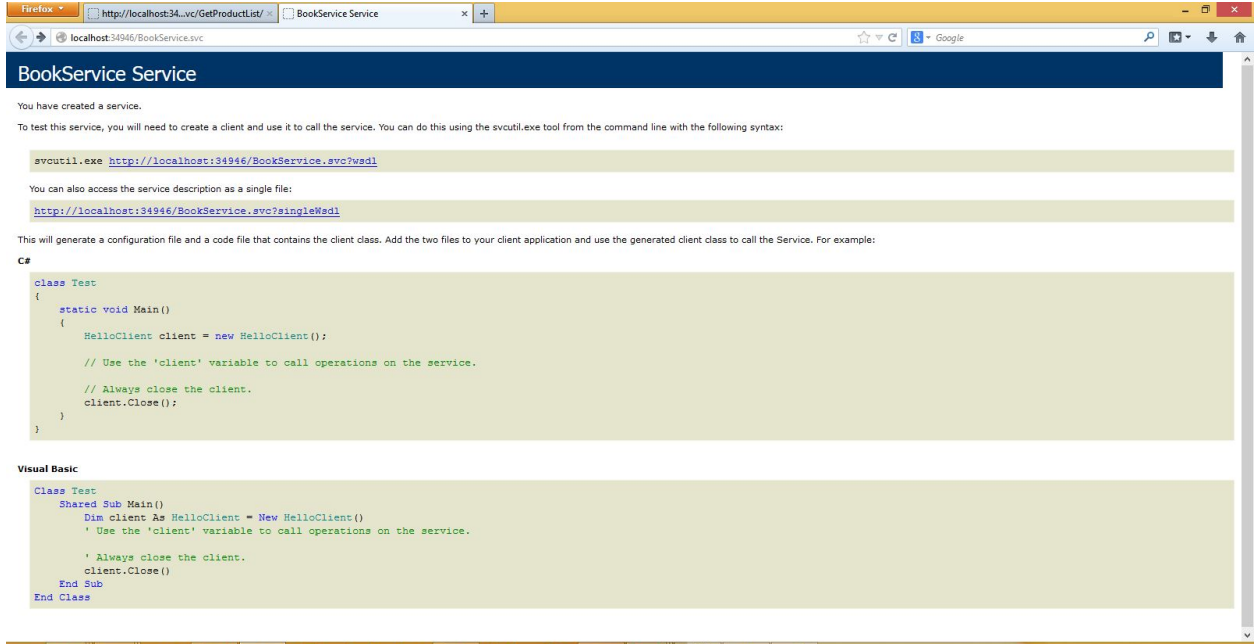
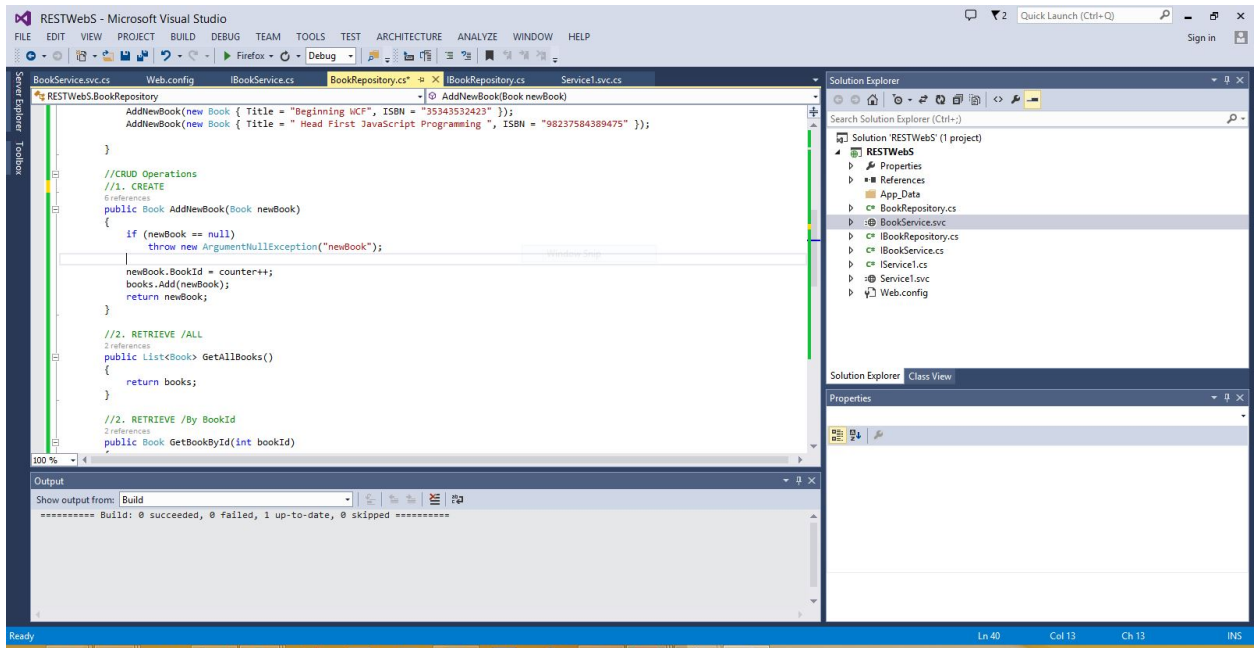
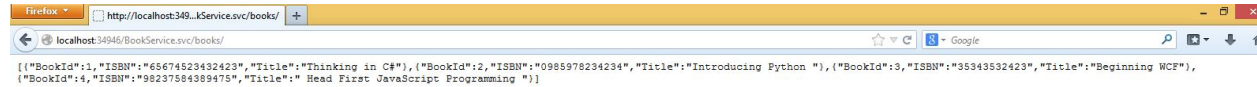


A)REST Web Service

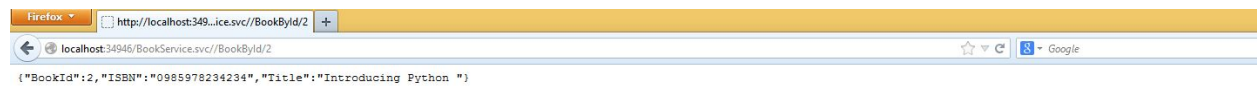


CRUD Operations

Read all resources /books



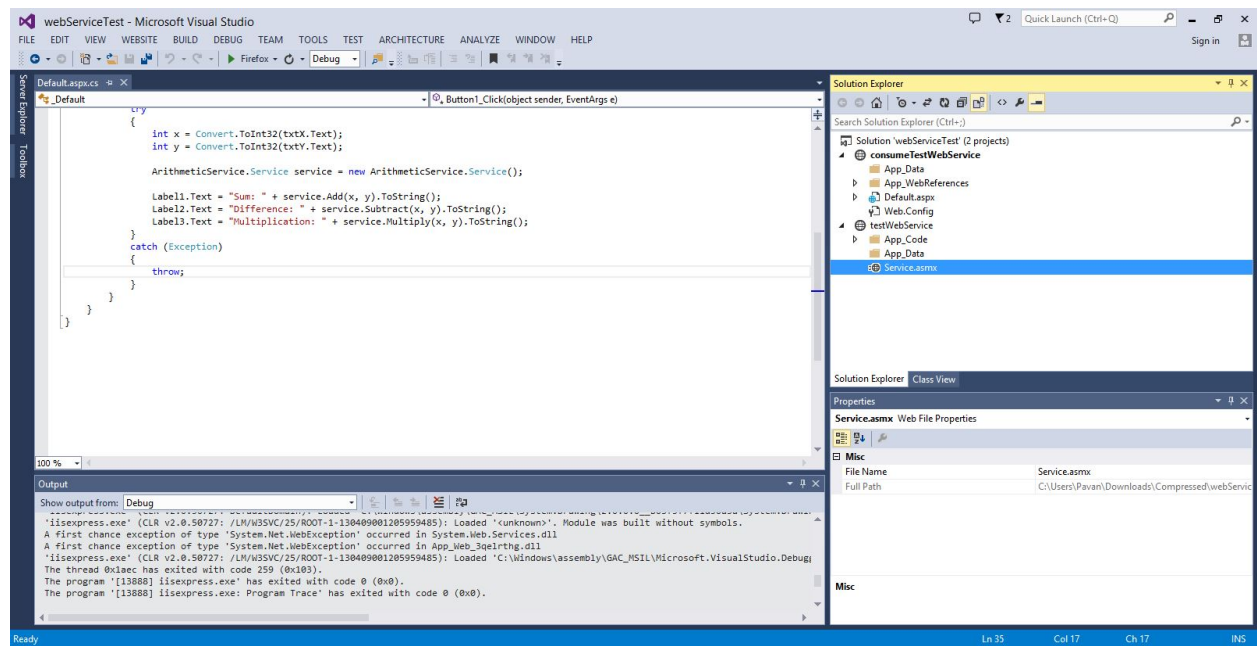
Find resource by ID /BookbyId/(id)



B) SOAP Web Service

A simple module for arithmetic operations

1



Invoking a method from service

The screenshot shows a web browser window with the address bar displaying `localhost:1972/Service.aspx?op=Multiply`. The page title is "Service". Below the title, there is a link: "Click [here](#) for a complete list of operations."

The main section is titled "Multiply" and contains a "Test" section. It instructs the user: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below this, there is a form with two input fields labeled "X:" and "Y:", and an "Invoke" button.

Below the form, there is a section titled "SOAP 1.1" which states: "The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values."

The sample SOAP 1.1 request is shown in a code block:

```
POST /Service.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Multiply"

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Multiply xmlns="http://tempuri.org/">
      <input>
        <x>
      </x>
    </Multiply>
  </soap:Body>
</soap:Envelope>
```

The sample SOAP 1.1 response is shown in a code block:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <MultiplyResponse xmlns="http://tempuri.org/">
      <MultiplyResult>
      </MultiplyResult>
    </MultiplyResponse>
  </soap:Body>
</soap:Envelope>
```

C) Mobile Client

jQuery Mobile

Implemented mobile client for above mentioned REST service

The screenshot shows the JSFiddle web application interface. The left sidebar contains a "Frameworks & Extensions" section with "No-Library (pure JS)" selected. Below it are "Fiddle Options", "External Resources", "Languages", "Ajax Requests", and "Legal, Credits and Links".

The main editor area is divided into three panes: HTML, CSS, and JavaScript. The HTML pane contains a table structure with the following code:

```
<table border="1" id="books">
  <!-- Make a Header Row -->
  <tr>
    <td><b>Book ID</b></td>
    <td><b>ISBN</b></td>
    <td><b>Title</b></td>
  </tr>
</table>
```

The JavaScript pane contains the following code:

```
// GET ALL
function GetAllBooks()
{
  $.ajax({
    type: "GET",
    url: "http://localhost:34946/BookService.svc/books",
    contentType: "json",
    dataType: "json",
    success: function (data) {
      $.each(data, function (key, value) {
        //stringify
        var jsonData = JSON.stringify(value);
        //Parse JSON
        var objData = $.parseJSON(jsonData);
        var bookId = objData.BookId;
        var isbn = objData.ISBN;
        var title = objData.Title;
        $('tr>td' + bookId + '</td>' + isbn + '</td>' + title + '</td>').appendTo('#books');
      });
    },
    error: function (xhr) {
      alert(xhr.responseText);
    }
  });
}
```

The CSS pane is empty. The "Result" pane on the right shows a table with the following data:

| Book ID | ISBN | Title |
|---------|------|-------|
|---------|------|-------|