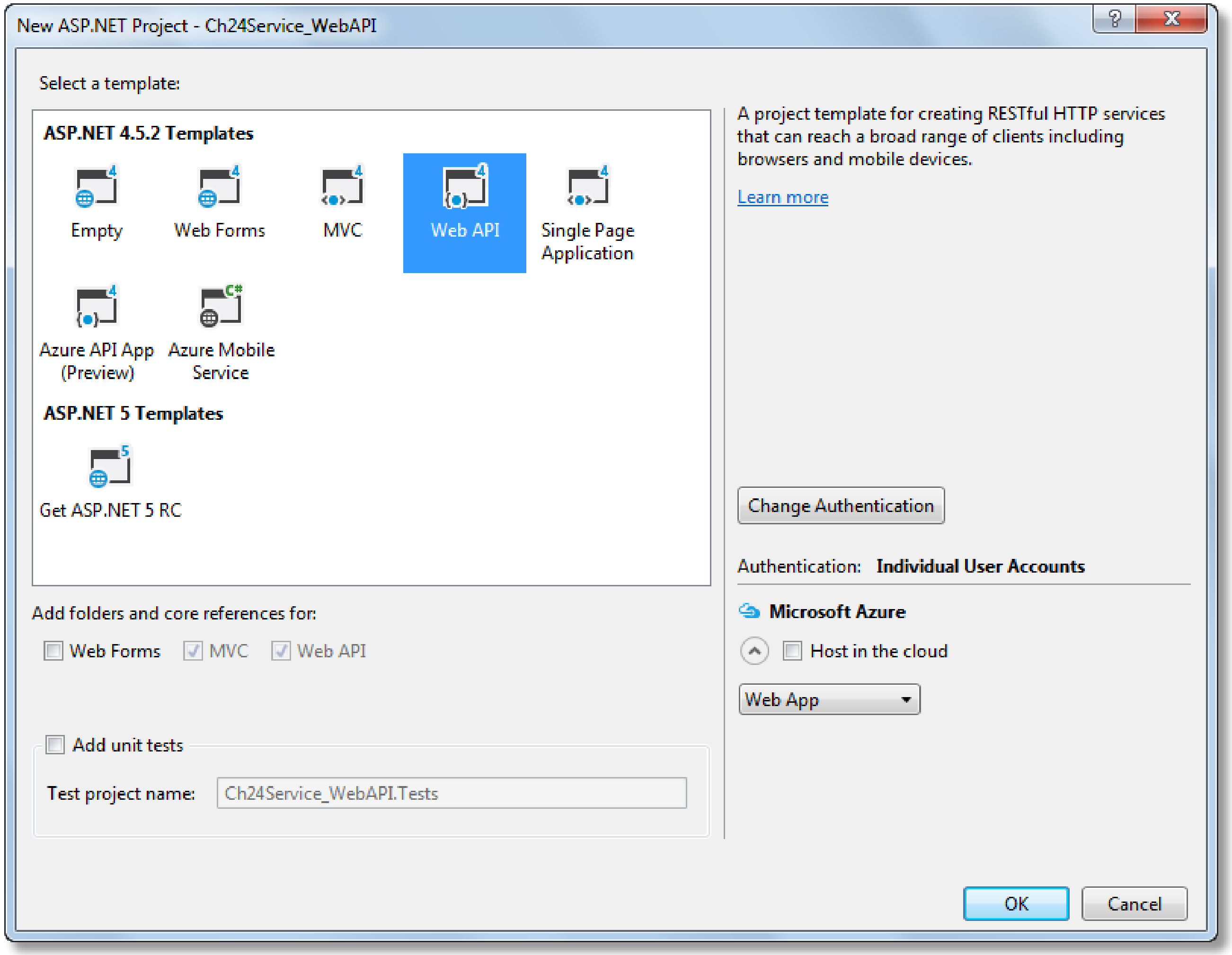
How to create and use Web API services

The New ASP.NET Project dialog box



New ASP.NET Project - Ch24Service_WebAPI

Select a template:

ASP.NET 4.5.2 Templates

ASP.NET 5 Templates

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Learn more

Authentication: **Individual User Accounts**

Microsoft Azure

☐ Host in the cloud

Web App

Add folders and core references for:

☐ Web Forms ☑ MVC ☑ Web API

☐ Add unit tests

Test project name: Ch24Service_WebAPI.Tests

How to start a new Web API service

- Select the File→New→Project command, select the Web template group, and select the ASP.NET Web Application template.
- Enter a name and location for the service application, and click OK.
- In the New ASP.NET Project dialog box, select the Web API template and click OK.

How to adjust the initial files and folders for the Edit Categories page

- Right-click the App_Data folder and add the Halloween.mdf file.
- Right-click the Models folder and add class files named Category.cs and CategoryDB.cs.
- The CategoryDB class is the data access class.
- To add a new controller file to the Controllers folder, right-click the folder and select Add→Controller.
- In the Add Controller dialog box, name your controller, select Empty API controller in the Template drop-down list, and click Add.
- Add a connection string for the Halloween database to the Web.config file.

The CategoriesController.cs file

```
namespace Ch24Service_WebAPI.Controllers
{
    public class CategoriesController : ApiController
    {
        CategoryDB data;
        public CategoriesController() {
            this.data = new CategoryDB();
        }
        // GET: api/categories
        public IEnumerable<Category> GetCategories() {
            //System.Threading.Thread.Sleep(3000);
            return data.GetCategories();
        }
        // GET: api/categories/masks
        public Category GetCategoryById(string id) {
            return data.GetCategoryById(id);
        }
        // GET: api/categories/?name=Masks
        public IEnumerable<Category>
            GetCategoriesByShortName(string name) {
            return data.GetCategoriesByShortName(name);
        }
```

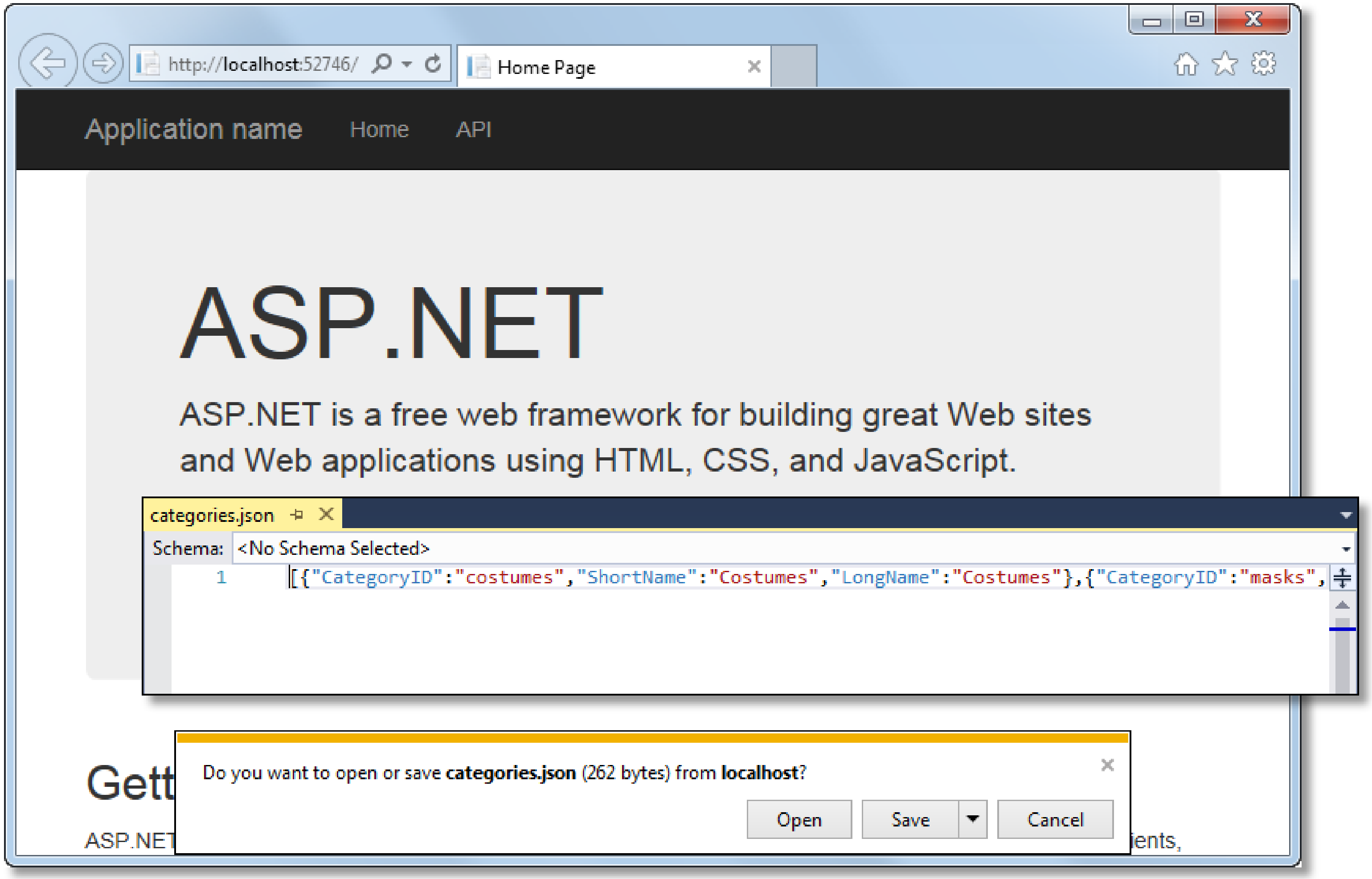
The CategoriesController.cs file (cont.)

```
        // POST(Insert): api/categories
        public int PostCategory([FromBody]Category value)
        {
            return data.InsertCategory(value);
        }
        // PUT(Update): api/categories/masks
        public int PutCategory(string id, [FromBody]Category value)
        {
            value.CategoryID = id;
            return data.UpdateCategory(value);
        }
        // DELETE: api/categories/masks
        public int DeleteCategory(string id)
        {
            Category value = new Category() { CategoryID = id };
            return data.DeleteCategory(value);
        }
    }
}
```

HTTP verbs and corresponding methods and URLs in the Category service

Verb	Action method	URL
GET	GetCategories()	/api/categories
GET	GetCategoryById(string id)	/api/categories/masks
GET	GetCategoriesByShortName(string name)	/api/categories/?name=Masks
POST	PostCategory(Category value)	/api/categories
PUT	PutCategory(string id, Category value)	/api/categories/masks
DELETE	DeleteCategory(string id)	/api/categories/masks

Testing the GET methods in a browser



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

categories.json

Schema: <No Schema Selected>

```
[{"CategoryID":"costumes","ShortName":"Costumes","LongName":"Costumes"},{"CategoryID":"masks",
```

Do you want to open or save **categories.json** (262 bytes) from **localhost**?

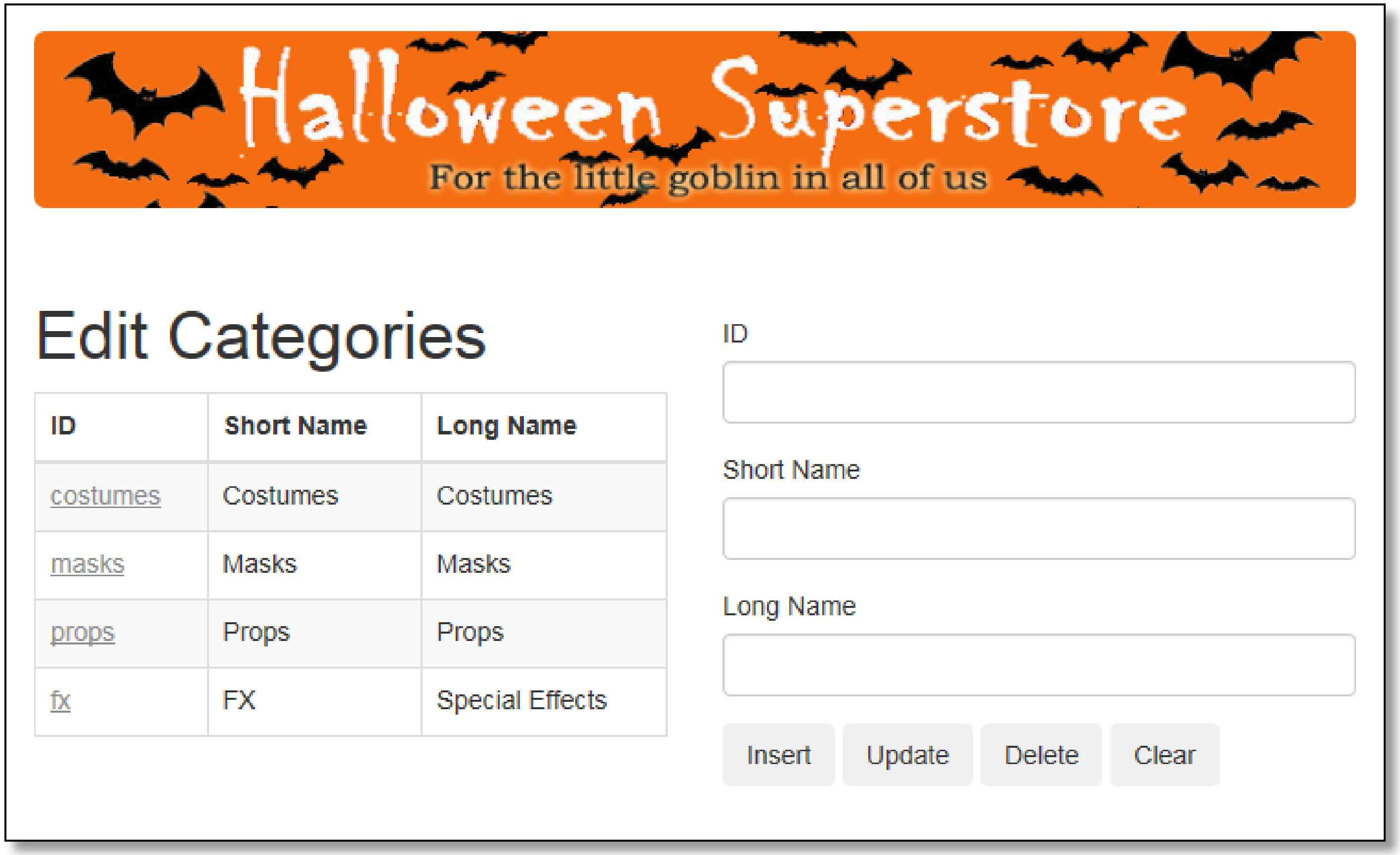
How to test the GET methods of a Web API service

- In Visual Studio, click the browser name in the Standard toolbar or press F5. Then, a home page like the one above is displayed in the browser.
- Add one of the GET URLs shown in the table to the URL in the address bar of the browser and press Enter.
- In IE, click the Open button to see the JSON response from the server in Visual Studio. Or, click the Save button to download the JSON and then view it in any text editor.
- In Chrome or Firefox, the response from the server displays as XML in the browser.

How to view information about a Web API service

- Click the “API” link at the top right of the home page shown above.
- This will display a Help page that lists basic information about your service.

A Categories page when jQuery is used to consume the Web API service



Edit Categories

ID	Short Name	Long Name
costumes	Costumes	Costumes
masks	Masks	Masks
props	Props	Props
fx	FX	Special Effects

ID

Short Name

Long Name

The head element and the column divs that contain the category data

```
<head id="Head1" runat="server">
    <!-- Bootstrap and jQuery link and script tags here -->
    <script src="Scripts/webapi.js"></script>
</head>

<div class="col-sm-6 table-responsive">
  <h1>Edit Categories</h1>
  <table id="categories" class="table table-bordered table-striped">
    <thead>
        <tr>
            <th>ID</th>
            <th>Short Name</th>
            <th>Long Name</th>
        </tr>
    </thead>
    <tbody></tbody>
  </table>
  <span id="message">Loading...</span>
</div>
```

The head element and the column divs that contain the category data (cont.)

```
<div id="details" class="col-sm-6">
  <div class="form-group">
    <label class="control-label">ID</label>
    <input type="text" id="id" name="CategoryID"
           class="form-control" />
  </div>
  <div class="form-group">
    <label class="control-label">Short Name</label>
    <input type="text" id="short" name="ShortName"
           class="form-control" />
  </div>
  <div class="form-group">
    <label class="control-label">Long Name</label>
    <input type="text" id="long" name="LongName"
           class="form-control" />
  </div>
```

The head element and the column divs that contain the category data (cont.)

```
  <div class="form-group">
    <input type="button" value="Insert" class="btn"
           onclick="insertCat();" />
    <input type="button" value="Update" class="btn"
           onclick="updateCat();" />
    <input type="button" value="Delete" class="btn"
           onclick="deleteCat();" />
    <input type="button" value="Clear" class="btn"
           onclick="clearAll();" />
  </div>
</div>
```

The JavaScript and jQuery in the webapi.js

```
var api = "http://localhost:52746/api/categories/";

$(document).ready(function () {
    displayCategories();
    $(document.body).on('click', 'a', function (e) {
        e.preventDefault();
        findCategory($(this).attr("href"));
    });
});
```

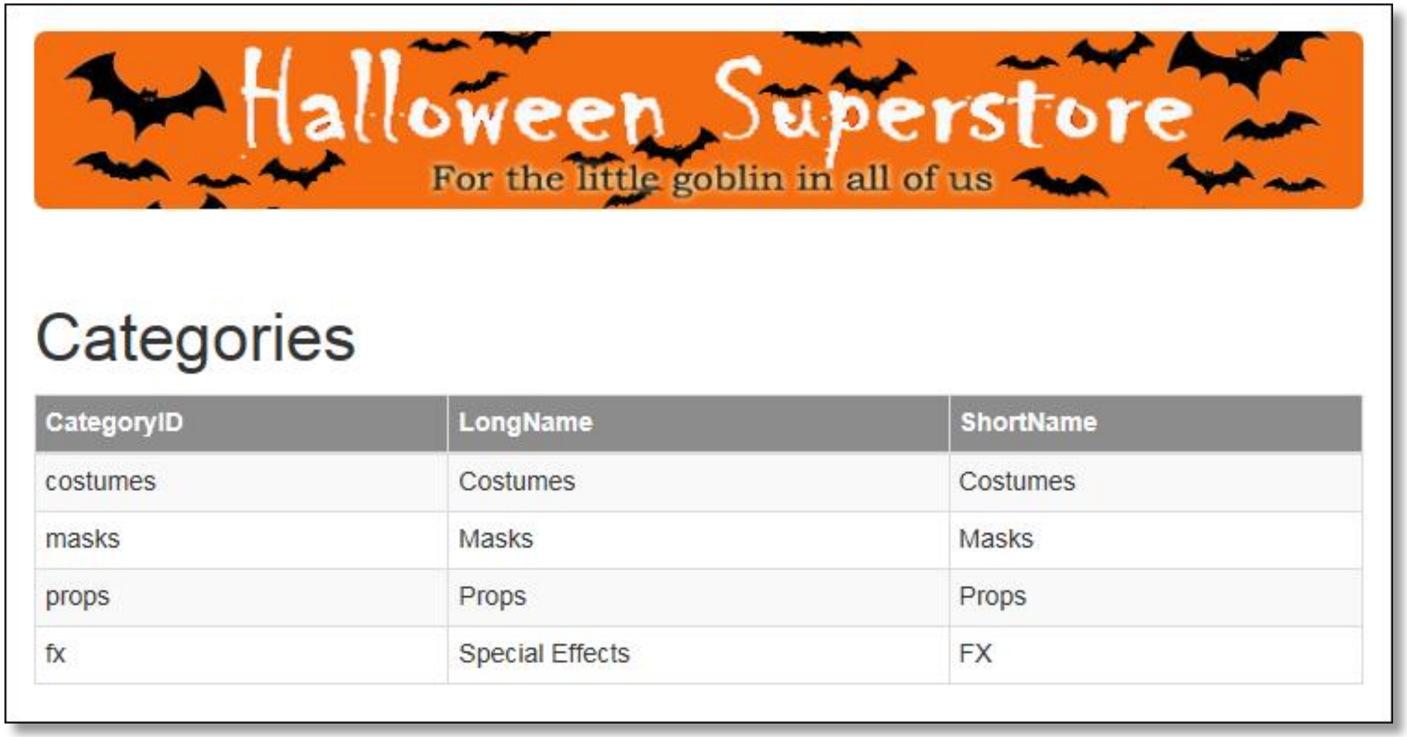
The JavaScript and jQuery in the webapi.js (cont.)

```
function displayCategories() {
    $.getJSON(api, function (data) {
        var rows = "";
        $.each(data, function (key, val) {
            rows += "<tr><td><a href=" + api
                    + val.CategoryID + ">"
                    + val.CategoryID + "</a></td>";
            rows += "<td>" + val.ShortName + "</td>";
            rows += "<td>" + val.LongName + "</td></tr>";
        });
        $('#categories > tbody tr').remove();
        $('#categories > tbody').append(rows);
        clearAll();
    })
    .fail(showError);
}
```

The JavaScript and jQuery in the webapi.js (cont.)

```
function findCategory(href) {
    $('#message').html("Finding...");
    $.getJSON(href, function (data) {
        $('#id').val(data.CategoryID);
        $("#id").attr("disabled", "disabled");
        $('#short').val(data.ShortName);
        $('#long').val(data.LongName);
        $('#message').text("");
    })
    .fail(showError);
}
function insertCat() {
    $('#message').text("Inserting...");
    $.ajax({
        type: 'POST',
        url: api,
        data: $('#form1').serialize(),
        dataType: "json",
        success: displayCategories,
        error: showError
    });
} // the updateCat, deleteCat, and other functions follow
```

A Categories page when C# is used to consume the Web API service



Categories

CategoryID	LongName	ShortName
costumes	Costumes	Costumes
masks	Masks	Masks
props	Props	Props
fx	Special Effects	FX

The code-behind file for the page

```
...
using System.Net;
using System.IO;
using System.Data;

public partial class ServerSide : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //configure web request object
        string url =
            "http://localhost:52746/api/categories/";
        HttpWebRequest request =
            (HttpWebRequest)WebRequest.Create(url);
        request.Method = "GET";
        request.ContentType =
            "text/xml; encoding='utf-8'";
```

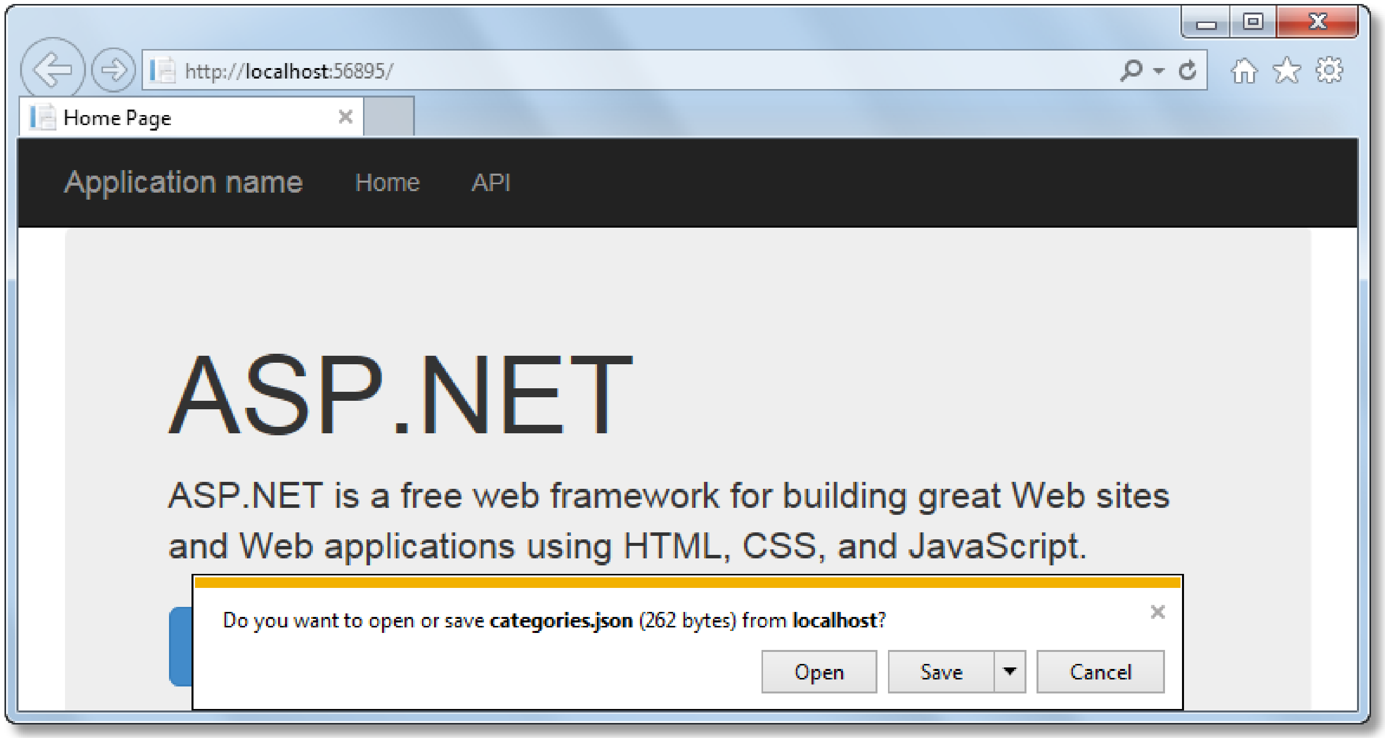
The code-behind file for the page (cont.)

```
        //send request, get xml response
        //and convert to stream
        WebResponse response = request.GetResponse();
        Stream stream = response.GetResponseStream();

        //read stream into a dataset
        DataSet ds = new DataSet();
        ds.ReadXml(stream);

        //bind dataset to gridview
        grdCategories.DataSource = ds.Tables[0];
        grdCategories.DataBind();
    }
}
```

Test and consume the Categories Web API service

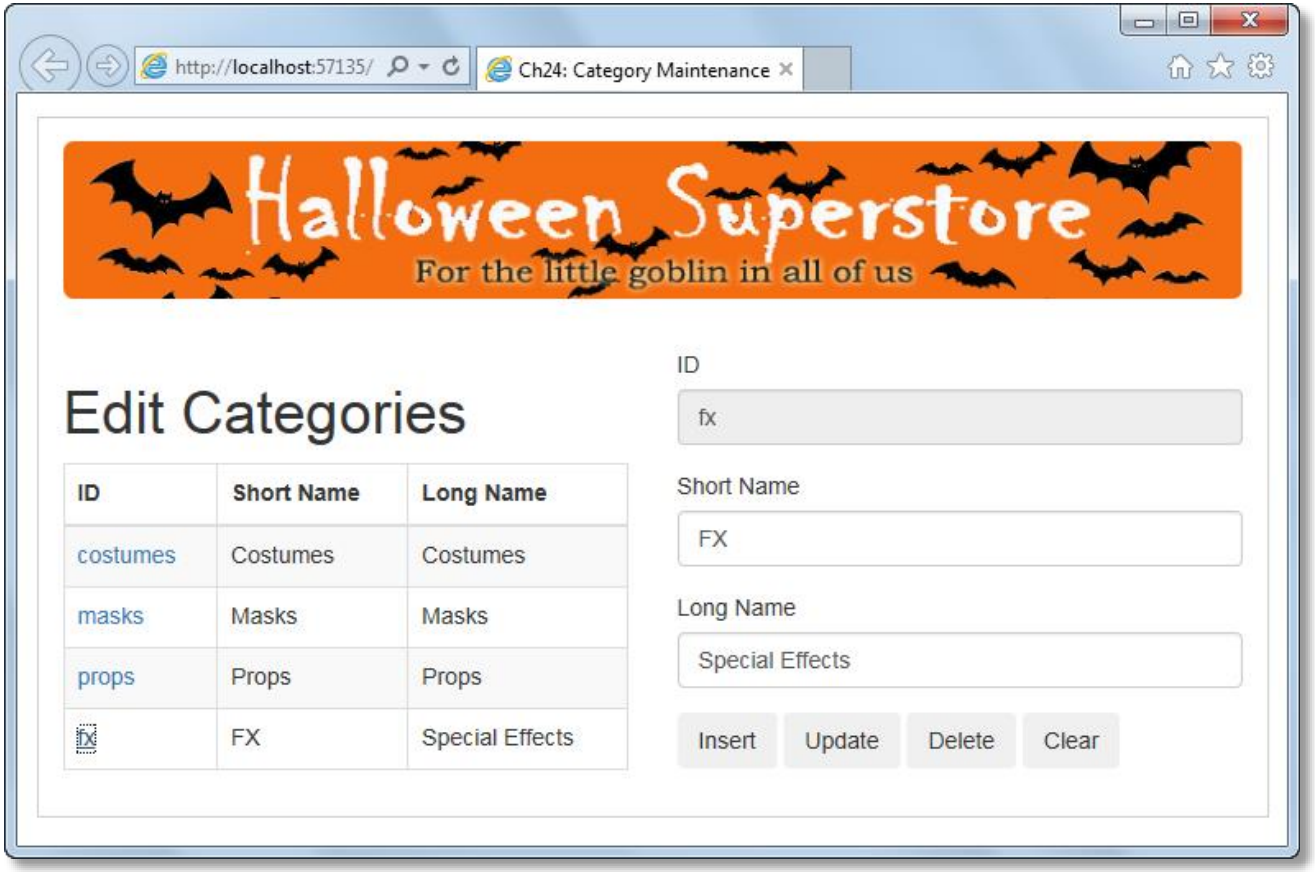


ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Do you want to open or save **categories.json** (262 bytes) from **localhost**?

Test and consume the Categories Web API service (cont.)


http://localhost:57135/
Ch24: Category Maintenance
Halloween Superstore
For the little goblin in all of us

Edit Categories

ID	Short Name	Long Name
costumes	Costumes	Costumes
masks	Masks	Masks
props	Props	Props
fx	FX	Special Effects

ID	fx
Short Name	FX
Long Name	Special Effects