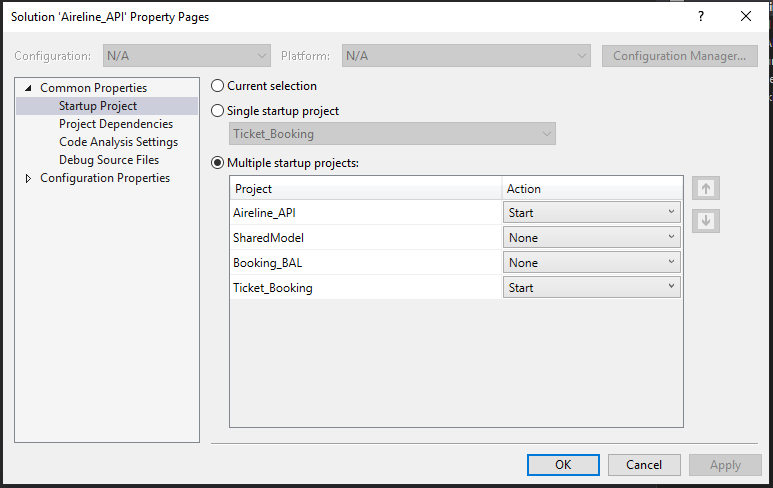
**Prerequisite:**

.Net Core 2.2

Visual Studio 2017 with update (15.9.7)

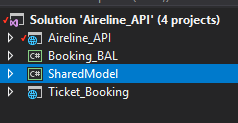
**To do’s to run Application:**

1. Ensure the **launchSetting.json** of “**Airline\_API**”, having “applicationURL” as [**http://localhost:49916**](http://localhost:49916) and the port “49916” available
2. Ensure the line “[TypeFilter(typeof(HealthCheckFilterAttribute),Arguments =new object[] { "**http://localhost:49916**" })]” under **HomeController** having the same “applicationURL” as API have
3. Ensure the solution is enabled for multiapplication start up as:



1. Once the application starts the “Airline\_API” will run silent and not showing any UI, to ensure the API is up and running we can do health check using POSTMAN, Chrome or any other API testing client application as pinging the URL “[**http://localhost:49916**](http://localhost:49916)**/healthcheck”** and this URL will return text as “Healthy” if API is running.

**Architecture of Application:**



The application is divided into one MVC (client) Application as “Ticket\_Booking”, supported by two class library projects for data binding as “SharedModel” & business layer as “Booking\_BAL” and one web API to handle booking as “Airline\_API”.

**Ticket\_Booking**: This is .NET Core MVC Application, this application holds highlights as:

1. Database.json: this is the file to load the first drop down of application from where user can select to From-To (single shot) value.
2. Filters (Airline\_Action\_filter): This filter ensure the client application is starting up only when Airline\_API is up and running else this filter redirect the user to Error page.
3. Views: This application contains one View to load initial application and one partial view to load the web API data once fetched by user.

**Shared\_Model**: This is .NET Core class library application, having three main module as:

1. AirlineModle: This model is base model to handle all data related operation from called WEP API to MVC controller.
2. BookingModle: This model contains booking dropdown model to bind drop down on View from controller and holding data for dropdown
3. BookingViewModel: this is a shared view model in between of MVC application view and WEB API data, this model have some extra properties to get hold on view and dynamic operation perform by view with the help of JQUERY.

**Booking\_BAL**: This is .NET Core library application, having business logic as:

1. One interface to hold the method exposed for business operation
2. Implementation of interface

**Airline\_API**: This is .NET Core REST API having features:

1. Having option to do health check by default behavior provided by .NET Core, using this feature the client application will check the health of this API before initiation/call
2. Having option to read the database.json file and provide the json output as REST call for airline booking process based on the route selected and provided by client application