

As a full-stack developer at chambers and partners we regularly design and build dotnet core web API's which are consumed by varying clients.

The purpose of this technical test is to see how you approach a real-world development story. Please do not exceed 1 hour's development, just get as far as you can, finishing the API is not as important as your approach.

## The test

A new requirement has come in for a small document management solution.

As a publisher

I would like to upload, manually re-order, download and delete pdf's

So, I can place a list of documents on my client apps and website for users to download

And in an arbitrary order of my choosing

**NOTE:** For the purpose of this test, we just require the web API building, **not** the website or any client applications.

There is no requirement to secure the API.

## Acceptance criteria

Given I have a PDF to upload

When I send the PDF to the API

Then it is uploaded successfully

Given I have a non-pdf to upload

When I send the non-pdf to the API

Then the API does not accept the file and returns the appropriate messaging and status

Given I have a max pdf size of 5MB

When I send the pdf to the API

Then the API does not accept the file and returns the appropriate messaging and status

Given I call the new document service API

When I call the API to get a list of documents

Then a list of PDFs' is returned with the following properties: name, location, file-size

Given I have a list of PDFs'

When I choose to re-order the list of PDFs'

Then the list of PDFs' is returned in the new order for subsequent calls to the API

Given I have chosen a PDF from the list API

When I request the location for one of the PDF's

The PDF is downloaded

Given I have selected a PDF from the list API that I no longer require

When I request to delete the PDF

Then the PDF is deleted and will no longer return from the list API and can no longer be downloaded from its location directly

Given I attempt to delete a file that does not exist  
When I request to delete the non-existing pdf  
Then the API returns an appropriate response

## How we verify your submission

Please provide a GIT repository where we can clone your solution.

It is important the API is a working solution that we can run from our local development machines, we will try out your solution with real files.

Ideally build this in .net core as this represents most of our new work.

Ideally use any of the following data-stores:

- Cosmos (emulator)
- Azure Storage (blobs/tables/etc.) (emulator)
- SQL

If you choose a different store, please provide instructions on installation (it must be a safe installation of a well-known technology and with a free to use licence for this purpose).

We will be checking for code quality so code this as you would for a production system.

Prioritise your hour to ensure you show us the features/code you really want us to see.