

Call the web API with JavaScript

```
using Microsoft.EntityFrameworkCore;
using ToDoApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllers();
builder.Services.AddDbContext<TodoContext>(opt => opt.UseInMemoryDatabase("TodoList"));

builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseDefaultFiles(); // URL Rewriter: search wwwroot for default.html/, index.html/
app.UseStaticFiles(); // Serve files outside the web root (wwwroot)

app.UseHttpsRedirection();
app.UseAuthorization();

app.MapControllers();

app.Run();
```

Solution Explorer



Search Solution Explorer (Ctrl+;) 🔍

🔒 Solution 'ToDoApi' (1 of 1 project)

🔒 **ToDoApi**

- ▶ Connected Services
- ▶ Dependencies
- ▶ Properties
 - ✓ launchSettings.json
- ▶ wwwroot
 - ▶ css
 - + site.css
 - ▶ js
 - + site.js
 - + index.html

- ▶ Controllers
- ▶ Models
- ▶ appsettings.json
- ▶ ✓ Program.cs
- ▶ WeatherForecast.cs

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>To-do CRUD</title>
  <link rel="stylesheet" href="css/site.css" />
</head>
<body>
  <h1>To-do CRUD</h1>
  <h3>Add</h3>
  <form action="javascript:void(0);" method="POST" onsubmit="addItem()">
    <input type="text" id="add-name" placeholder="New to-do">
    <input type="submit" value="Add">
  </form>

  <div id="editForm">
    <h3>Edit</h3>
    <form action="javascript:void(0);" onsubmit="updateItem()">
      <input type="hidden" id="edit-id">
      <input type="checkbox" id="edit-isComplete">
      <input type="text" id="edit-name">
      <input type="submit" value="Save">
      <a onclick="closeInput()" aria-label="Close">&#10006;</a>
    </form>
  </div>

  <p id="counter"></p>
```

```
</div>
```

```
<p id="counter"></p>
```

```
<table>
```

```
  <tr>
```

```
    <th>Is Complete?</th>
```

```
    <th>Name</th>
```

```
    <th></th>
```

```
    <th></th>
```

```
  </tr>
```

```
  <tbody id="todos"></tbody>
```

```
</table>
```

```
<script src="js/site.js" asp-append-version="true"></script>
```

```
<script type="text/javascript">
```

```
  getItems();
```

```
</script>
```

```
</body>
```

```
</html>
```



```
input[type='submit'], button, [aria-label] {  
    cursor: pointer;  
}  
  
#editForm {  
    display: none;  
}  
  
table {  
    font-family: Arial, sans-serif;  
    border: 1px solid;  
    border-collapse: collapse;  
}  
  
th {  
    background-color: #f8f8f8;  
    padding: 5px;  
}  
  
td {  
    border: 1px solid;  
    padding: 5px;  
}
```

```
const uri = 'api/todoitems';  
let todos = [];  
  
function getItems() {  
  fetch(uri)  
    .then(response => response.json())  
    .then(data => _displayItems(data))  
    .catch(error => console.error('Unable to get items.', error));  
}
```

```
function _displayItems(data) {
  const tBody = document.getElementById('todos');
  tBody.innerHTML = '';

  _displayCount(data.length);

  const button = document.createElement('button');

  data.forEach(item => {
    let isCompleteCheckbox = document.createElement('input');
    isCompleteCheckbox.type = 'checkbox';
    isCompleteCheckbox.disabled = true;
    isCompleteCheckbox.checked = item.isComplete;

    let editButton = button.cloneNode(false);
    editButton.innerText = 'Edit';
    editButton.setAttribute('onclick', `displayEditForm(${item.id})`);

    let deleteButton = button.cloneNode(false);
    deleteButton.innerText = 'Delete';
    deleteButton.setAttribute('onclick', `deleteItem(${item.id})`);

    let tr = tBody.insertRow();

    let td1 = tr.insertCell(0);
    td1.appendChild(isCompleteCheckbox);

    let td2 = tr.insertCell(1);
    let textNode = document.createTextNode(item.name);
    td2.appendChild(textNode);

    let td3 = tr.insertCell(2);
    td3.appendChild(editButton);

    let td4 = tr.insertCell(3);
    td4.appendChild(deleteButton);
  });

  todos = data;
}
```



```
function deleteItem(id) {  
  fetch(`${uri}/${id}`, {  
    method: 'DELETE'  
  })  
    .then(() => getItems())  
    .catch(error => console.error('Unable to delete item.', error));  
}
```

```
function addItem() {  
    const addNameTextbox = document.getElementById('add-name');  
  
    const item = {  
        isComplete: false,  
        name: addNameTextbox.value.trim()  
    };  
  
    fetch(uri, {  
        method: 'POST',  
        headers: {  
            'Accept': 'application/json',  
            'Content-Type': 'application/json'  
        },  
        body: JSON.stringify(item)  
    })  
        .then(response => response.json())  
        .then(() => {  
            getItems();  
            addNameTextbox.value = '';  
        })  
        .catch(error => console.error('Unable to add item.', error));  
}
```

```
function updateItem() {  
  const itemId = document.getElementById('edit-id').value;  
  const item = {  
    id: parseInt(itemId, 10),  
    isComplete: document.getElementById('edit-isComplete').checked,  
    name: document.getElementById('edit-name').value.trim()  
  };  
  
  fetch(`${uri}/${itemId}`, {  
    method: 'PUT',  
    headers: {  
      'Accept': 'application/json',  
      'Content-Type': 'application/json'  
    },  
    body: JSON.stringify(item)  
  })  
    .then(() => getItems())  
    .catch(error => console.error('Unable to update item.', error));  
  
  closeInput();  
  
  return false;  
}
```



```
-function closeInput() {  
  document.getElementById('editForm').style.display = 'none';  
}  
  
-function displayEditForm(id) {  
  const item = todos.find(item => item.id === id);  
  
  document.getElementById('edit-name').value = item.name;  
  document.getElementById('edit-id').value = item.id;  
  document.getElementById('edit-isComplete').checked = item.isComplete;  
  document.getElementById('editForm').style.display = 'block';  
}  
  
-function _displayCount(itemCount) {  
  const name = (itemCount === 1) ? 'to-do' : 'to-dos';  
  
  document.getElementById('counter').innerText = `${itemCount} ${name}`;  
}
```


To-do CRUD

Add

0 to-dos

Is Complete?	Name
--------------	------

To-do CRUD

Add

0 to-dos

Is Complete?	Name
--------------	------

To-do CRUD

Add

1 to-do

Is Complete? Name			
<input type="checkbox"/>	Meeting	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

To-do CRUD

Add

Add

Edit

☒

Meeting at Board room

Save

×

1 to-do

Is Complete?		Name	
<input type="checkbox"/>		Meeting	<div>EditDelete</div>

To-do CRUD

Add

1 to-do

Is Complete?				Name	
<input checked="" data-bbox="136 1273 208 1345" type="checkbox"/>		Meeting at Board room		<input data-bbox="1619 1257 1798 1369" type="button" value="Edit"/>	<input data-bbox="1861 1257 2119 1369" type="button" value="Delete"/>