Enterprise System Integration (MTAT.03.229)

LECTURE 6: WEB SERVICES - REST

MOHAMAD GHARIB
UNIVERSITY OF TARTU
SOAP - recap

SOAP protocol governs the interaction between the service provider and service consumer.
SOAP - recap

REST?

X Service definition

SOAP

✓ Service definition

Client

SOAP protocol governs the interaction between the service provider and service consumer

Server
REST is a term coined by Roy Fielding to describe an architecture style to be used for “creating” Web services.
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.
REST - Representational State Transfer

2. Request Verbs (GET, POST, PUT, and DELETE)

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.
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.
2. Request Verbs (GET, POST, PUT, and DELETE)  
3. Request Header/Body  
   HTTP request  
4. Response Header/Body  
   HTTP response  

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.
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 - API**

**Application Programming Interface (API)** is a set of definitions and protocols for building and integrating application software.
1. Resources - API

**API** is a software “**interface**”, offering a **service** to other software(s).

**API** is a software **intermediary** that allows two applications to talk to each other.
Uniform Resource Identifier (URI) is a string of characters that unambiguously identifies a particular resource.
1. Resources - API

Uniform Resource Identifier (URI) is a string of characters that unambiguously identifies a particular resource.

Uniform Resource Locator (URL) is a reference to a web resource that specifies its location and a mechanism for retrieving it.

**Uniform Resource Identifier (URI)** is a string of characters that unambiguously identifies a particular resource.
2. Request Verbs

2. Request Verbs (GET, POST, PUT, and DELETE)

1. Resources
Request verbs (HTTP verbs/ HTTP methods) indicates the desired action to be performed for a given resource.
2. Request Verbs

Request verbs (HTTP verbs/ HTTP methods) indicates the desired action to be performed for a given resource.
2. Request Verbs

Request verbs (HTTP verbs/ HTTP methods) indicates the desired action to be performed for a given resource.
2. Request Verbs

Request verbs (HTTP verbs/ HTTP methods) indicates the desired action to be performed for a given resource.
Request verbs (HTTP verbs/ HTTP methods) indicates the desired action to be performed for a given resource.
3. Request Header/Body

2. Request Verbs (GET, POST, PUT, and DELETE)

1. Resources

Client

HTTP request

HTTP

Server

API
3. Request **Header/Body**

**Request header**

[Method][Path][Protocol]

Host:

Content-Type:

Accept:

Date:

**Request body**

**A request header** can be used to provide information (meta-data) about the request context.
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 /booking HTTP/1.1
Host: www.rentit.ee
Content-Type: application/json
Accept: application/json
Date: Mon, 09 May 2022..
```

Request body

```
{
  "PO_Id": 0031,
  "Plant_Id": 1234,
  "Customer": "Eng. Sara"
  "Engagement Duration": 12
  ...
}
```

*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:**

### Example

**Get / HTTP/1.1**

**Host:** www.rentiit.ee

**Content-Type:** application/json

**Accept:** application/json

**Date:** Mon, 09 May 2022 ..

## Response header

- **Protocol**
- **Code**
- **Reason**

**Date:**

**Server:**

**Content-Type:**

**Last modified:**

### Example

- **Protocol:**
- **Code:**
- **Reason:**

**Date:**

**Server:**

**Content-Type:**

**Last modified:**

---

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

### Request header

<table>
<thead>
<tr>
<th><strong>[Method]</strong>[Path][Protocol]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host:</td>
</tr>
<tr>
<td>Content-Type:</td>
</tr>
<tr>
<td>Accept:</td>
</tr>
<tr>
<td>Date:</td>
</tr>
</tbody>
</table>

*Get / HTTP/1.1*

- Host: www.rentit.ee
- **Content-Type:** application/json
- **Accept:** application/json
- **Date:** Mon, 09 May 2022 ..

### Response header

<table>
<thead>
<tr>
<th><strong>[Protocol]</strong>[Code][Reason]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date:</td>
</tr>
<tr>
<td>Server:</td>
</tr>
<tr>
<td>Content-Type:</td>
</tr>
<tr>
<td>Last modified:</td>
</tr>
</tbody>
</table>

*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.
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.
REST – the full picture

Plants list

Web sever
REST – the full picture

HTTP GET request

HTTP response
HTML, JSON, etc.

Web server

Plants list

Read/ Query

Result

1xx  2xx  3xx
4xx  5xx  6xx
REST – the full picture

HTTP GET request

HTTP response
HTML, JSON, etc.

HTTP GET request

HTTP response
HTML, JSON, etc.

Web server

Plants list

Read/Query
Result
Read/Query
Result

From:
To:
Dates:
search
REST – the full picture

HTTP GET request

HTTP response
HTML, JSON, etc.

HTTP GET request

HTTP response
HTML, JSON, etc.

Web server

Plants list

Read/Query
Result

Read/Query
Result

Booking list
REST – the full picture

HTTP GET request
HTTP response
HTML, JSON, etc.

HTTP GET request
HTTP response
HTML, JSON, etc.

HTTP POST request
HTTP response
XML, JSON, etc.

Plants list
Read/Query
Result

Book list
Create
Result

ID:
Location:
Dates:
search

From:
To:
Dates:
search

4xx
5xx
6xx
1xx
2xx
3xx

Web server
REST – the full picture

HTTP GET request
HTTP response
HTML, JSON, etc.

HTTP GET request
HTTP response
HTML, JSON, etc.

Loaded/updated depending on GET

Plants list

Read/Query
Result

Read/Query
Result

Booking list
REST – the full picture
REST – the full picture

HTTP GET request

HTTP response
HTML, JSON, etc.

HTTP GET request

HTTP response
HTML, JSON, etc.

Web server

Plants list

Read/Query

Result

Read/Query

Result

Booking list

Loaded/updated depending on GET
REST – the full picture

HTTP GET request

HTTP response
HTML, JSON, etc.

HTTP GET request

HTTP response
HTML, JSON, etc.

HTTP DELETE request

HTTP response
XML, JSON, etc.

Plants list

Result

Result

Delete

Booking list

Loaded/updated depending on GET
Final remarks

**REST is NOT a**
- A Framework
- A Technology
- Standard specifications
- A Protocol

**REST is**
- **REST** is a set of guidelines and architectural principles, which talks about how should a client interact with a server.
- **RESTful** is the implementation of **REST** guidelines
- **RESTful** is Web Services
Thank You for your attention

Mohamad Gharib
mohamad.gharib@ut.ee

unitartu
tartuuniversity