Pre req:

Check whether the project is already configured or not, if not check git bash is installed or not.

If git bash is installed, create a new folder and start gitbash in there. Run following command

Git init

Then

Git pull Repopath

This should download code.

Update packages?

Step 1 (Time 5-7 mins):

1. Install package “Swashbuckle” to web api project and run in debug mode and navigate to /swagger
2. Go in properties\build and enable XML documentation file checkbox
3. change following line in swagger.config file

c.IncludeXmlComments([$@"{System.AppDomain.CurrentDomain.BaseDirectory}\bin\QuestionServiceWebApi.xml](mailto:$@\"{System.AppDomain.CurrentDomain.BaseDirectory}\\bin\\QuestionServiceWebApi.xml)");

1. Add xml comments to the model and API methods.
2. Add remarks tag in api methods to see implementation notes.
3. Add following in RouteConfig to point to swagger by default routes.MapHttpRoute(

name: "swagger\_root",

routeTemplate: "",

defaults: null,

constraints: null,

handler: new RedirectHandler(message => message.RequestUri.ToString(), "swagger"));

Step 2 (Exception handling in API):

1. There is no exception handling so lets add some. Create a class GlobalExceptionHandler and inherit it from ExceptionHandler. Ovverride its Handle(ExceptionHandlerContext context) method.

public override void Handle(ExceptionHandlerContext context)

{ var error = "An unhandled exception occured, check log.";

context.Result = new ServerError(error, context.Request); }

1. Add following line in web.config to config.Services.Replace(typeof(IExceptionHandler), new GlobalExceptionHandler());

Step 3 (Api Key handling in API):

1. There is no security so lets add some. Create a class ApiKeyHandlerand inherit it from DelegatingHandler. Ovverride its SendAsync method.

protected override async Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, CancellationToken cancellationToken)

{

if (!request.RequestUri.AbsolutePath.Contains("swagger") && request.RequestUri.PathAndQuery != "/")

{

var apiKey = request.Headers.GetValues(Constants.HeaderKey).FirstOrDefault();

if (string.IsNullOrEmpty(apiKey))

{

return await TokenInvalidResponseAsync(cancellationToken);

}

}

var response = await base.SendAsync(request,

cancellationToken);

// Return the response back up the chain

return response;

}

And following method

private static async Task<HttpResponseMessage> TokenInvalidResponseAsync(CancellationToken cancellationToken)

{

return await Task.Factory.StartNew(

() => new HttpResponseMessage(HttpStatusCode.BadRequest)

{

Content = new ObjectContent(typeof (object),

"Invalid Token",

new JsonMediaTypeFormatter())

},

cancellationToken);

}

1. Add following line in the webapiconfig config.MessageHandlers.Add(new ApiKeyHandler());

Step 4 (Website):

1. Change Default route to point to Questionnaire controller in Web project by going in app\_start\RouteConfig
2. Add interface IApiCallingUtil and IApiSettings interface
3. Add concrete classes for both as in code
4. Add

https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api