5. Tutorial

Task 1

1. Explain the semantics of the following HTTP methods: HEAD, GET, PUT, DELETE, and POST. Which of them are safe, which are idempotent and which are cacheable?
2. Explain the purpose of the following HTTP headers:
   a. Host
   b. Content-Type
   c. Content-Length
   d. Accept
   e. User-Agent
   f. Location
3. Implement a class for sending HTTP/1.1 requests to a given URL. Use the template below. Create a HTTP request message using **string concatenation** only.

Task 2

Implement an HTTP message *parser* and *builder* based on the template below (take care of differentiation between request and response messages). Complete the methods `Parse` and `ToString`.

Task 3

1. What are the goals of HTTPS and how they are achieved?
2. What is the difference between HTTP and HTTPS request/response messages?

Assignment 1

Inform yourself about the "chunked" transfer encoding and its purpose. Extend the HTTP message parser and builder from Task 2 with the support for “chunked” transfer encoding.

Assignment 2

Based on the template below implement a server, which is able to deliver requested resources from the `HttpServer.DOCUMENT_ROOT` folder (for POST requests return only `201 Created`). Test your implementation in the browser.

Assignment 3

Modify[1] the HTTP request implementation of Task 1 to request the following resource:

https://www.tu-chemnitz.de (HTTPS)

---

[1] Use http://msdn.microsoft.com/en-us/library/system.net.security.sslstream.aspx as a reference