# Blog

Summary

Cloudthing being an employee friendly company has come up with an initiative where in employees can voice out their opinion, share ideas or any random posts through a blog website.

The website will be a SPA with the backend API on .net core running on a full framework.

However, the developer who is working on the API is on vacation ☹ and the API is half done.

Goal

The goal for you is to complete the API and make it fully functional.

1. Implement the below API’s under the POST controller
   1. GetByTag – This API Gets all Posts having the particular tag provided in the GET request. If no posts are available return NotFound response. The related logic should be written in the PostManager.
2. Implement the below methods in the PostManager
   1. Save – This method needs to save the post into the database.
      1. If the Id being passed is 0, then it should be an Insert (new post). For saving the related tags, if it is a new tag which does not exist in the database then first insert the tag into the database and link it to the post. Else link the existing tag to the newly created Post
      2. If the Id being passed is > 0, then it should be an Update to an existing Post record. Saving of the tags will be same as the above. However, you need to make sure to untag the old tags from the post if the tags are changed.
   2. Get – This method gets a particular post based on the Id. Please be careful with Tags property - map it manually if not possible through Automapper. Throw ArgumentException when postId is not valid.
   3. Remove – Implement delete functionality of a post. Related tags should be untagged and any orphan tags not associated with any posts should be deleted.
3. Brownie points – Implement Swagger 😊. Related Nuget libraries are already added for your convenience.

Technical Details

* Framework - .net core 2.2 on full framework
* Database – In memory.
* Dependency Injection through Autofac
* Automapper for object to object mapping
* Swashbukle for Swagger documentation
* Code Map as below.

