1. On my experience of securing an API mostly I have used/implemented JWT. This allows the API to be used only by authorized users. Also .NET has a built in way for authorizing and authenticating by the used of IdentityContext. Before creating a new project you just have to just select what type of authentication you need and Visual Studio will automatically add this library to your project.
2. APIs implemented are not only used by the same resource. This can also be used by 3rd party users or outside the same resource hence the name CROSS Origin Resource Sharing.

* GET – commonly used for getting/retrieving data from the API
* POST – Used for submitting an entity or resource. Commonly used for adding/creating data. But can also be used to update a resource
* PUT – Can be used to update the current resource to the ones the user specified in the request
* Delete – Used for deleting resource based on what the user specified indicated in the URL.

1. Attribute routing is where you specify your own route to make the api call possible rather than the default call. In this way you can have the api be called without exposing the name of your API in your code

Example:

[HttpPost(“/Address/action”)]

1. Most 3rd party APIS have its own documentation on how to integrate it in your own application so you must first understand how you can use it. And since it’s a 3rd party it’s also has its own authentication and authorization process. So getting access to the API is your next step. This is usually in the way of tokens. In my experience before I have also noticed that some APIs have only a limit on how much call can you do each day. So being aware on how much you use the API is also crucial for integrating 3rd party.

TASK

Firstly if the application has logs, I’ll look on what’s the exact error indicated rather than just having a 404 error. Secondly Since the error occurred when Project A sends a different set of data. I’ll look at the resource I passed from Project A. The user must have inputted a wrong set of data from the first one. Comparing the failed requests from the successful ones can also be helpful since this just might be a data issue.

If there’s nothing wrong with the data and it still sending an error 404. I might have to look how the Project B is handling each request by debugging. In this way I can see where the error is occurring if the logs are not indicating on what the exact error is.