

---

Subject: Re: Core Http API change proposals - HttpStatus and HttpMethod as enum class

Posted by [mirek](#) on Mon, 18 Jun 2018 05:38:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Klugier wrote on Sun, 17 June 2018 21:02Hello,

I agree with your, however I think we can still make it better for the final user. Instead of defining enum class we can define consts values. For example:

```
class HttpMethod {
public:
    static const String Get = "GET";
    static const String Head = "HEAD";

    HttpMethod() = delete;
};
```

The same thing we can do with Http status code:

```
class HttpStatus {
public:
    static const int Continue = 100;
    ...

    static const int OK = 200;

    String ToString(int status) {
        // Can returned string basing on status code.
    }

    HttpStatus() = delete;
};
```

I do not like the current approach when you need to explicitly write String in your code. This is more risk prone for the final user, because he can make spelling mistake in the code. In my approach it is verified on compilation level.

My idea is basing on following go standard library implementation:

- <https://golang.org/src/net/http/method.go>
- <https://golang.org/src/net/http/status.go>

Sincerely,  
Klugier

With status codes, this is outright dangerous. With methods, not worth it. I doubt that users would prefer to write `HttpMethod::Get` instead "GET".

What might make sense, although barely, is to add "IsHEAD", "IsGET" methods to `HTTPHeader` - at least we could add faster comparison (than `operator==(String, const char *)`), but it is still not worth it.

---