ADVANCED TOPIC IN BIG DATA

Quick Review

- Syllabus
- JSON
- JSON Schema
- Creating strongly types system with JSON
- The need for validation
- Addressing (briefly)

Creating strongly typed data with Json

- Every object is an instance of a type
- System exposes aspects, e.g. _id, _type, etc. that are used in any object.
- Define the type version in the system, and associate with it the property test.
- The property test has datatype array of integers

- {
- "_name": "version",
- "_type": "Entry",
- "_org": "logoilabs.com",
- "_id": "version",
- "properties": [{
- "_name": "test",
- "dataType": ["int"],
- "_type": "Field",
- "_org": "logoilabs.com",
- "_id": "version----test",
- "isFieldOf": "version"}]
- }

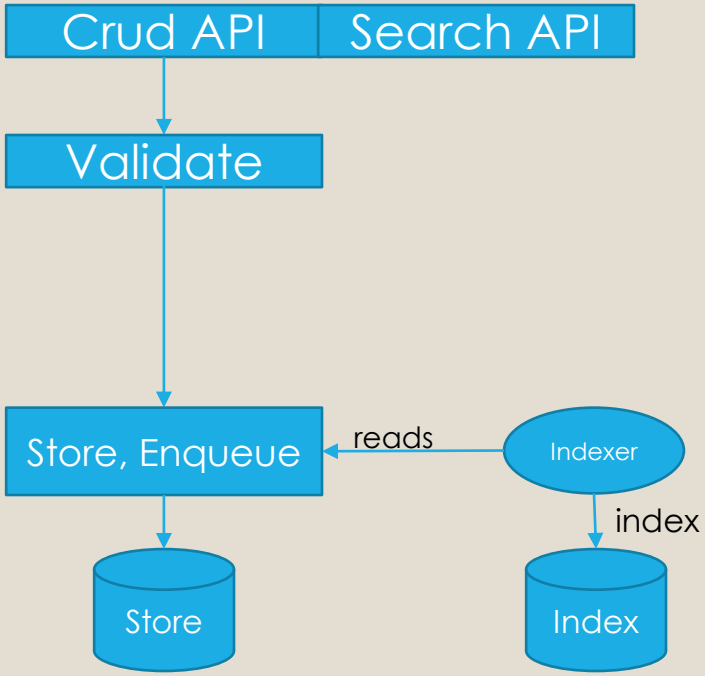
Variety of strongly typed system

- define types and properties
- define references to objects
- support for inheritance?
- extending the definition of types with additional properties
- system defined types and properties
- aspects
- support for versioning?
- advanced data modeling primitives: intersection, one of, cardinality support, union
- Examples:
  - GDATA, https://developers.google.com/gdata/
  - Protobuf, https://developers.google.com/protocol-buffers/
  - Microsoft Odata
  - Facebook:GraphQL

Class exercise

- Come up with a convention to depict a reference to an object in a json payload
- See if you can come up with two different ways of doing it, and then compare the two to choose the better one
- 15 minutes

Architecture



Rest API Specifications

- URI conventions
  - /type/id
- Headers
  - Students should review the HTTP standard headers
  - Various uses of Etag, Not Modified Since, Authorization in Rest APIs
- Payload structure and serialization
- Security
- Status Code
- Example: https://www.hl7.org/fhir/http.html

◦ marwansabbouh@gmail.com