is used
- Endpoints are “human-readable”
 - POST `www.database.com/api/students`POST
`www.database.com/api/students/create`
 - POST `www.database.com/api/students/1/modify`
 - POST `www.database.com/api/students/1/delete`

Support in C++ CppRestSdk (“Casablanca”)

Advantages of using standardized protocol

- WWW Compatibility between multiple languages, i.e. Java, C#, ...
- Secure HTTPS
- Tokens, authentication
- ...

CppRestSdk provides

- Providing REST API from C++ language (or any type of API)
- Built-in support of **json** data format
- Support of **pplx tasks** (parallel tasks)
- Client + Server
- Support of wildcard
- www.database.com/api/students/{XYZ}
- Server multithreaded - each request generates own thread (pplx::task, asynchronous task)
- Funny fact: In server supports it, POST request can have bodies up to **2GB** large

Client

```
const std::string path =  
"https://www.apache.org/licenses/LICENSE-2.0.txt",  
web::http::method method = web::http::methods::GET  
  
web::http::client::http_client client_dummy(U(path));  
auto response_dummy = client_dummy.request(method).get();  
std::cout << response_dummy.to_string();
```

Note: This is how NOT to use it, because `get()` will block caller thread. Instead do it with built-in **`pplx::task`**

Demo time - Client

Server

```
class handler
{
public:
    pplx::task<void> open();
    pplx::task<void> close();

    void handle_get(http_request message);
private:
    http_listener m_listener;
}

handler::handler(utility::string_t url) : m_listener(url){}
```

Server #2

```
Const std::string addr = "http://0.0.0.0:34569";  
std::unique_ptr<handler> g_httpHandler =  
    std::unique_ptr<handler>(new handler(addr));  
  
g_httpHandler->open().wait();
```

Demo time - Server

Testing

VS code ThunderClient - <https://www.thunderclient.io/>

- Performing JSON request
- Testing responses (response time, response code, body pattern, ...)
- Creating sequence collection of requests

Other tools (i.e. Postman):

Pattern for response

Storing response and using it in next request

Demo time - Testing

Facts

- MIT License - free to use, but required to mention
- Multiplatform support, Windows, Linux, MacOS, iOS, Android, Windows Phone

Unfortunately...

- *“cpprestsdk is in maintenance mode and we do not recommend its use in new projects. We will continue to fix critical bugs and address security issues.”*
- Latest release on Feb 02, 2021

- Alternatives
- Pistache - <https://github.com/pistacheio/pistache>
- Restinio - <https://github.com/Stiffstream/restinio>

Links

https://github.com/ZdenoPavlik/dockerTesting/tree/main/05_OnlineTalksDemo

- *please, ignore other folders, there is plenty of mess...*

<https://github.com/microsoft/cpprestsdk>

<https://github.com/Microsoft/cpprestsdk/wiki>

How to build

```
sudo apt install libcpprest2.10 libcpprest-dev gcc-10
```

```
cd ${projectDir}  
mkdir build && cd build  
cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug ..  
make -j 16
```

```
./05_OnlineTalksDemo/CppRestSdk_Demo
```

Q&A

A person with curly hair and glasses is looking down at a smartphone held in their hand. The phone is connected to a charging dock. In the background, a computer monitor displays a technical diagram or code. The entire scene is overlaid with a semi-transparent teal filter.

Thanks for attention...