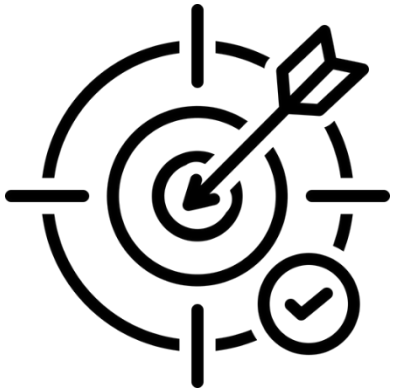
Designing REST Services: HTTP Verbs

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 use of HTTP verbs.



Outlines

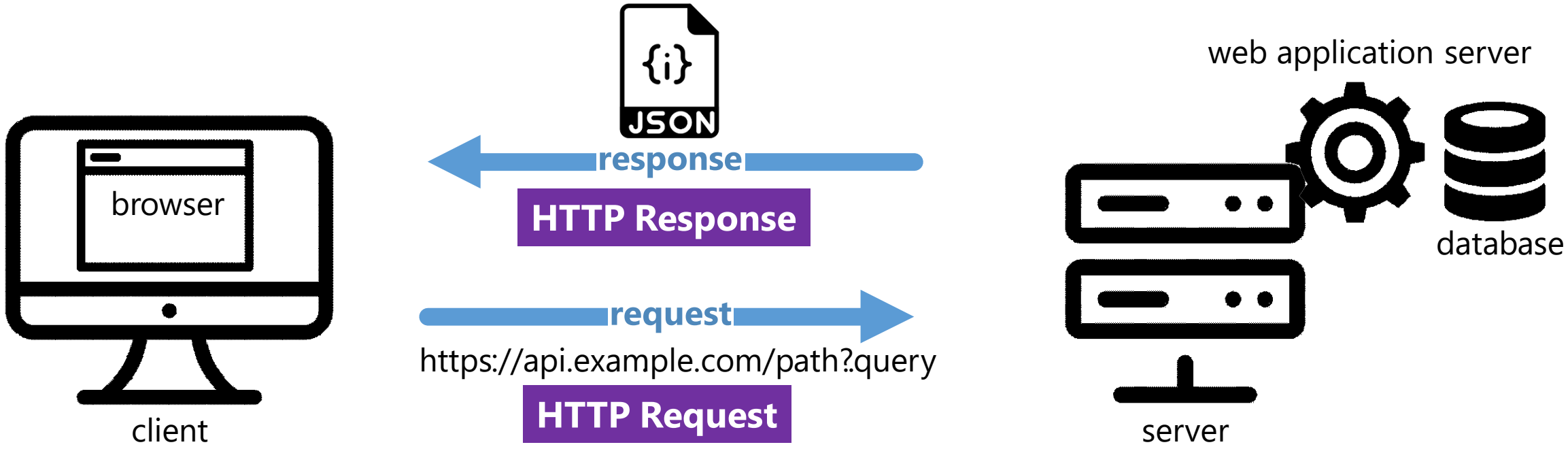
1. Semantic in REST Request-Response Interactivity.
2. HTTP Verb Rules.

Semantic in REST
Request-Response Interactivity

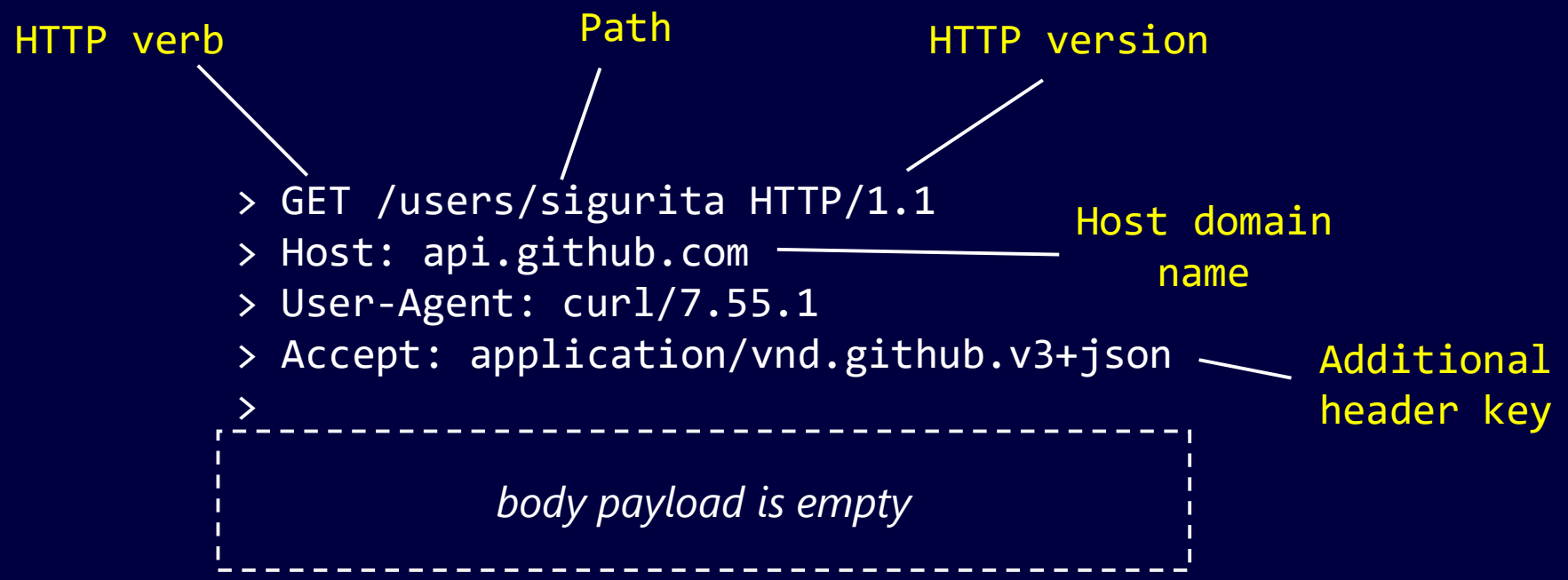
Semantic?

- REST APIs are meant to be easily understood and more self-explanatory.
- To achieve such thing, HTTP request verbs and response codes are utilized.
- Another level of semantic, hypermedia, is spared for a later discussion.

Request-Response Cycle

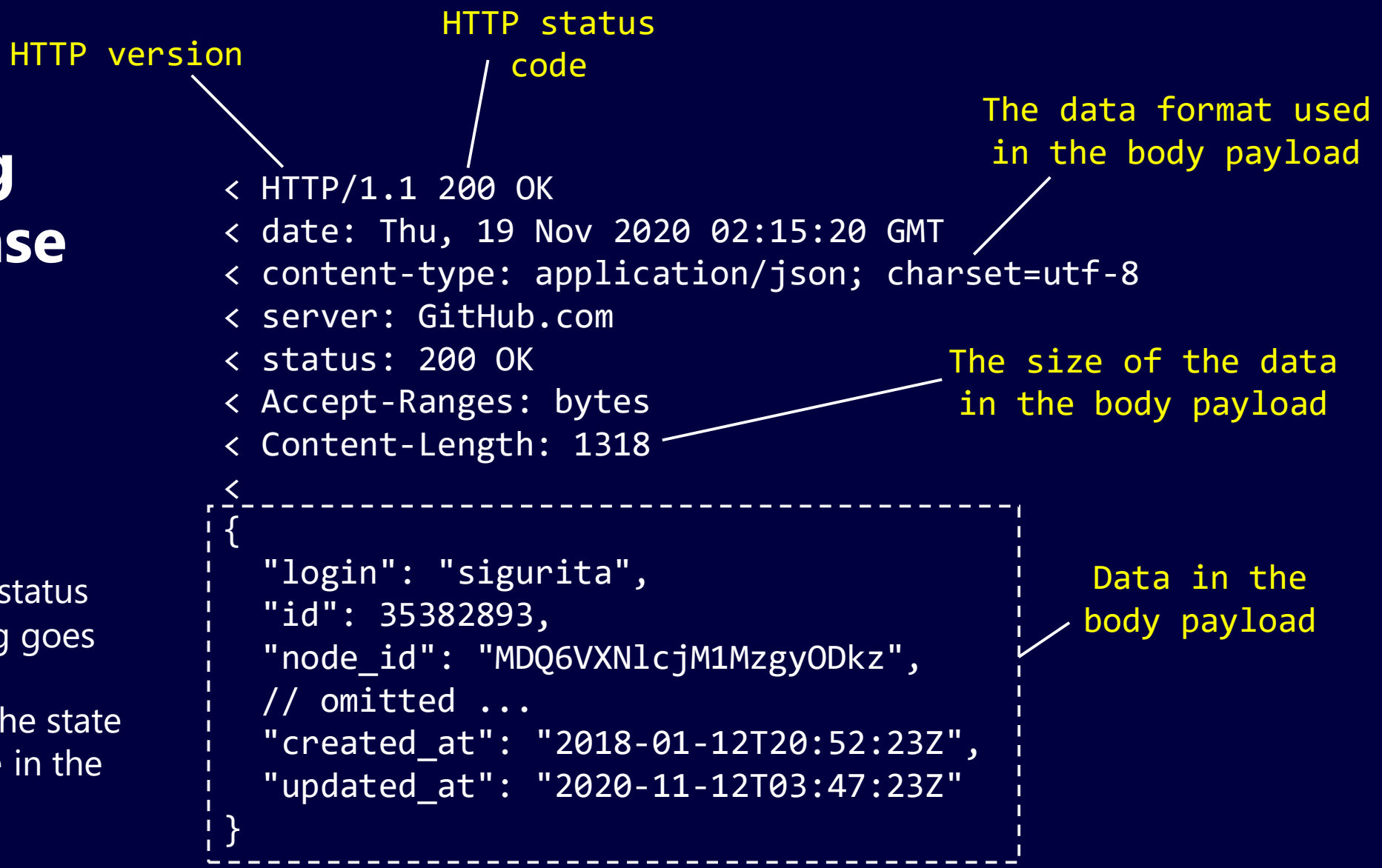


Decomposing HTTP Request



The request uses GET method, meaning it will not cause state change the requested resource. In this example, the request does not attach any data in the body payload.

Decomposing HTTP Response



The response retuns 200 status code, meaning everything goes fine (OK).
The response also bring the state of the requested resource in the form of JSON.

HTTP Verb Rules

HTTP Verb	Purpose
HEAD	Retrieve the resource state meta data
GET	Retrieve the resource state and its meta data
POST	Create a new resource in a collection
PUT	Create or modify a resource in a store
PATCH	Similar to PUT but for partial modification
DELETE	Remove a resource
OPTIONS	Retrieve the resource availability
TRACE	Sending a loopback, for debugging purposes

https://www.rfc-editor.org/rfc/rfc2616.txt

HTTP Verb Rules

- Rule #16: `GET` and `POST` should not be used to tunnel other request methods.
  - e.g. Using `GET` request to remove a resource or `POST` to modify it.

- Rule #17: `GET` method must be used to retrieve a resource's state representation.
  - A `GET` request consists of header only without any payload (body).

HTTP Verb Rules

- Rule #18: HEAD should only be used to retrieve a resource state meta data not the whole state.
  - Similar to the GET request but expecting the header section only.
- Rule #19: POST must be used in the case of creating new resource in a collection.
  - Very likely to attach the corresponding data along with the request.
- Rule #20: POST must be used to invoke a controller on a particular resource.

HTTP Verb Rules

- Rule #21: PUT must be used to both create or modify a resource in a store.
  - Very likely to attach the corresponding data along with the request.

- Rule #22: PATCH is an alternative to the PUT.
  - Used when the modification is partial.
  - Will cause a small traffic compared to the PUT.

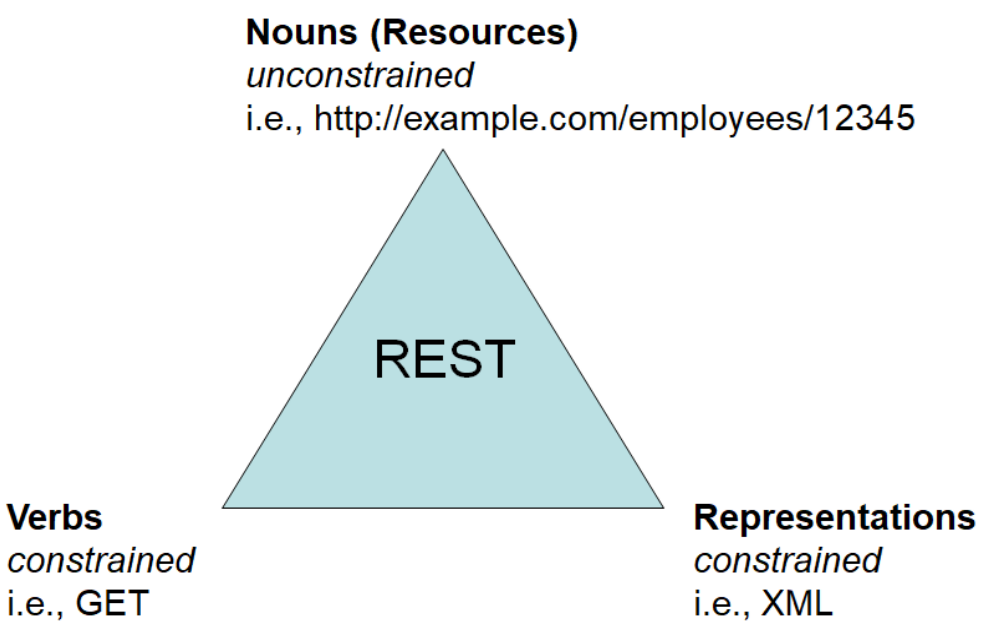
HTTP Verb Rules

- Rule #23: `DELETE` should be used to remove a resource or to make it unavailable.
  - Applicable to resource that resides in both collection or store.

- Rule #24: `OPTIONS` request should be used to retrieve metadata that describe a resource's available interactions.
  - The response may include a payload to further describe each interaction.

To-dos

1. Next time we will discuss more on the semantic aspects, both the HTTP response codes.
2. Afterwards, we will discuss about the representational aspect.
3. Explore the GitHub 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

GitHub REST API Documentation. https://docs.github.com/

Thank you