## **PROJECT: BLOG API**

### **Introduction**

Do you know what you need? You need a Blog. Or maybe you don’t, or maybe you already have one, in any case, this project will be a great way to practice and see the benefits of creating an API only backend. We’re actually going to create the backend and *two* different front-ends for accessing and editing your blog posts. One of the front-end sites will be for people that want to read and comment on your posts while the other one will be just for you to write, edit and publish your posts.

Why are we setting it up like this? Because we can! If you already have a portfolio site and you want to add your blog posts to that site feel free to do that instead of creating a new site just for that. The important exercise here is setting up the API and then accessing it from the outside. There are some security benefits to setting up separate websites for blog consumption and blog editing, but really we’re just doing it like this to demonstrate the power and flexibility of separating your backend code from your frontend code.

### **Assignment**

1. Begin by designing your back end models and schemas. How you design it is up to you, but you might want to think through a few things:  
   - For a simple blog with only a single author you might not need a user model, but you might want to set up authentication so that you can protect the editing functions with a username and password. In that case, it might make sense to set up a minimal user model, even if you are the only user.  
   - Your blog should have posts and comments, so think about the fields you are going to want to include for each of those.  
   - Are you going to require users to leave a username or email with their comments?  
   - Are you going to display a date or a timestamp for posts and comments?  
   - Posts should probably have a title, but should comments?  
   - A useful feature for a blog is the ability to have posts that are in the database but not published for the public to read. How might you designate published vs unpublished posts in your DB?
2. Set up your express app, and define the models in mongoose.
3. Set up your routes and controllers! Think about RESTful organization for this one. Most of the examples in the previous lesson were centered around posts and comments so this shouldn’t be too tricky.  
   - You can test your routes however you want. Using curl in a terminal is one handy way, but it can be just as effective to use a web browser. There are some platforms that allow you to send PUT and POST requests without needing to set up and fill out HTML forms. [Postman](https://www.postman.com/downloads/) is probably the most popular.
4. Once your API is working you can focus on your front-end code. Really, how you go about this is up to you. If you are comfortable with a front-end framework then go for it! If you’re happier using plain HTML and CSS that’s fine too. All you should have to do to get your posts into a website is to fetch the correct API endpoint and then display the results. Working with fetch and APIs from a front-end perspective is covered in [this lesson](https://www.theodinproject.com/courses/javascript/lessons/working-with-apis)
5. Create a second website for authoring and editing your posts. You can set this up however you like but the following features might be useful:  
   - A list of all posts that shows whether or not they have been published.  
   - A button to publish unpublished posts, or to unpublish published ones!  
   - A ‘NEW POST’ form. If you want to get fancy, you could use a rich text editor such as [TinyMCE](https://www.tiny.cloud/docs/quick-start/).  
   - The ability to manage comments (i.e. delete or edit them).
6. How much work you want to put into the front-end code on this one is up to you. Technically this is a back-end focused course so if you don’t actually need or want a blog on your website feel free to focus mainly on getting a REST API up and running.

### **Examples**

* <https://noobieblog.netlify.app/>
* <https://blog-client-dovimaj.vercel.app/>
* <https://generic-blogsite.herokuapp.com/>
* <https://irlgabriel.github.io/odin-nodejs-foundations/#/>