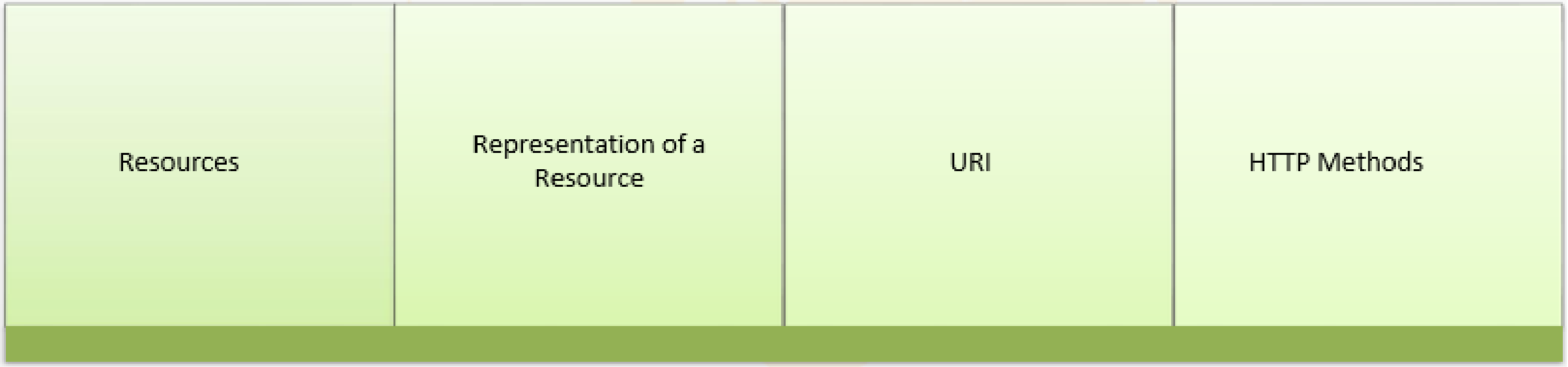
Bài 04
Restful Web Services

MỤC TIÊU

- Giải thích restful web services
- Xây dựng restful web services với thư viện jersey

Restful Web services

- **REST:** Viết tắt của Representational State Transfer.
- Là các dịch vụ web dựa vào kiến trúc chuẩn web và được truy xuất sử dụng giao thức của HTTP.
- Kiến trúc của restful web services:

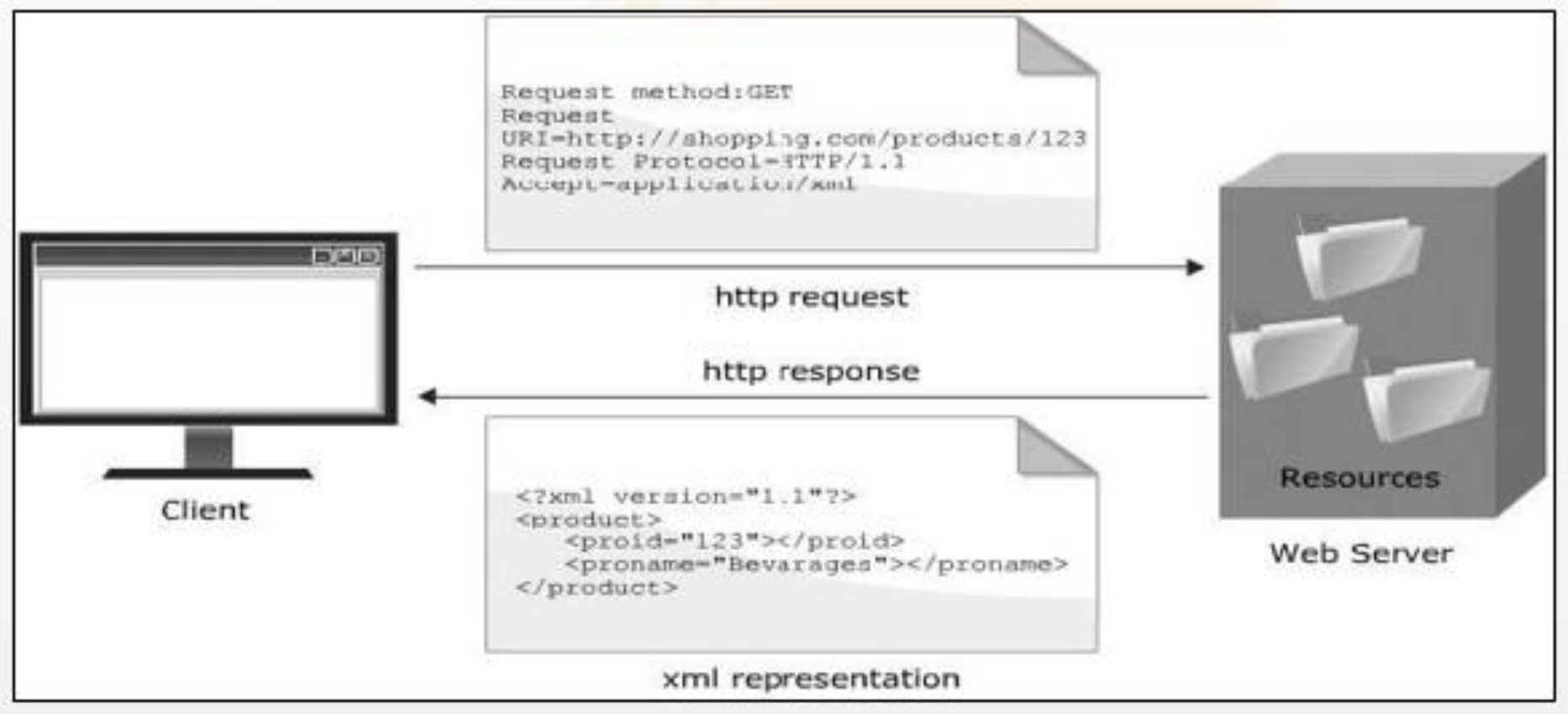


Các tài nguyên

- Được xem như bất kỳ kiểu thông tin nào được tìm thấy qua mạng
- Thông tin có thể là: Một tài liệu, một hình ảnh, hoặc một thực thể như thông tin của một sinh viên, chi tiết về chuyến bay từ CSDL.
- Các ví dụ về tài nguyên của Restful web services:
  - Các khóa học được cung cấp bởi một trường Đại học
  - Bảng tin thời sự của kênh CNN
  - Kết quả tìm kiếm cho bất kỳ chủ đề nào trong máy tìm kiếm
  - Thông tin chuyến bay được lưu trữ trong CSDL bay.

Biểu diễn một tài nguyên

- Việc biểu diễn có thể được xem như là định dạng trong đó tài nguyên được gửi và nhận trong suốt quá trình tương tác của client và server ở trên web.



URI và HTTP Methods

- Với restful web services, một tài nguyên được xác định bởi một URI.
- Một URI là kết hợp của tên và địa chỉ tài nguyên, giúp client và server thay đổi dữ liệu trong suốt quá trình tương tác.
- Restful web service hỗ trợ các phương thức của HTTP để gửi và nhận dữ liệu được biểu diễn như một tài nguyên trên mạng. Các phương thức của HTTP
  - GET
  - POST
  - PUT
  - DELETE

Tạo restful web service với thư viện jersey

- Bước 1: Tạo ứng dụng dynamic web project và convert về maven project
- Bước 2: Thêm các thư viện vào file pom.xml
  - jersey-server
  - jersey-servlet
  - jersey-core
  - gson
  - hibernate-core
  - hibernate-entitymanager
  - Add và build path thư viện kết nối CSDL sqljdbc4-4.0.jar

- Bước 3: Cấu hình file web.xml

*web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee; http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>Demo_WebService</display-name>
    <servlet>
      <servlet-name>myservlet</servlet-name>
      <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
      <init-param>
          <param-name>com.sun.jersey.config.property.packages</param-name>
          <param-value>webapi.services</param-value>
      </init-param>
    </servlet>
    <servlet-mapping>
      <servlet-name>myservlet</servlet-name>
      <url-pattern>/rest/*</url-pattern>
    </servlet-mapping>
</web-app>
```

- Bước 4: Cấu hình file hibernate.cfg.xml

hibernate.cfg.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServerDriver</property>
        <property name="hibernate.connection.password">1234$</property>
        <property name="hibernate.connection.url">jdbc:sqlserver://localhost:1433;databaseName=DBService_RestFul</property>
        <property name="hibernate.connection.username">sa</property>
        <property name="hibernate.dialect">org.hibernate.dialect.SQLServer2008Dialect</property>
        <mapping class="webapi.entities.City"/>
        <mapping class="webapi.entities.Country"/>
    </session-factory>
</hibernate-configuration>
```

- Bước 4: Cấu hình file HibernateUtil.java

HibernateUtil.java

```
1  package webapi.hibernate.util;
2
3  import org.hibernate.SessionFactory;
8
9  public class HibernateUtil {
10     private static SessionFactory sessionFactory;
11     static {
12         if(sessionFactory==null) {
13             ServiceRegistry reg = new StandardServiceRegistryBuilder().configure().build();
14             MetadataSources source = new MetadataSources(reg);
15             Metadata metadata = source.getMetadataBuilder().build();
16             sessionFactory = metadata.getSessionFactoryBuilder().build();
17         }
18     }
19
20     public static SessionFactory getSessionFactory() {
21         return sessionFactory;
22     }
23 }
```

- Bước 5: Tạo các lớp entity ánh xạ với các bảng trong CSDL

City.java

```
1  package webapi.entities;
2
3  import java.util.Date;
14
15 @Entity
16 @Table(name = "City")
17 public class City {
18     @Id
19     @GeneratedValue(strategy = GenerationType.IDENTITY)
20     @Column(name = "CityId")
21     private Integer cityId;
22     @Column(name = "CityName")
23     private String cityName;
24     @Column(name = "FoundationDate")
25     private Date foundationDate;
26     @Column(name = "Area")
27     private Double area;
28     @Column(name = "Population")
29     private Long population;
30
31     @ManyToOne(fetch = FetchType.EAGER)
32     @JoinColumn(name = "CountryId",referencedColumnName = "CountryId")
33     private Country objCountry;
34
35
36     public City() {
37         super();
38         // TODO Auto-generated constructor stub
39     }
```

- Bước 6: Tạo các lớp entity DTO để làm việc với kiểu JSON trong web service

*CityDTO.java

```java
package webapi.entities.dto;

import java.util.Date;

public class CityDTO {
    private Integer cityId;
    private String countryId;
    private String cityName;
    private Date foundationDate;
    private Double area;
    private Long population;
    public CityDTO() {
        super();
    }
    public CityDTO(Integer cityId, String countryId, String cityName, Date foundationDate, Double area,
            Long population) {
        super();
        this.cityId = cityId;
        this.countryId = countryId;
        this.cityName = cityName;
        this.foundationDate = foundationDate;
        this.area = area;
        this.population = population;
    }
    public Integer getCityId() {
        return cityId;
    }
    public void setCityId(Integer cityId) {
        this.cityId = cityId;
```

- Bước 7: Khai báo hàm trong giao diện DAO

CityDAO.java

```
1  package webapi.dao;
2
3  import java.util.List;
6
7  public interface CityDAO {
8      public List<City> getListCitites();
9      public City getCityById(Integer cityId);
10     public boolean insertCity(City city);
11     public boolean updateCity(City city);
12     public boolean deleteCity(Integer cityId);
13     public List<City> getCitiesByCountryName(String countryName);
14 }
```

- Bước 8: Cài đặt các hàm trong lớp thực thi DAO

CityDAOImpl.java

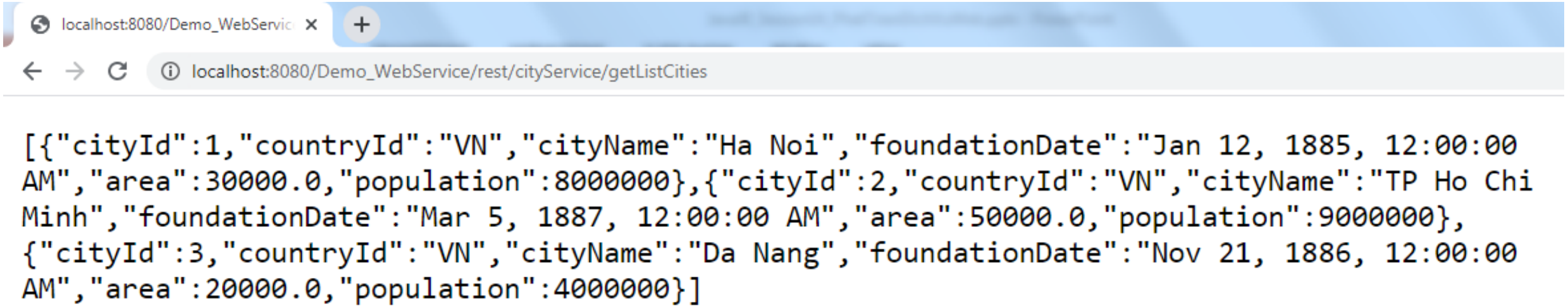
```java
package webapi.dao;

import java.util.List;

public class CityDAOImpl implements CityDAO{
    @Override
    public List<City> getListCitites() {
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        Session session = sessionFactory.openSession();
        try {
            session.beginTransaction();
            List list = session.createQuery("from City").list();
            session.getTransaction().commit();
            return list;
        } catch (Exception e) {
            e.printStackTrace();
            session.getTransaction().rollback();
        }finally {
            session.close();
        }
        return null;
    }
    @Override
    public City getCityById(Integer cityId) {
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
        Session session = sessionFactory.openSession();
        try {
            session.beginTransaction();
            City c = session.get(City.class, cityId);
            session.getTransaction().commit();
            return c;
        } catch (Exception e) {
            e.printStackTrace();
            session.getTransaction().rollback();
        }finally {
            session.close();
        }
        return null;
    }
```

- Bước 9: Cài đặt lớp service

CityService.java

```
1  package webapi.services;
2
3  import java.util.ArrayList;
19
20 @Path(value = "/cityService/")
21 public class CityService {
22
23     @GET
24     @Path("/getListCities")
25     @Produces(MediaType.APPLICATION_JSON)
26     public String getListCities() {
27         List<City> listCitites = new CityDAOImpl().getListCitites();
28         Gson son = new Gson();
29         List<CityDTO> listData = new ArrayList<CityDTO>();
30         for (City c : listCitites) {
31             CityDTO cdto = new CityDTO(c.getCityId(), c.getObjCountry().getCountryId(), c.getCityName(), c.getFoundationDate(), c.getArea(
32             listData.add(cdto);
33         }
34         String data = son.toJson(listData);
35         return data;
36     }
37
38     @POST
39     @Path("/insertCity")
40     @Consumes(MediaType.APPLICATION_JSON)
41     public String insertCity(String c) {
42         Gson son = new Gson();
43         City objCity = son.fromJson(c, City.class);
44         boolean bl = new CityDAOImpl().insertCity(objCity);
```

- Kết quả test với trình duyệt



HỎI ĐÁP

TRẢI NGHIỆM THỰC HÀNH

TRÂN TRỌNG CẢM ƠN!

238 Hoàng Quốc Việt, Bắc Từ Liêm, Hà Nội

0968.27.6996

tuyensinh@bachkhoa-aptech.edu.vn

www.bachkhoa-aptech.edu.vn