Designing REST Services: Representational Format

Web Programming and Testing

Mario Simaremare, S.Kom., M.Sc.
Program Studi Sarjana Sistem Informasi
Institut Teknologi Del

Objectives

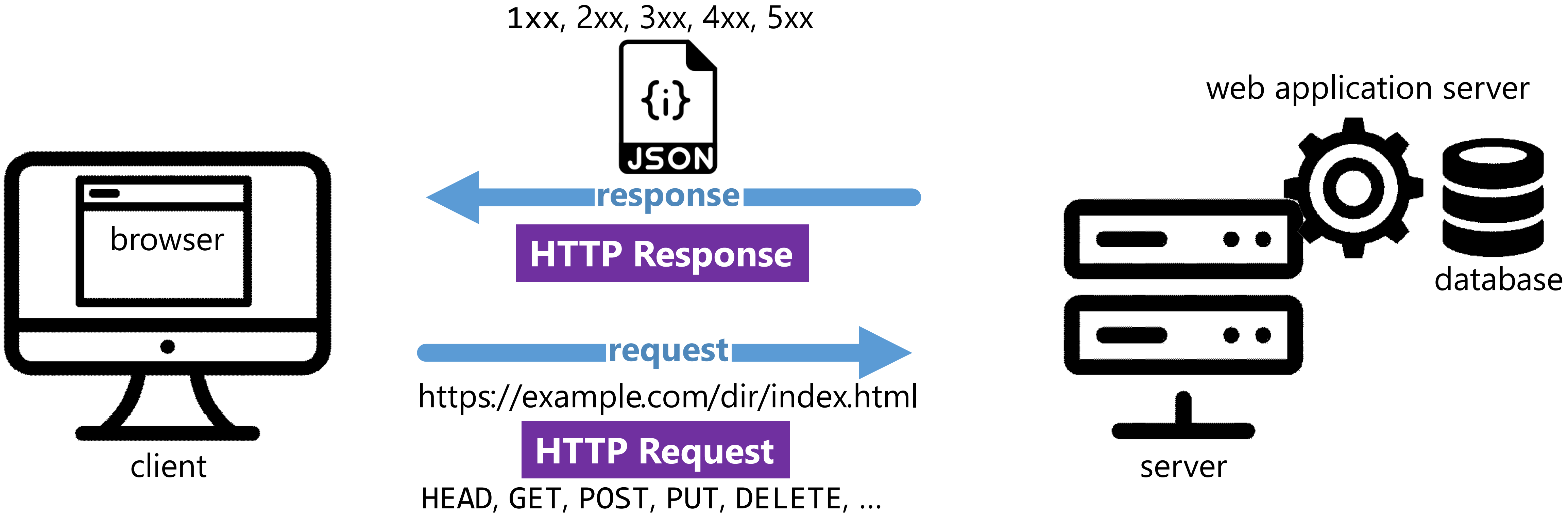
- The objective of this session is the following:
  - The students are able to develop REST-based services. On this session, we focus on the design phase especially the representational format.

Outlines

1. Semantic in REST Request-Response Interactivity.
2. Representational Formats.
3. XML.
4. JSON.
5. Representational Rules.

Semantic in REST
Request-Response Interactivity

Request-Response Cycle



Representational Formats

State Representation

- A resource state is exchanged between the two parties.
  - The state describe the resource at a moment in time.
- In REST, the state are written in a format described in the contract (documentation).
  - The consumers agree with the format.
- Commonly used representational format:
  - `JSON` and `XML`.
  - Other? `RDF`, `YAML`, etc.

XML

XML

- eXtensible Markup Language
    - Traditional state wrapper.
    - Commonly used in:
        - legacy systems with SOAP-based services.
        - configuration files.
- To read an XML document, a parser is used.
    - e.g. XPATH, XSLT, Xquery, etc.

XML Formatting

- States are wrapped in a nested nodes.
  - A node may have one or more attributes.
  - No standard tags.
- A node is written in an open-closed tag.
  - `<tag attribute=“attribute-value”>value</tag>`

XML: An Example

Root node

An node with
a value

An array of
values

Node attribute

```
<?xml version="1.0" encoding="UTF-8"?>
<profile>
  <display>Sigurita Tralala</display>
  <age>22</age>
  <favourite-fruits>
    <fruit>Apple</fruit>
    <fruit>Cherry</fruit>
  </favourite-fruits>
  <phones>
    <phone active="true" type="private">085262211212</phone>
    <phone active="true" type="office">0632331234</phone>
  </phones>
</profile>
```

JSON

JSON

- JavaScript Object Notation
  - A way to represent an object in JavaScript.
  - Later, it is used as data exchange format.

- Why `JSON`?
  - Most of the REST clients are in the form of web pages. This makes `JSON` is a friendlier format compared to XML.
  - Plain and interoperable (cross-platform).

See: https://tools.ietf.org/html/rfc8259

JSON Formatting

- Key formatting:
  - An object is described in a curly braces `{}`.
  - Attributes are written in a key-value pair,
    with key name written in a double quotes `""`.
  - An array of values are written in a square braces `[]`.
  - Nested object and nested array are allowed.

- MediaType
  - To let another application about the `JSON` formatting
    use the `application/json` MIME type.

JSON: An Example

It is possible to have a nested object or nested array.

Attribute name

Attribute value

An array

An array of objects

```
{
  "display": "Sigurita Tralala",
  "github_username": "@sigurita",
  "age": 22,
  "favourite_fruit": [
    "Apple",
    "Cherry"
  ],
  "phone": [
    {
      "no": "085262211212",
      "type": "private",
      "active": true
    },
    {
      "no": "0632331234",
      "type": "office",
      "active": true
    }
  ]
}
```

Representational Rules

Representational Rules

- Rule #42: JSON Should be the default representational format.
- Rule #43: JSON must be well-formed. Follow the standard described in RFC 8259.
- Rule #44: Other formats, beside JSON are optional to be supported.

To-dos

1. Next time we will discuss REST as a hypermedia.
2. Design your REST APIs.

References

Srinivasan, M. (2012). Web Technology: Theory and Practice. Pearson.

Erl T. (2016). Service-Oriented Architecture: Analysis and Design for Services and Microservices. Pearson

Massé, M. (2012). REST API Design Rulebook. O'Reilly

Thank you

institut teknologi del

MARTUHAN-MARROHA-MARBISUK

2001