Subject: Core Http API change proposals - HttpStatus and HttpMethod as enum class
Posted by Klugier on Fri, 01 Jun 2018 23:29:05 GMT
View Forum Message <> Reply to Message

Hello,

I think U++ should provide better way of dealing with Http internals such as methods and status code. Currently method is implement using String and status code is implemented using regular int. It is not perfect, because the int can be out of rage of http status code. The same thing happens with method - it can be for example "WrongMethod" that doesn't exist in the standard.

My proposal for the change is to provide enum classes that will handle such problems:

```
// RFC compatibility
enum class HttpMethod {
   GET,
   HEAD,
   ...
};

enum class HttpStatus {
   ...
   BAD_REQUEST = 400,
   ...
   NOT_FOUND = 404,
   ...
};
```

We should consider changing current API or provide implicit conversion to int and String for compatibility reasons. The usage of the API will look like this (Basing on Mireks REST example):

```
void RESTProcess(TcpSocket& r)
{
 HttpHeader hdr;
 if(hdr.Read(r)) {
  VectorMap<String, String> urlvar;

  String uri = hdr.GetURI();
  int q = uri.Find('?');
   if(q >= 0) {
    ParseUrlVars(urlvar, ~uri + q + 1);
    uri.Trim(q);
   }

   String req, n;
```

```
  q = uri.Find('/');
  if(q >= 0) {
   n = uri.Mid(q + 1);
   req = Filter(n, CharFilterAlphaToLower);
  }

  switch (hdr.GetMethod()) { // Switch support is now possible...
            case HttpMethod::GET:
               if(req == "order") { }
               break;
            case HttpMethod::POST:
               if(req == "order") { }
               break;
            case HttpMethod::PUT:
               if(req == "order") { }
               break;
            case HttpMethod::DELETE:
               if(req == "order") { }
               break;
            default:
               break;
        }
 }
 HttpResponse(r, false, HttpStatus::BAD_REQUEST, "INVALID REQUEST");
}
```

General idea of this change is to make the API easier and safer to use for the library client.

Sincerely,
Klugier