

## Bài thực hành 06

# Xây dựng ứng dụng Web client với Restful Web Services

### 1. Mục tiêu

- Xây dựng ứng dụng web bên phía client để gọi và sử dụng các dịch vụ được cung cấp bởi Restful Web services

### 2. Bài thực hành Step by Step

**Bài thực hành 1:** Từ một ứng dụng restful web services đã được xây dựng và triển khai. Tạo một ứng dụng web bên phía client gọi và sử dụng các chức năng CRUD được cung cấp bởi web services.

#### Bước 1: Tạo ứng dụng Dynamic Web Project có tên RestfulWebservice\_Client

File → New → Other → Web → Dynamic Web Project . Đặt tên project  
"RestfulWebservice\_Client"

#### Bước 2: Convert to maven project

Convert project về maven project để quản lý các thư viện của dự án trong file pom.xml

Click phải vào project → Configure → Convert to Maven Project → Finish

#### Bước 3: Thêm các thư viện cho ứng dụng trong file pom.xml

Mở file pom.xml chèn cặp thẻ <dependencies>...</dependencies> giữa thẻ <packaging> và thẻ <build>

Thêm các thư viện sau:

- o jersey-client
- o gson
- o jstl

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
```



```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>RestfulWebservice_Client</groupId>
<artifactId>RestfulWebservice_Client</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<dependencies>
<!-- https://mvnrepository.com/artifact/com.sun.jersey/jersey-client -->
<dependency>
<groupId>com.sun.jersey</groupId>
<artifactId>jersey-client</artifactId>
<version>1.19.4</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
<groupId>com.google.code.gson</groupId>
<artifactId>gson</artifactId>
<version>2.8.6</version>
</dependency>
<!-- https://mvnrepository.com/artifact/jstl/jstl -->
<dependency>
<groupId>jstl</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
</dependencies>
<build>
<sourceDirectory>src</sourceDirectory>
<plugins>
<plugin>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>1.8</source>
<target>1.8</target>
</configuration>
</plugin>
<plugin>
<artifactId>maven-war-plugin</artifactId>
<version>3.2.3</version>
<configuration>
<warSourceDirectory>WebContent</warSourceDirectory>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

#### **Bước 4: Tạo class entity Book.java**

```
package book.entities;
```



```
import java.util.Date;

public class Book {
    private Integer bookId;
    private String bookName;
    private String author;
    private String publisher;
    private Date publishDate;
    private Integer pages;
    private Double price;

    public Book() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Book(Integer bookId, String bookName, String author, String publisher,
        Date publishDate, Integer pages,
        Double price) {
        super();
        this.bookId = bookId;
        this.bookName = bookName;
        this.author = author;
        this.publisher = publisher;
        this.publishDate = publishDate;
        this.pages = pages;
        this.price = price;
    }

    public Integer getBookId() {
        return bookId;
    }

    public void setBookId(Integer bookId) {
        this.bookId = bookId;
    }

    public String getBookName() {
        return bookName;
    }

    public void setBookName(String bookName) {
        this.bookName = bookName;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getPublisher() {
        return publisher;
    }
}
```



```
}

public void setPublisher(String publisher) {
    this.publisher = publisher;
}

public Date getPublishDate() {
    return publishDate;
}

public void setPublishDate(Date publishDate) {
    this.publishDate = publishDate;
}

public Integer getPages() {
    return pages;
}

public void setPages(Integer pages) {
    this.pages = pages;
}

public Double getPrice() {
    return price;
}

public void setPrice(Double price) {
    this.price = price;
}

}
```

## CHỨC NĂNG LOAD DATA

### Bước 5: Tạo trang index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <jsp:forward page="LoadBooks"></jsp:forward>
</body>
</html>
```

### Bước 6: Tạo servlet LoadBooks.java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
```



```
// TODO Auto-generated method stub
Client client = Client.create();
WebResource resource = client
    .resource("http://localhost:8080/Webservice_Session05_DB/rest/book-
service/listBooks");
String data = resource.get(String.class);

Gson son = new Gson();
GenericType<List<Book>> listBookType = new GenericType<List<Book>>() {};
List<Book> listBook = son.fromJson(data, listBookType.getType());

// xu ly success
Object success = request.getAttribute("success");
if (success != null) {
    request.setAttribute("success", success);
}

// xu ly error
Object error = request.getAttribute("error");
if (error != null) {
    request.setAttribute("error", error);
}

request.setAttribute("list", listBook);
request.getRequestDispatcher("listBooks.jsp").forward(request, response);
}
```

### **Bước 7:** Tạo trang listBooks.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>LIST BOOKS</h1>
    <h3 style="color:green">${success}</h3>

    <table border="1">
        <tr>
            <th>Book Id</th>
            <th>Book Name</th>
            <th>Author</th>
            <th>Publisher</th>
            <th>Publish Date</th>
            <th>Pages</th>
            <th>Price</th>
            <th>Detail</th>
            <th>Delete</th>
        </tr>
    </table>
</body>
</html>
```



```
</tr>
<c:forEach items="${list}" var="b">
  <tr>
    <td>${b.bookId}</td>
    <td>${b.bookName}</td>
    <td>${b.author}</td>
    <td>${b.publisher}</td>
    <td>${b.publishDate}</td>
    <td>${b.pages}</td>
    <td>${b.price}</td>
    <td>
      <a href="DetailBook?bookId=${b.bookId}">detail</a>
    </td>
    <td>
      <a href="DeleteBook?bookId=${b.bookId}"
onclick="return confirm('Sure?')">delete</a>
    </td>
  </tr>
</c:forEach>
</table>
<a href="insertBook.jsp">Add New Book</a>
</body>
</html>
```

Chạy ứng dụng lên chúng ta sẽ có kết quả của trang load data

Insert title here x +

localhost:8080/Webservice\_Client/

### LIST BOOKS

Book Id	Book Name	Author	Publisher	Publish Date	Pages	Price	Detail	Delete
3	Lập trình Android	Lê Bá Đức	NXB Tuổi Trẻ	Tue May 26 00:00:00 ICT 2020	2000	400000.0	<a href="#">detail</a>	<a href="#">delete</a>

[Add New Book](#)

## CHỨC NĂNG INSERT DATA

### Bước 1: Tạo liên kết tới trang insert từ trang listBooks.jsp

```
<a href="insertBook.jsp">Add New Book</a>
```

### Bước 2: Tạo trang insertBook.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
```



```
<body>
  <h1>INSERT NEW BOOK</h1>
  <h3 style="color:red">${error}</h3>

  <form action="InsertBook" method="post">
    <table>
      <tr>
        <td>Book Name:</td>
        <td><input type="text" name="bookName"/></td>
      </tr>
      <tr>
        <td>Author:</td>
        <td><input type="text" name="author"/></td>
      </tr>
      <tr>
        <td>Publisher:</td>
        <td><input type="text" name="publisher"/></td>
      </tr>
      <tr>
        <td>Publish Date:</td>
        <td><input type="date" name="publishDate"/></td>
      </tr>
      <tr>
        <td>Pages:</td>
        <td><input type="number" name="pages"/></td>
      </tr>
      <tr>
        <td>Price:</td>
        <td><input type="number" name="price"/></td>
      </tr>
      <tr>
        <td></td>
        <td>
          <input type="submit" value="Insert"/>
          <input type="reset" value="Clear"/>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

### Bước 3: Tạo servlet InsertBook

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    // TODO Auto-generated method stub
    request.setCharacterEncoding("UTF-8");
    String bookName = request.getParameter("bookName");
    String author = request.getParameter("author");
    String publisher = request.getParameter("publisher");
    Date publishDate = null;
    SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
    try {
        publishDate = sf.parse(request.getParameter("publishDate"));
    }
```





```
} catch (ParseException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
  
int pages = Integer.parseInt(request.getParameter("pages"));  
double price = Double.parseDouble(request.getParameter("price"));  
  
Book book = new Book();  
book.setBookName(bookName);  
book.setAuthor(author);  
book.setPublisher(publisher);  
book.setPublishDate(publishDate);  
book.setPages(pages);  
book.setPrice(price);  
  
//goi webservice de insert  
Gson son = new Gson();  
String jsonDataBook = son.toJson(book);  
  
Client client = Client.create();  
WebResource resource =  
client.resource("http://localhost:8080/Webservice_Session05_DB/rest/book-  
service/insertBook");  
ClientResponse clientResponse =  
resource.type("application/json").post(ClientResponse.class, jsonDataBook);  
String result = clientResponse.getEntity(String.class);  
Boolean bl = son.fromJson(result, Boolean.class);  
  
if(bl) {  
    request.setAttribute("success", "Insert successfully!");  
    request.getRequestDispatcher("LoadBooks").forward(request, response);  
}else {  
    request.setAttribute("error", "Insert failed1");  
    request.getRequestDispatcher("insertBook.jsp").forward(request,  
response);  
}  
}
```

Chạy thử chúng ta có kết quả thực hiện chức năng insert

Insert title here x +

localhost:8080/Webservice\_Client/insertBook.jsp

### INSERT NEW BOOK

Book Name:

Author:

Publisher:

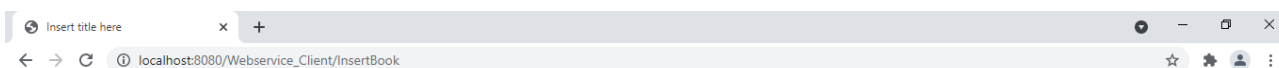
Publish Date:  ☐

Pages:

Price:







## LIST BOOKS

Insert successfully!

Book Id	Book Name	Author	Publisher	Publish Date	Pages	Price	Detail	Delete
3	Lập trình Android	Lê Bá Đức	NXB Tuổi Trẻ	Tue May 26 00:00:00 ICT 2020	2000	400000.0	<a href="#">detail</a>	<a href="#">delete</a>
4	Lập trình PHP	Phạm Trung Kiên	NXB Tuổi Trẻ	Sun Dec 20 00:00:00 ICT 2020	500	150000.0	<a href="#">detail</a>	<a href="#">delete</a>

[Add New Book](#)

## CHỨC NĂNG XEM CHI TIẾT DỮ LIỆU

### Bước 1: Tạo liên kết để gọi tới servlet

```
<td>  
    <a href="DetailBook?bookId=${b.bookId}">detail</a>  
</td>
```

### Bước 2: Tạo servlet DetailBook

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    // TODO Auto-generated method stub  
    int bookId = Integer.parseInt(request.getParameter("bookId"));  
  
    Client client = Client.create();  
    WebResource resource =  
client.resource("http://localhost:8080/Webservice_Session05_DB/rest/book-  
service/getBookById/"+bookId);  
    resource.accept(MediaType.APPLICATION_JSON);  
    String data = resource.get(String.class);  
  
    Gson son = new Gson();  
    Book book = son.fromJson(data, Book.class);  
  
    request.setAttribute("b", book);  
    request.getRequestDispatcher("detailBook.jsp").forward(request, response);  
}
```

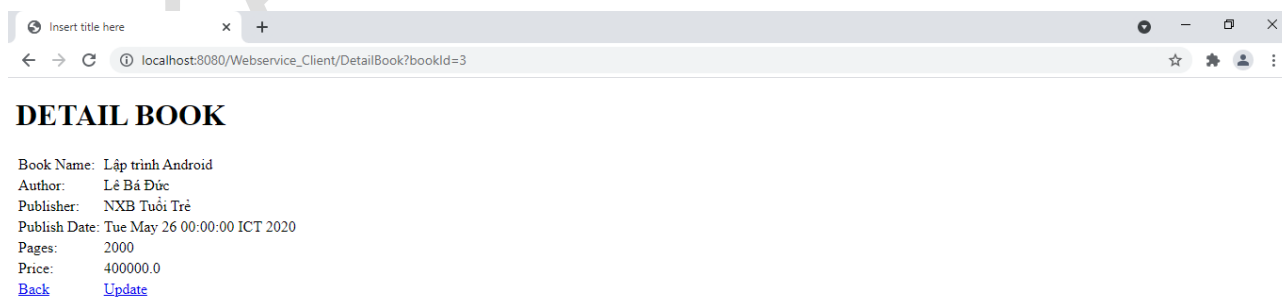
### Bước 3: Tạo trang detailBook.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Insert title here</title>
```



```
</head>
<body>
    <h1>DETAIL BOOK</h1>
    <table>
        <tr>
            <td>Book Name:</td>
            <td>${b.bookName}</td>
        </tr>
        <tr>
            <td>Author:</td>
            <td>${b.author}</td>
        </tr>
        <tr>
            <td>Publisher:</td>
            <td>${b.publisher}</td>
        </tr>
        <tr>
            <td>Publish Date:</td>
            <td>${b.publishDate}</td>
        </tr>
        <tr>
            <td>Pages:</td>
            <td>${b.pages}</td>
        </tr>
        <tr>
            <td>Price:</td>
            <td>${b.price}</td>
        </tr>
        <tr>
            <td><a href="index.jsp">Back</a></td>
            <td><a href="PreUpdateBook?bookId=${b.bookId}">Update</a></td>
        </tr>
    </table>
</body>
</html>
```

Chạy thử chúng ta có kết quả thực hiện chức năng xem chi tiết dữ liệu



## CHỨC NĂNG DELETE DỮ LIỆU

### Bước 1: Tạo liên kết để gọi tới servlet

```
<td>  
    <a href="DeleteBook?bookId=${b.bookId}" onclick="return  
confirm('Sure?')">delete</a>  
</td>
```

### Bước 2: Tạo servlet DeleteBook

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    // TODO Auto-generated method stub  
    int bookId = Integer.parseInt(request.getParameter("bookId"));  
  
    Client client = Client.create();  
    WebResource resource =  
client.resource("http://localhost:8080/Webservice_Session05_DB/rest/book-  
service/deleteBook/"+bookId);  
    resource.accept(MediaType.APPLICATION_JSON);  
    String data = resource.type(MediaType.APPLICATION_JSON).delete(String.class);  
  
    Gson son = new Gson();  
    Boolean bl = son.fromJson(data, Boolean.class);  
  
    if(bl) {  
        request.setAttribute("success", "Delete successfully!");  
    }else {  
        request.setAttribute("error", "Delete failed!");  
    }  
    request.getRequestDispatcher("LoadBooks").forward(request, response);  
}
```

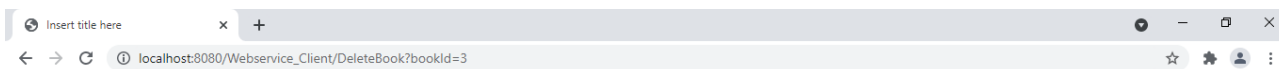
Chạy thử chúng ta có kết quả thực hiện chức năng xóa dữ liệu

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Webservice\_Client/index.jsp'. The main content area has the title 'LIST BOOKS' and a table with the following data:

Book Id	Book Name	Author	Publisher	
3	Lập trình Android	Lê Bá Đức	NXB Tuổi Trẻ	Tue Mi
4	Lập trình PHP	Phạm Trung Kiên	NXB Tuổi Trẻ	Sun Dec 20 00:00:00 ICT 2020   500   150000.0   <a href="#">detail</a>   <a href="#">delete</a>

Below the table is a link 'Add New Book'. A confirmation dialog box is open, displaying the text 'localhost:8080 cho biết Sure?' with 'OK' and 'Hủy' buttons.





## LIST BOOKS

Delete successfully!

Book Id	Book Name	Author	Publisher	Publish Date	Pages	Price	Detail	Delete
4	Lập trình PHP	Phạm Trung Kiên	NXB Tuổi Trẻ	Sun Dec 20 00:00:00 ICT 2020	500	150000.0	<a href="#">detail</a>	<a href="#">delete</a>

[Add New Book](#)

## CHỨC NĂNG UPDATE DỮ LIỆU

### Bước 1: Tạo liên kết để gọi tới servlet

```
<td>  
    <a href="PreUpdateBook?bookId=${b.bookId}">Update</a>  
</td>
```

### Bước 2: Tạo servlet PreUpdateBook

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    // TODO Auto-generated method stub  
    int bookId = Integer.parseInt(request.getParameter("bookId"));  
  
    Client client = Client.create();  
    WebResource resource =  
client.resource("http://localhost:8080/Webservice_Session05_DB/rest/book-  
service/getBookById/"+bookId);  
    resource.accept(MediaType.APPLICATION_JSON);  
    String data = resource.get(String.class);  
  
    Gson son = new Gson();  
    Book book = son.fromJson(data, Book.class);  
  
    request.setAttribute("b", book);  
    request.getRequestDispatcher("updateBook.jsp").forward(request, response);  
}
```

### Bước 3: Tạo trang updateBook.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>  
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Insert title here</title>  
</head>
```



```
<body>
  <h1>UPDATE BOOK</h1>
  <h3 style="color:red">${error}</h3>

  <form action="UpdateBook" method="post">
    <table>
      <tr>
        <td>Book Id:</td>
        <td><input type="text" name="bookId" value="${b.bookId}"
readonly="readonly"/></td>
      </tr>
      <tr>
        <td>Book Name:</td>
        <td><input type="text" name="bookName"
value="${b.bookName}"/></td>
      </tr>
      <tr>
        <td>Author:</td>
        <td><input type="text" name="author"
value="${b.author}"/></td>
      </tr>
      <tr>
        <td>Publisher:</td>
        <td><input type="text" name="publisher"
value="${b.publisher}"/></td>
      </tr>
      <tr>
        <td>Publish Date:</td>
        <td><input type="date" name="publishDate"
value="<fmt:formatDate pattern="yyyy-MM-dd" value="${b.publishDate}"/>"/></td>
      </tr>
      <tr>
        <td>Pages:</td>
        <td><input type="number" name="pages"
value="${b.pages}"/></td>
      </tr>
      <tr>
        <td>Price:</td>
        <td><input type="number" name="price"
value="${b.price}"/></td>
      </tr>
      <tr>
        <td></td>
        <td>
          <input type="submit" value="Update"/>
          <input type="reset" value="Clear"/>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

#### **Bước 4: Tạo servlet UpdateBook**



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    int bookId = Integer.parseInt(request.getParameter("bookId"));
    String bookName = request.getParameter("bookName");
    String author = request.getParameter("author");
    String publisher = request.getParameter("publisher");
    Date publishDate = null;
    SimpleDateFormat sf = new SimpleDateFormat("yyyy-MM-dd");
    try {
        publishDate = sf.parse(request.getParameter("publishDate"));
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    int pages = Integer.parseInt(request.getParameter("pages"));
    double price = Double.parseDouble(request.getParameter("price"));

    Book book = new Book();
    book.setBookId(bookId);
    book.setBookName(bookName);
    book.setAuthor(author);
    book.setPublisher(publisher);
    book.setPublishDate(publishDate);
    book.setPages(pages);
    book.setPrice(price);

    //goi webservice de insert
    Gson son = new Gson();
    String jsonDataBook = son.toJson(book);

    Client client = Client.create();
    WebResource resource =
client.resource("http://localhost:8080/Webservice_Session05_DB/rest/book-
service/updateBook");
    ClientResponse clientResponse =
resource.type("application/json").post(ClientResponse.class, jsonDataBook);
    String result = clientResponse.getEntity(String.class);
    Boolean bl = son.fromJson(result, Boolean.class);

    if(bl) {
        request.setAttribute("success", "Update successfully!");
        request.getRequestDispatcher("LoadBooks").forward(request, response);
    } else {
        request.setAttribute("error", "Update failed!");
        request.getRequestDispatcher("updateBook.jsp").forward(request,
response);
    }
}
```

Chạy thử chúng ta có kết quả chức năng update dữ liệu



Insert title here x +

localhost:8080/Webservice\_Client/PreUpdateBook?bookId=4

## UPDATE BOOK

Book Id:

Book Name:

Author:

Publisher:

Publish Date:

Pages:

Price:

Insert title here x +

localhost:8080/Webservice\_Client/PreUpdateBook?bookId=4

## UPDATE BOOK

Book Id:

Book Name:

Author:

Publisher:

Publish Date:

Pages:

Price:

Insert title here x +

localhost:8080/Webservice\_Client/UpdateBook

## LIST BOOKS

Uupdate successfully!

Book Id	Book Name	Author	Publisher	Publish Date	Pages	Price	Detail	Delete
4	Lập trình PHP update	Phạm Trung Kiên update	NXB Tuổi Trẻ	Sun Dec 20 00:00:00 ICT 2020	500	150000.0	<a href="#">detail</a>	<a href="#">delete</a>

[Add New Book](#)





### 3. Bài tập tự làm

Bài 1. Tạo database: DB\_WebAPI1

Tên bảng: TblEmployee

Tên cột	Kiểu dữ liệu	Mô tả
EmpId	int – tự tăng	Khóa chính
EmpName	Nvarchar(50)	
Gender	Bit	
Birthday	Date	
Address	Nvarchar(200)	
Position	Nvarchar(100)	
Salary	float	

- Thêm một số dữ liệu ban đầu cho bảng này.
- Tạo restul web services cung cấp các phương thức:
  - Get all employees
  - Get employee by id
  - Insert new employee
  - Update employee
  - Delete employee
- Tạo ứng dụng web client để gọi tới các phương thức được cung cấp bởi web services trên

Bài 2: Tạo database như sau

```
create database DB_WebAPI2  
go
```



```
use DB_Class_Student
go
create table Classes(
    ClassId varchar(15) not null primary key,
    ClassName nvarchar(100) unique)

insert into Classes values ('C1808M',N'Lập trình quốc tế ACCPi17 Lớp C1808M')
insert into Classes values ('C1808I',N'Lập trình quốc tế ACCPi17 Lớp C1808I')
insert into Classes values ('C1807G',N'Lập trình quốc tế ACCPi17 Lớp C1807G')
insert into Classes values ('C1807G2H',N'Lập trình quốc tế ACCPi17 Lớp C1807G2H')
insert into Classes values ('C18010M',N'Lập trình quốc tế ACCPi17 Lớp C18010M')

create table Student(
    StuId int identity primary key,
    ClassId varchar(15) foreign key references Classes(ClassId),
    FullName nvarchar(70),
    Gender bit,
    Birthday datetime,
    Address nvarchar(200))

insert into Student values ('C1808M',N'Nguyễn Đức Trọng',1,'2001-05-05',N'Ba Đình - Hà Nội')
insert into Student values ('C1807G',N'Trần Khánh Huyền',0,'2000-11-23',N'Sóc Sơn - Hà Nội')
insert into Student values ('C1808I',N'Nguyễn Thành Nam',1,'2000-12-24',N'Gia Lâm - Hà Nội')

select * from Classes
select * from Student
```

- Tạo restful web service cung cấp các phương thức sau:
  - Get all students
  - Get student by id
  - Insert new student
  - Update student
  - Delete student
- Tạo ứng dụng web client để gọi tới các phương thức được cung cấp bởi web services trên

