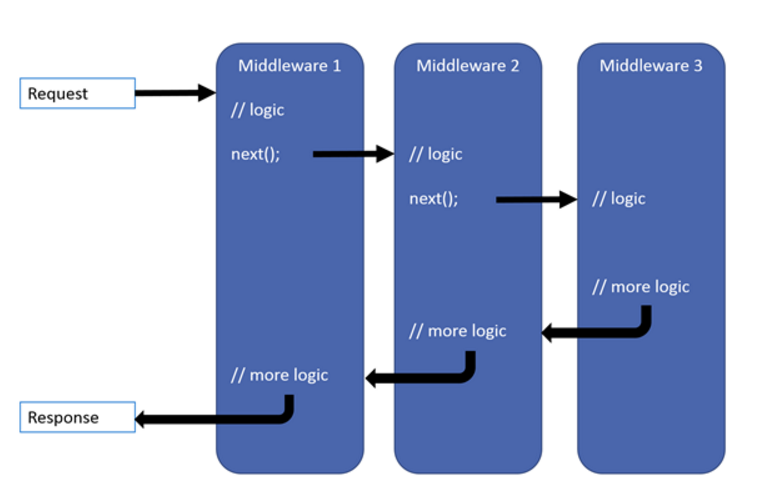
**Middleware**

* It’s a Software assembled into an application pipeline to handle requests and responses.
* Request delegates are used to build the request pipeline.
* Each component chooses whether to pass the request to next component in the pipeline and can perform certain actions before and after the next component is invoked in the Pipeline.
* Its configured using Run ,Map or use Extension method on the IApplicationBuilder instance.

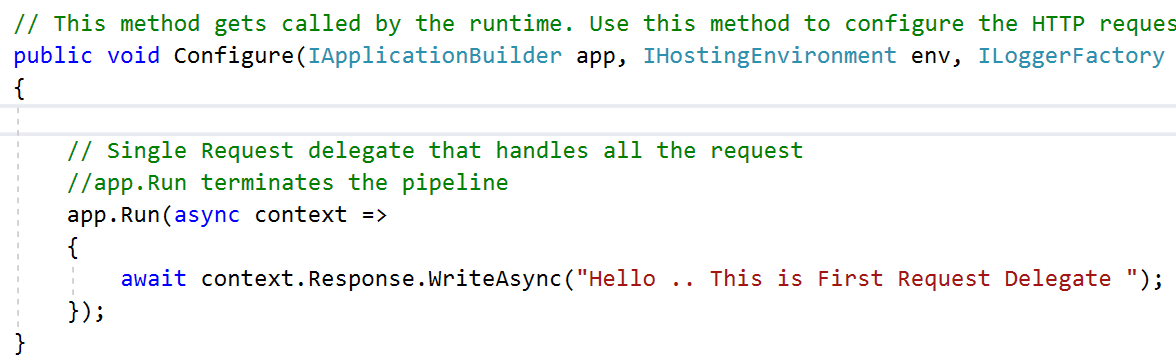


**Creating Middleware Pipeline with IApplicationBuilder**

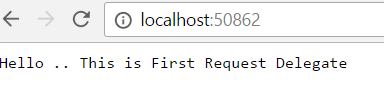
* ASP.Net Core Pipeline consists of sequence of Request delegate called one after another.
* Each delegate can perform operations , it can also decide not to pass a request to the next delegate Which is called short circuiting the request pipeline. Its needed to avoid unnecessary work.
* Exception handling delegates needs to be called early in the pipeline so they can catch exceptions that occur in later stages of the pipeline.
* App.Run delegate terminates pipeline
* You can chain multiple request delegate with app.use, the next parameter represents the next delegate in pipeline.

Demo : How to use Middleware

1. Create a ASP.Net Core Web Application.
2. Go to Configure method and add a single Req delegate

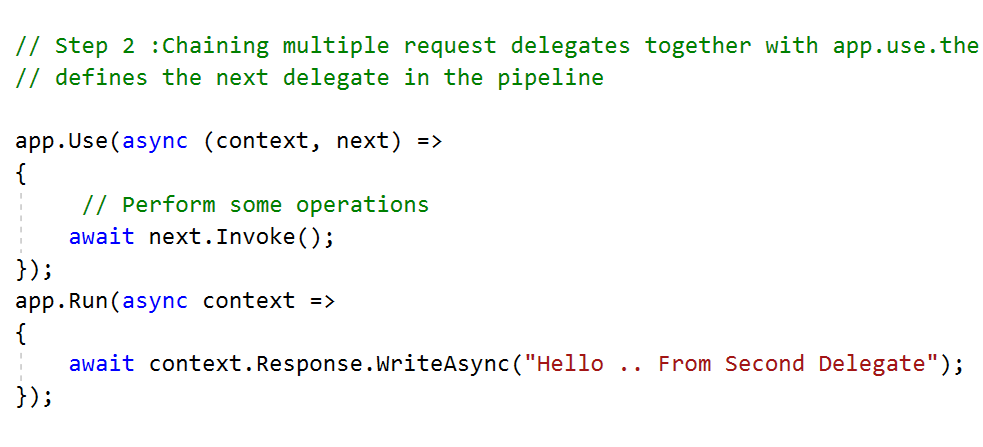


1. Output

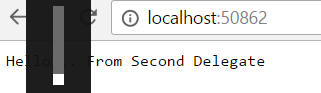


You can chain multiple request delegate as follows using app.use.

1. Add Multiple Request Delegate



1. See the output

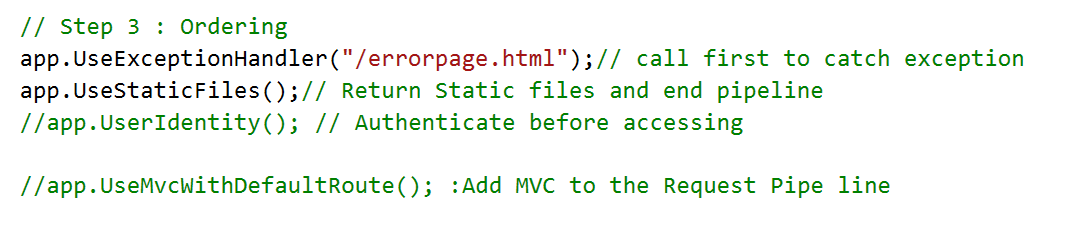


**Ordering**

* It defines the order in which they are invoked on requests.
* Ordering is critical for security, performance and functionality

You can add following components inside configure method

1. Exception and Error handling
2. Static File Server
3. Authentication
4. MVC



* UseExceptionHandler is the first component added to the pipeline hence it catches any exceptions that occur in later calls
* Static File Middleware is called early in the pipeline so it can handle request without going through remaining components.
* If the request is not handled by the static file middleware , its passed to the identity middleware app.UseIdentity which perfroms authentication .
* Authorization occurs only after MVC selects a specific controller and action.

**Built in Middleware**

1. Authentication
2. CORS
3. Response Caching
4. Response Compression
5. Routing
6. Session
7. Static Files
8. URL Rewriting Middleware

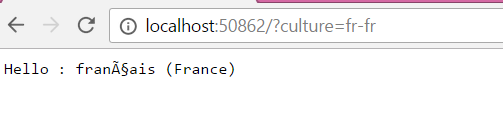
**Writing your own MiddleWare**

Sets the culture for the current request from the query string

1. Change the configure method

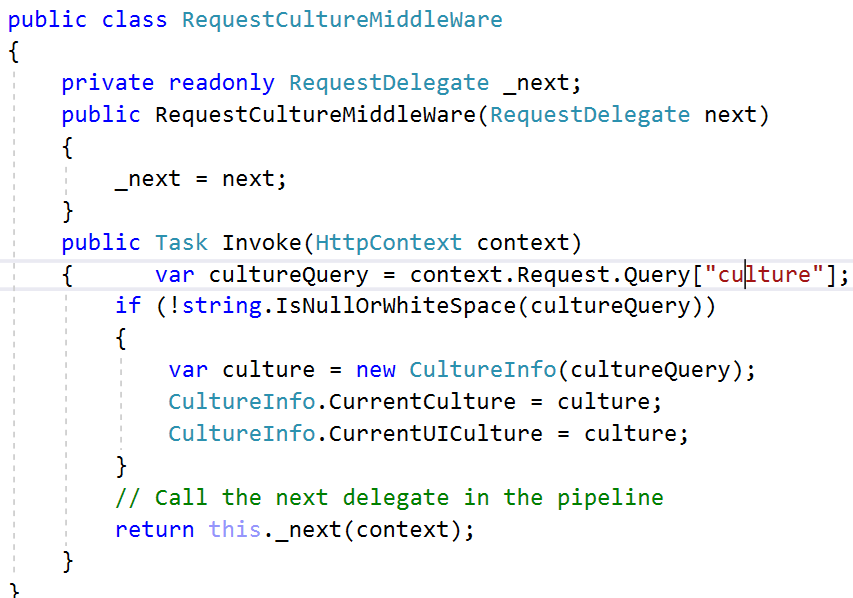


1. See the output and pass the culture from querystring

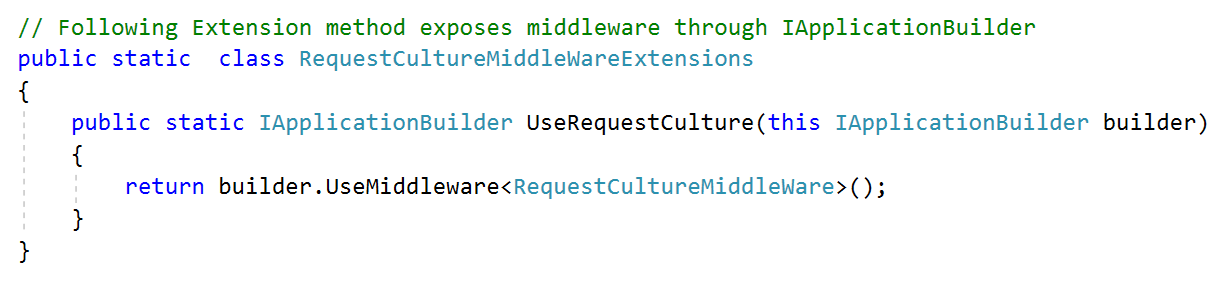


You can move it to different file

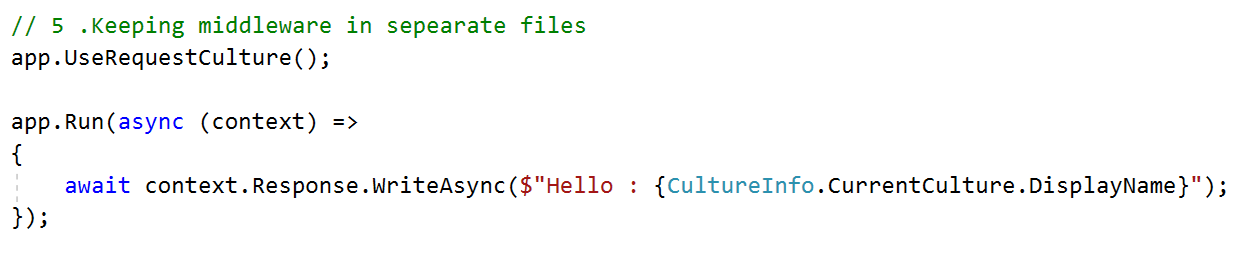
1. Add a class that contain the code of this middleware



1. Expose this middleware through IApplicationBuilder and using Extension methods



1. Go to configure method and use this middleware using use syntax.



1. Output.

