Contributing to Swashbuckle

=========

Contributions to Swashbuckle are welcomed in the form of construcutive, reproducible bug reports, feature requests that align to the project's goals, or better still a PR that's accompanied with passing tests.

If you have general questions or feedback around the use of Swashbuckle, PLEASE DON'T CREATE AN ISSUE AND POST TO STACKOVERFLOW INSTEAD.

## Bug Reports ##

If you're reporting a bug, please include a clear description of the issue, the version of Swashbuckle you're using, and a simple set of repro steps.

Please remember that Swashbuckle is a free and open-source project provided to the community with zero financial gain to the author(s). Any issues deemed to have a negative or arrogant tone will be closed without response.

## Feature Requests ##

Fundamentally, Swashbuckle is a library that attempts to generate an accurate description for APIs built on WebAPI, using [Swagger 2.0](https://swagger.io/docs/specification/2-0/basic-structure/), according to the routes, controllers and models that have been implemented. So, the resulting API documentation is driven by "actual" behavior as opposed to "intended" behavior. This is an important distinction to consider when submitting feature requests. For example, a feature that leverages built-in attributes that affect runtime behavior (e.g. AuthorizeAttribute, RequiredAttribute etc.) would be more aligned to the project goals than one that introduces custom attributes that drive documentation and nothing else.

It's also worth noting that Swashbuckle leverages the [swagger-ui project](https://github.com/swagger-api/swagger-ui) but doesn't actually implement any UI code. If you have a bug report or feature request that's UI-specific, PLEASE DON'T CREATE AN ISSUE HERE AND POST TO THE SWAGGER-UI REPO INSTEAD.

## Pull Requests ##

If you've identified a feature/bug fix that aligns to the project goals, or even just an addition to the docs, please submit a Pull Request (PR). If applicable, include tests and ensure all tests are passing locally before you commit.

Once you clone the repo, development should be straightforward bar one gotcha: the swagger-ui assets are pulled down as a git submodule and incorporated into the library as "embedded resources" during build-time. So, before building, you'll need to inititiate the submodule:

```

git submodule update --init

```