

Project Goals

- **Spend a few hours:** Put your best foot forward, submit a project that shows your best work. The evaluation will be code reviews and discussions of your implementation and architecture choices.
- **Download the empty project:** [Here](#). We have started you with an index.html file and an empty controller. When you are done you can zip it up and share it with the team for review either as a download link or zip attachment to an email. If you share the project by email keep in mind the zip has exe files inside it and Gmail and other virus scanners may block the email. To avoid issues, rename the extension to something like MyProject.CHANGEME
- **End-to-End Flow that Works:** Implement both server side and client side logic in a primitive end to end flow that runs without errors for a few small features. The key goal is to show a working end to end proof of concept with enough C# and JavaScript code that we can get a feel for your coding style and have some interesting classes to discuss in the code review.
 - **Simple is better:** The less boilerplate junk the better - if you can pick a barebones project type or delete all the default controllers and files you are not using it will help us focus on your code
 - **Focus on Code:** Spend most of your time actually writing code so that we can have a substantial work product to review. For the purpose of this exercise we really need to see original code and are not interested in seeing a lot of framework or template usage.

File & Directory Browsing Single Page App

The task is to create a web service API that allows users to query the contents of a directory on the web server and a single page web app that can be used to search and browse folders and files. A server side home directory should be configurable via variable.

Client side, the application should be deep-linkable (the state of the UI should be kept in the URL). All of the UI work should be done client side via JavaScript that renders HTML - do not render HTML server side. Styling is not as important for this task, we are more concerned with functionality. Build the UI using vanilla JavaScript or TypeScript (without React, Angular, or other UI library)

The goal is to produce a functional proof-of-concept to let us see how you think. It is important to provide a decent body of code, so that we can talk about your logic and choices. It isn't important to make the UI "pretty".

The point of this exercise is to end up with a body of work that is big enough that our team can review it and evaluate your design and coding style and skills.

Requirements

- Ensure your solution builds in Visual Studio (any version 2022 or newer is acceptable), Rider, VS Code, or command line SDK tools
- Web API that allows Browse and Search Files & Folders and returns JSON
- Deep linkable URL pattern

- SPA (Single Page App using JavaScript)
- Upload/download files from the browser
- Show file and folder counts and sizes for the current view
- Build the UI using vanilla JavaScript or TypeScript (without React, Angular, or other UI library)

Bonus

- Any cool stuff you want to show :)
- Entire component contained in a dialog widget, with a trigger element (button, etc)
- Delete, move, copy files and folders
- Performance - performance is highly valued