Web Application Development (LTAT.05.004)

WEB APPLICATIONS – BASIC CONCEPTS

MOHAMAD GHARIB
UNIVERSITY OF TARTU
Three-tier architecture is a software application architecture that organizes applications into three logical and physical tiers: the presentation tier; the application tier; and the data tier.
Three-tier architecture

The presentation tier is where end user interacts with the application. The application (logic/middle) tier is where information is processed. The data tier is where the information is stored and managed.
Benefits of the three-tier architecture

The three-tier architecture separates an application into physical tiers, which provides many benefits, including: 1- easier and faster development; 2- more reliable and easier to maintain; and 3- more secure applications.
Three-tier architecture in Wep App

The Front-end is where end user interacts with the application.
The Back-end is where information is processed.
The Database is where the information is stored and managed.
Three-tier architecture in Wep App

The **Front-end** is where end user interacts with the application.
The **Back-end** is where information is processed.
The **Database** is where the information is stored and managed.
A client-server architecture can be a three-tier architecture if such architecture offers presentation, application, and data tiers.
The Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML.

HTTP is a stateless protocol, which means that each HTTP request is totally independent.
Representational state transfer (REST) is a software architectural style designed for distributed hypermedia, which defines a set of constraints to be used for creating Web services.
1. Resources

Uniform Resource Identifier (URI) is a string of characters that unambiguously identifies a particular resource.
Request verbs (HTTP verbs/ HTTP methods) indicate the desired action to be performed for a given resource.
A request header can be used to provide information (meta-data) about the request context.
3. Request **Header/Body**

**Request header**

```
[Method][Path][Protocol]
Host:
Content-Type:
Accept:
Date:
```

Get / HTTP/1.1
Host: www.ut.ee
Content-Type: application/json
Accept: application/json
Date: Mon, 09 May 2022 ..

**A request header** can be used to provide information (meta-data) about the request context.
3. Request Header/Body

Request header

[Method][Path][Protocol]
Host:
Content-Type:
Accept:
Date:

POST /register HTTP/1.1
Host: www.ut.ee
Content-Type: application/json
Accept: application/json
Date: Mon, 09 May 2022 ..

Request body

```
{
"student_Id": 0031,
"student_number": 1234,
....
}
```

A request body contains the actual message to be delivered. Not all requests have body (e.g., GET, DELETE).
4. Response **Header/Body**

**Request header**

[Method][Path][Protocol]

Host:
Content-Type:
Accept:
Date:

**Response header**

[Protocol][Code][Reason]

Date:
Server:
Content-Type:
Last modified:

Get / HTTP/1.1
Host: www.ut.ee
Content-Type: application/json
Accept: application/json
Date: Mon, 09 May 2022 ..

*A response header* can be used to provide information (meta-data) about the request context.
4. Response Header/Body

Request header

[Method][Path][Protocol]
Host:
Content-Type:
Accept:
Date:

Get / HTTP/1.1
Host: www.ut.ee
Content-Type: application/json
Accept: application/json
Date: Mon, 09 May 2022 ..

Response header

[Protocol][Code][Reason]
Date:
Server:
Content-Type:
Last modified:

HTTP/1.1 200 OK
Date: Mon, 21 May 2022 ..
Server: Apache/2.2.14 (Win32)
Content-Type: application/json
Last modified: Tue, 04 May ..

A response header can be used to provide information (meta-data) about the request context.
4. Response Header/Body

Response header

[Protocol][Code][Reason]
Date:
Server:
Content-Type:
Last modified:

HTTP/1.1 200 OK
Date: Mon, 21 May 2022 ..
Server: Apache/2.2.14 (Win32)
Content-Type: application/json
Last modified: Tue, 04 May ..

Response body

{
"student_Id": 0031,
"student_number": 1234,
....
}

A response body contains the actual message to be delivered. Not all responses have body (e.g., PUT, DELETE).
5. Response Status Codes

Response codes classes:

- 1XX - Informational
- 2XX - Successful
- 3XX - Redirection
- 4XX - Client Error
- 5XX - Server Error

HTTP response status codes indicate the status of an HTTP.
5. Response Status Codes

Response codes classes:

1XX - Informational
2XX - Successful
3XX - Redirection
4XX - Client Error
5XX - Server Error

Common response codes:

200 – OK
301 - Moved to new URL
304 – Not modified (Cached version)
400 - Bad Request
401 - Unauthorized
403 - Forbidden
404 - Not found
500 - Internal Server Error
502 - Bad Gateway
503 - Service Unavailable

HTTP response status codes indicate the status of an HTTP.
Thank You for your attention

Mohamad Gharib
mohamad.gharib@ut.ee