# .NET Blog Engine App

**v 2.0 Feb 2019**

## DESCRIPTION

Build a .Net web app that allows to create, edit and publish text-based blog posts, with an approval flow where two different user types may interact.

## PART I

Build a simple blog engine using ASP.NET MVC 4.5 or later (*Core* is preferable). The app should use some kind of authentication to distinguish between non-authenticated and authenticated users. There must be two user roles defined: “writer” and “editor”.

- **Writer** users should be able to create, edit and *submit* posts. When a Writer submits a post, the post should update to a “pending publish approval” status where it cannot be further updated.
- **Editor** users should be able to query for “pending” posts and approve or reject their publishing. Once an Editor approves the publishing, the post is published and visible to non-authenticated users. If the post is rejected, it will be editable again by Writer users.
- **Editor** users should be able to delete posts.

Once a post is published, the app should allow both authenticated and non-authenticated users to add comments to posts. Each post should display its author name (Writer user) and publish (*approval*) date.

The posts can be composed in plain text, so it’s not necessary to include text formatting or images support, but will be considered as extra points if implemented.

Do not pay much attention to the UI/design side of things nor to the authentication mechanism (even hard-coded usernames and passwords are valid). The focus of the test should be the business logic and the overall architecture design. You can use Entity

Framework or another ORM of your choice, or even a different persistence solution (other than a rdbms).

**Design Rules**

- The Controllers should **not** have direct dependency on the DbContext/Data access classes.
- Use a Dependency Injection/IoC container of your choice for dependency resolution.

**PART II**

Build a REST API to manage the posts as an Editor user. The API must expose at least two endpoints: to query the pending posts, and to approve or reject a pending post:

- **Query endpoint**
  The endpoint should return a list of the posts that are pending for approval. For each post, the response must include the post identifier, the author name (Writer user) and the submit date.
- **Approval endpoint(s)**
  The endpoint(s) should receive the post identifier and an indicator about the action to perform (approve or reject). Feel free to implement one or two endpoints according to your API design.

All request and responses must be in JSON format.

**DELIVERABLES**

- Complete source code and script of database in a public repository on **github, gitlab** or **bitbucket**. If not possible please provide a .zip file but points will be deducted. *(required).*

- Readme file: **Detailed** step list to get the application up an running: software prerequisites, steps to build the app, etc, as well as the total time (in hours) it took to complete the test *(required)*
- Sample credentials for both types of user *(required)*

* **Bonus points:**

- Clean and well documented code
- Unit tests
- Use of Cloud services (Azure/AWS/GCloud) to deploy the application

zemoga