

Browser link in ASP.NET Core

Article • 06/07/2024

By [Nicolò Carandini](#) and [Tom Dykstra](#)

Browser link is a Visual Studio feature. It creates a communication channel between the development environment and one or more web browsers. Use browser link to:

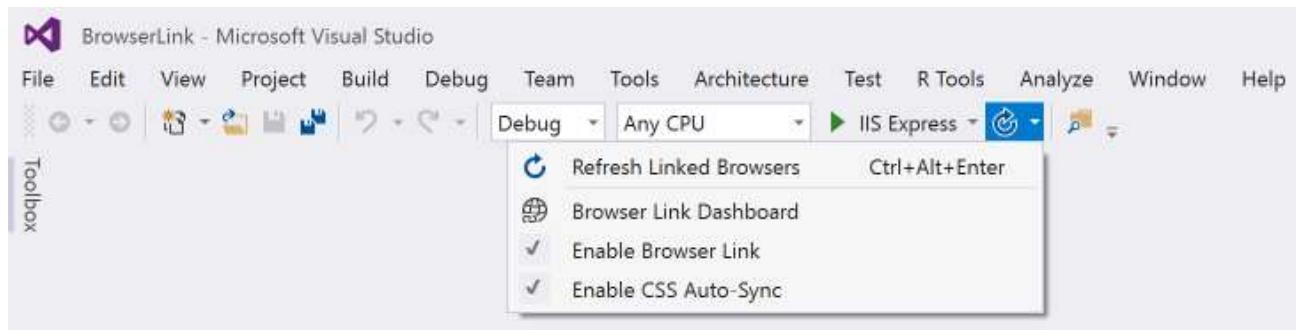
- Refresh your web app in several browsers at once.
- Test across multiple browsers with specific settings such as screen sizes.
- Select UI elements in browsers in real-time, see what markup and source it's correlated to in Visual Studio.
- Conduct real-time browser test automation.

Runtime compilation vs. hot reload

Use browser Link with [runtime compilation](#) or [hot reload](#) to see the effect of run-time changes in Razor (.cshtml) files. We recommend hot reload.

How to use browser link

When you have an ASP.NET Core project open, Visual Studio shows the browser link toolbar control next to the **Debug Type** toolbar control:

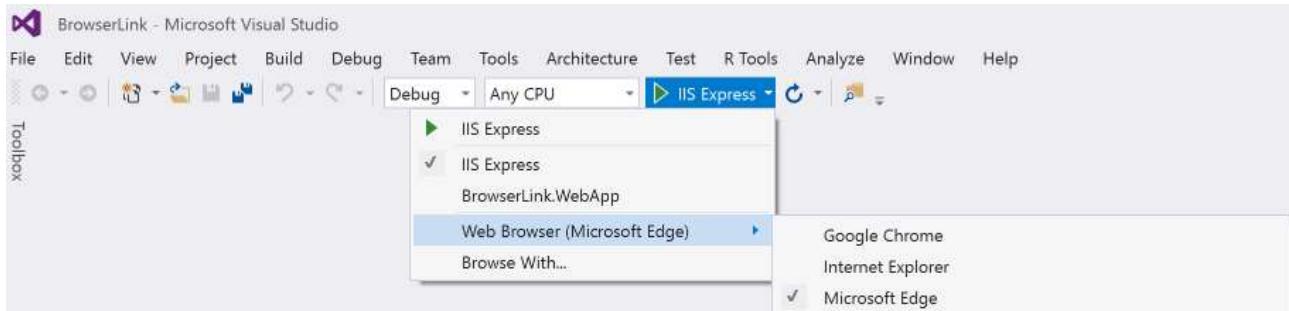


From the browser link toolbar control, you can:

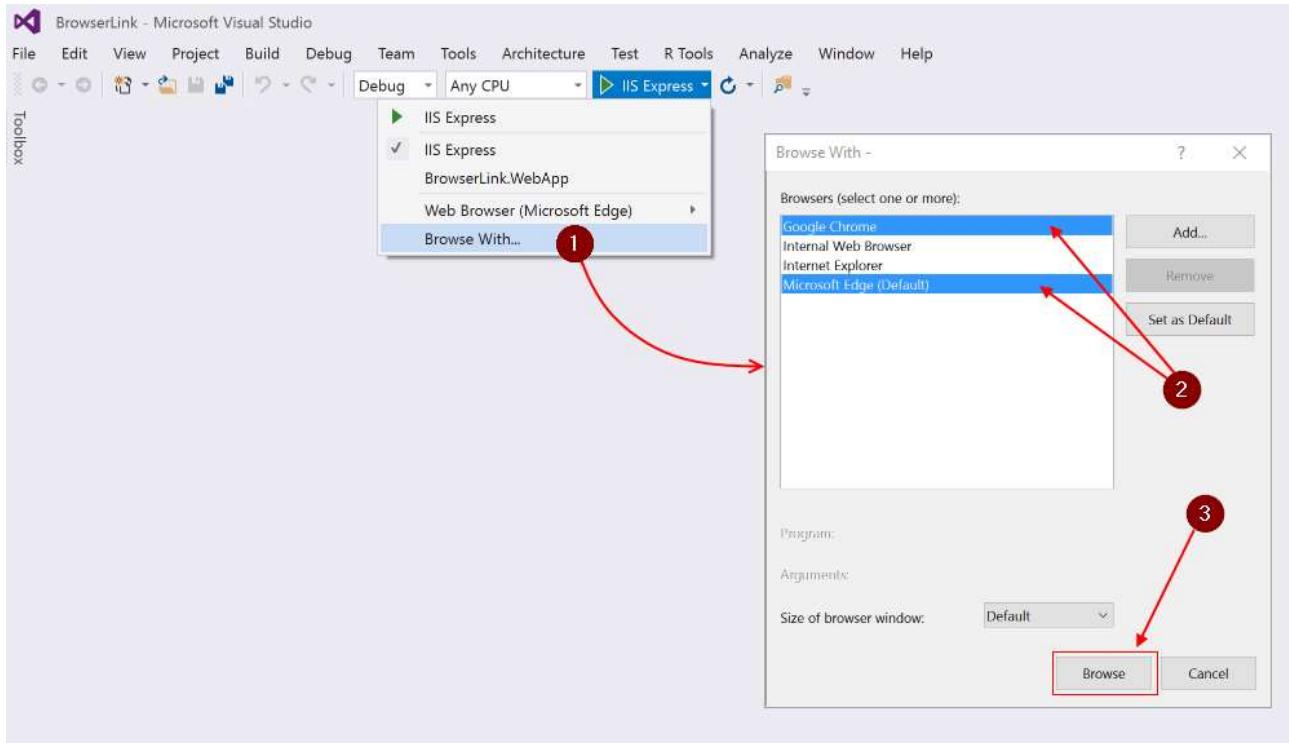
- Refresh the web app in several browsers at once.
- Open the **Browser Link Dashboard**.
- Enable or disable **Browser Link**.
- Enable or disable **CSS Hot Reload**.

Refresh the web app in several browsers at once

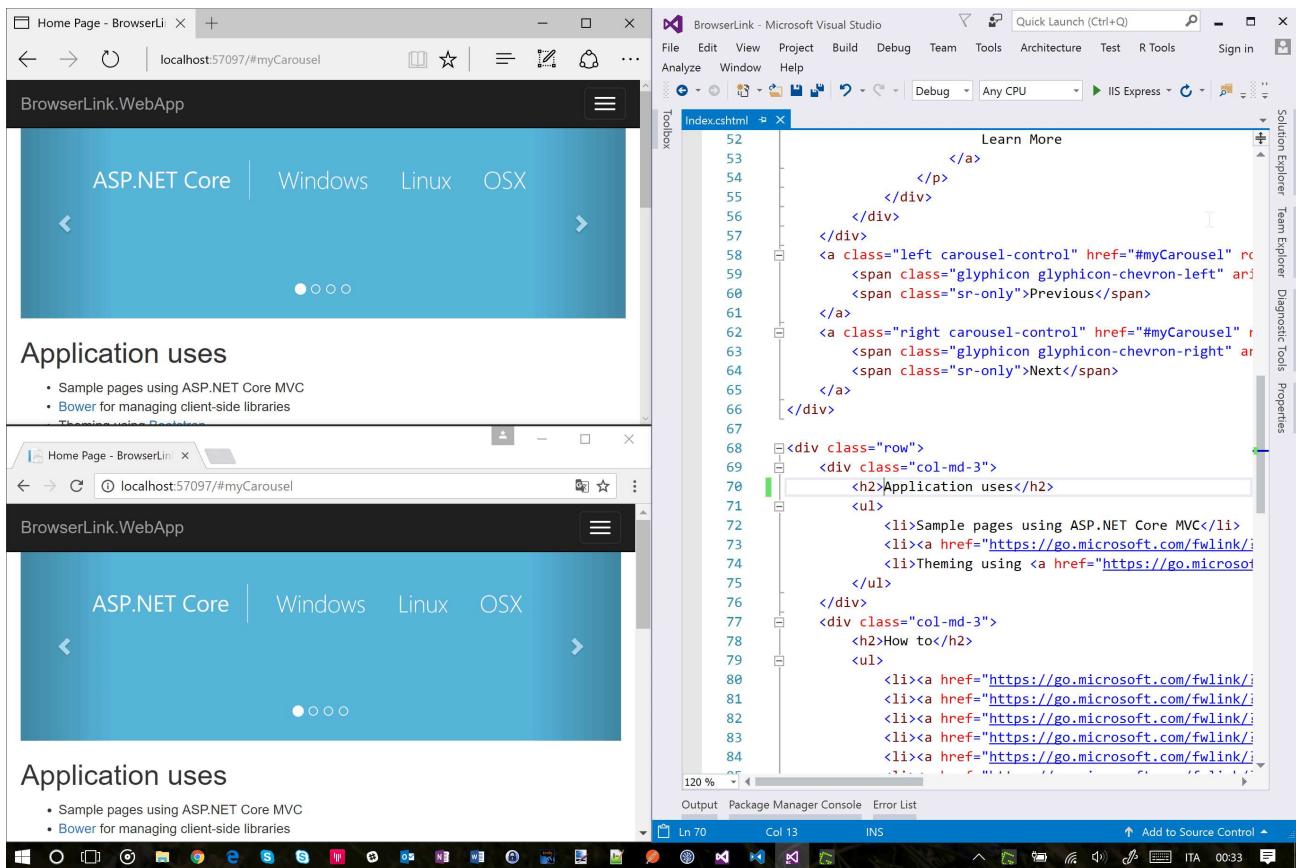
To choose a single web browser to launch when starting the project, use the drop-down menu in the **Debug Target** toolbar control:



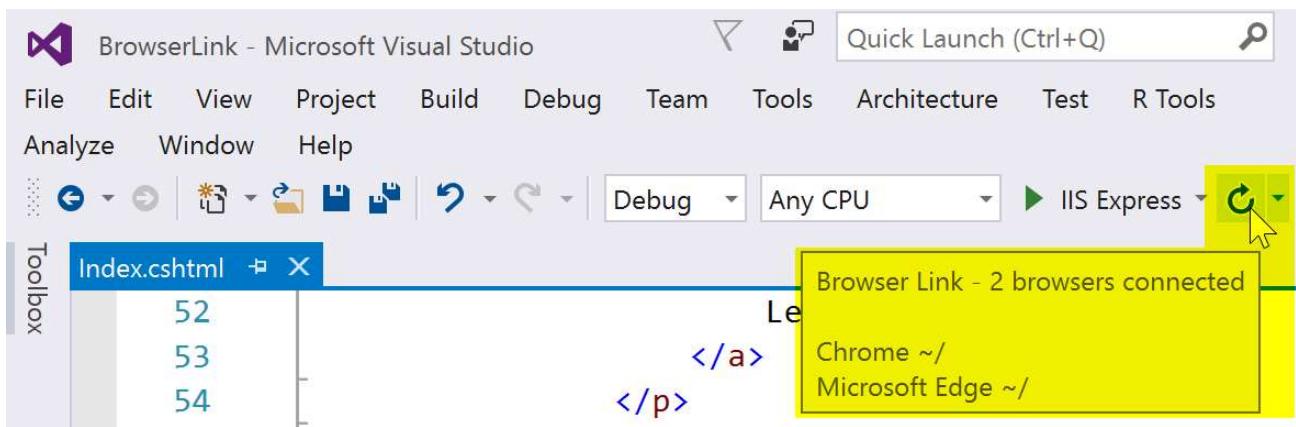
To open multiple browsers at once, choose **Browse with...** from the same drop-down. Hold down the **ctrl** key to select the browsers you want, and then click **Browse**:



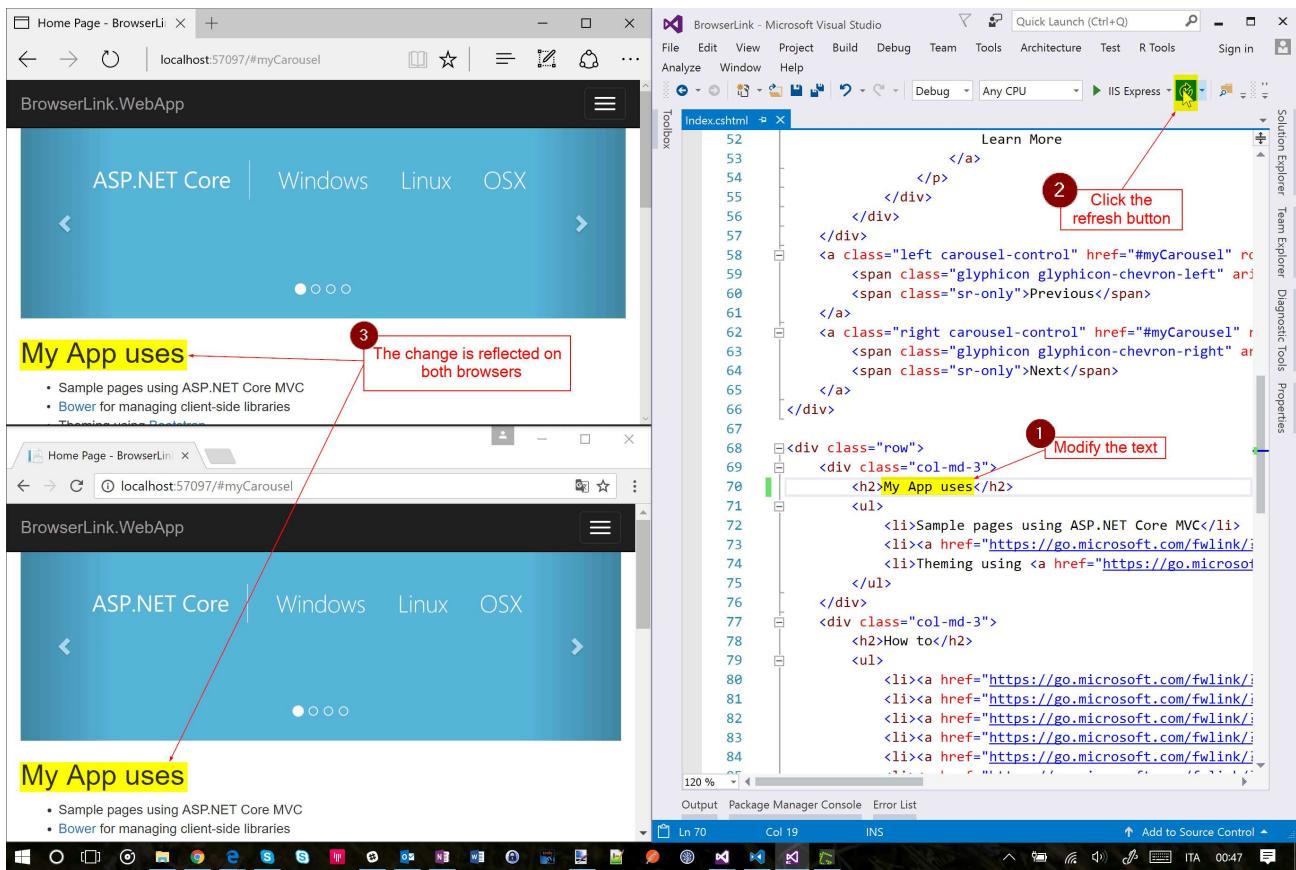
The following screenshot shows Visual Studio with the Index view open and two open browsers:



Hover over the browser link toolbar control to see the browsers that are connected to the project:



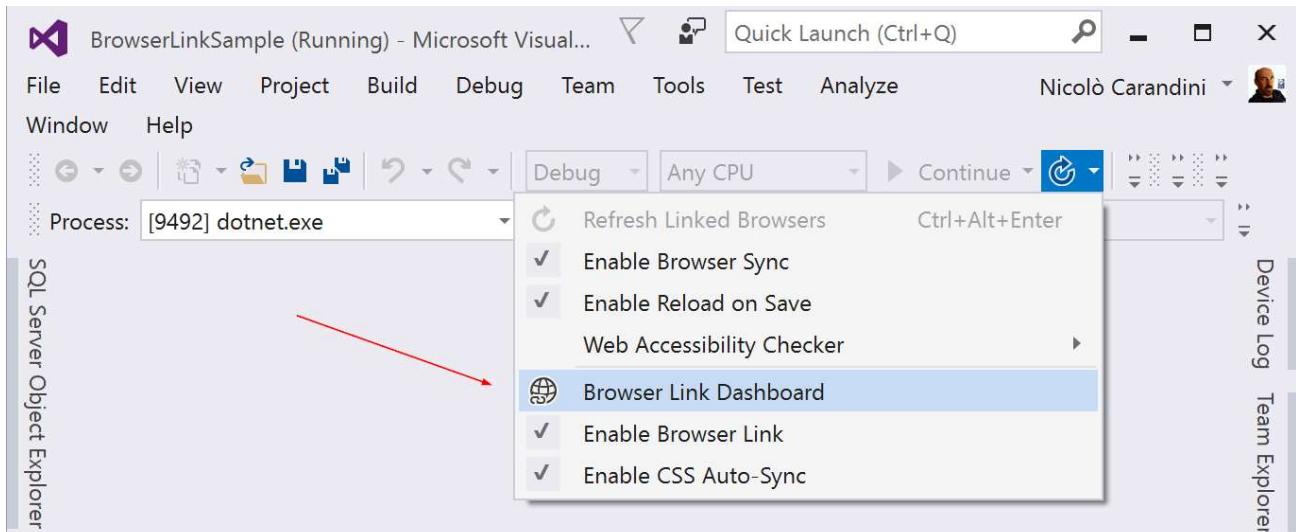
Change the Index view, and all connected browsers are updated when you click the browser link refresh button:



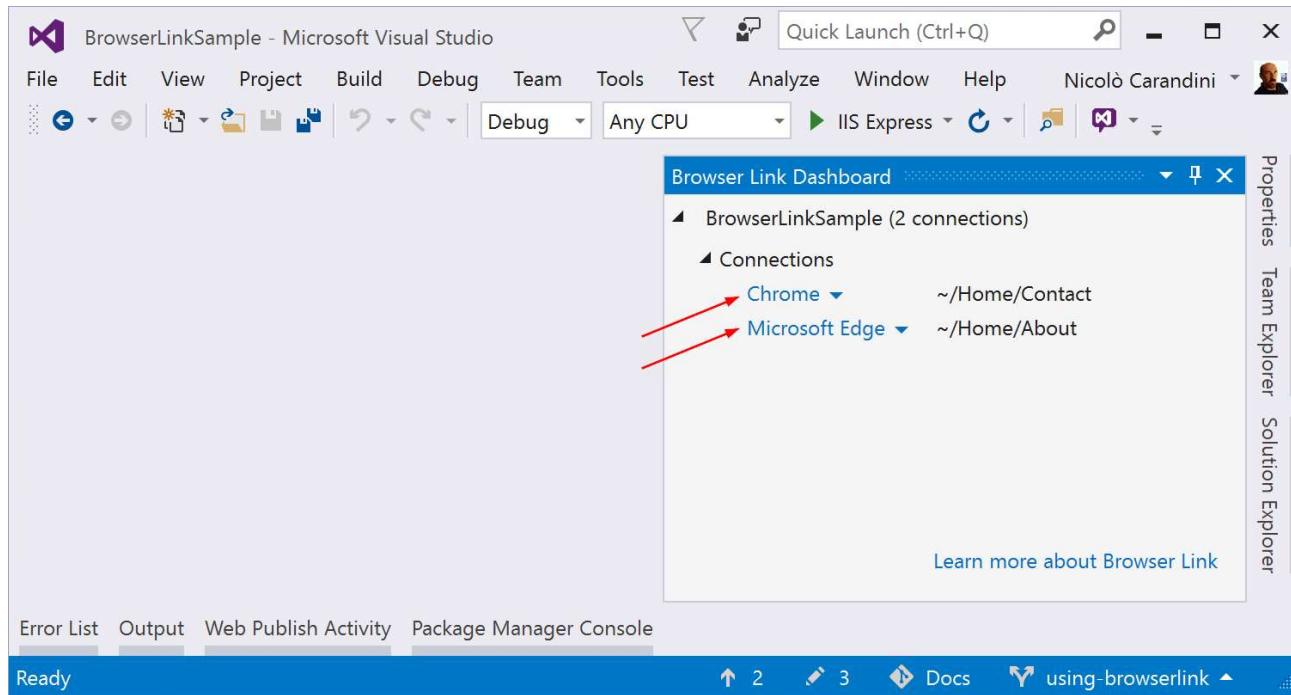
browser link also works with browsers that you launch from outside Visual Studio and navigate to the app URL.

The browser link dashboard

Open the **browser link dashboard** window from the browser link drop down menu to manage the connection with open browsers:



The connected browsers are shown with the path to the page that each browser is showing:



You can also click on an individual browser name to refresh only that browser.

Enable or disable browser link

When you re-enable browser link after disabling it, you must refresh the browsers to reconnect them.

Enable or disable CSS hot reload

When CSS hot reload is enabled, connected browsers are automatically refreshed when you make any change to CSS files.

How it works

browser link uses [SignalR](#) to create a communication channel between Visual Studio and the browser. When browser link is enabled, Visual Studio acts as a SignalR server that multiple clients (browsers) can connect to. browser link also registers a middleware component in the ASP.NET Core request pipeline. This component injects special `<script>` references into every page request from the server. You can see the script references by selecting **View source** in the browser and scrolling to the end of the `<body>` tag content:

HTML

```
<!-- Visual Studio browser link -->
<script type="application/json" id="__browserLink_initializationData">

{"requestId":"a717d5a07c1741949a7cefd6fa2bad08","requestMappingFromServer":false}
</script>
<script type="text/javascript"
src="http://localhost:54139/b6e36e429d034f578ebccd6a79bf19bf/browserLink"
async="async"></script>
<!-- End browser link -->
</body>
```

Your source files aren't modified. The middleware component injects the script references dynamically.

Because the browser-side code is all JavaScript, it works on all browsers that SignalR supports without requiring a browser plug-in.

Collaborate with us on GitHub

The source for this content can be found on GitHub, where you can also create and review issues and pull requests. For more information, see [our contributor guide](#).

ASP.NET Core feedback

ASP.NET Core is an open source project. Select a link to provide feedback:

 [Open a documentation issue](#)

 [Provide product feedback](#)