Java Microservices MOOC

Lesson 1-3: REST and the Web

ORACLE®

Lesson 1-3:
REST and the Web

Representational State Transfer

REST is the architecture that the entire web uses. It is based on the following design principles:

- Client/server interactions: Browser – Web Server

- Uniform interface: HTTP, URL

- Layered System: IP: Proxies, Gateways, and so on

- Cache: Web browser cache, intermediary caches

- Stateless: HTTP request-and-response model

- Code-on-demand: JavaScript code

REST in Action

To use REST in web applications, you need to address the following:

- The HTTP method that you need

- The resource identified by its URI

- The content for the request and the response
  - The preferred data exchange format is JSON.

REST

HTTP Methods

GET
Get a resource

Used to get a resource from the server.
For example, Get user list, or get the details of
    a user.

POST
Add a resource

Used to create a new resource in the server.
For example, add a new user.

PUT
Update a resource

Used to update a resource in the server.
For example, update the user details.

DELETE
Delete a resource

Used to delete a resource in the server.
For example, delete or disable a user.

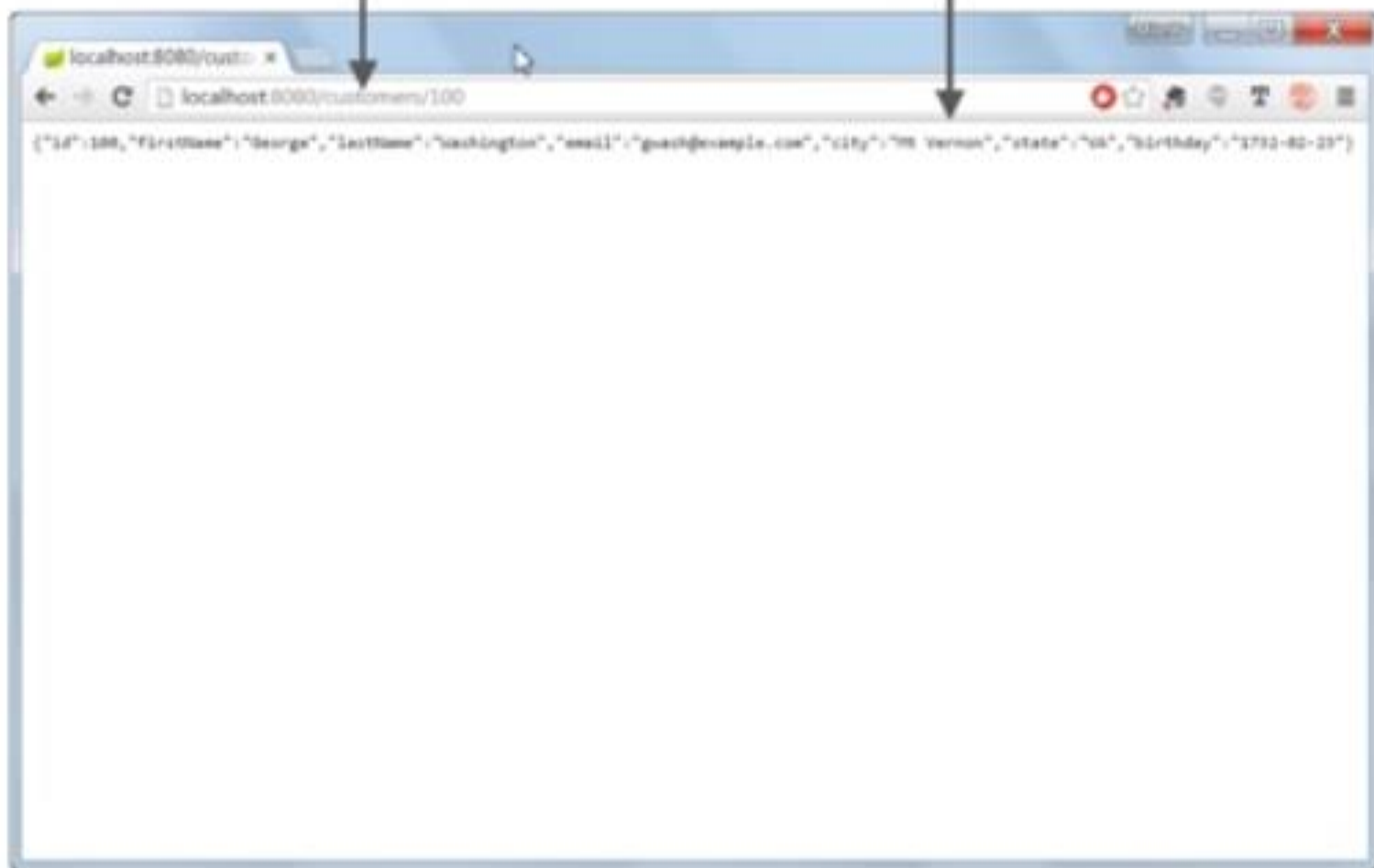# RESTful URI

How are RESTful URIs designed to access resources?

- GET http://www.example.com/users/
  - Lists all the users

- GET http://www.example.com/users/john
  - Gives the details of a user identified as "john"

- POST http://www.example.com/users/
  - Creates a new user

- PUT http://www.example.com/users/john
  - Updates the user identified as "john"

Practical REST Example

Address: Uniform Resource Identifier

Pressing Enter executes a "GET" request on the resource. A JSON response is returned.

Subscribe

For more information

Oracle Application Container Cloud Service
http://cloud.oracle.com/application-container-cloud

Oracle Learning Library
oracle.com/oll