

## Team EN 08 – API Design

The theme of our presentation is API Design. We have based it on an episode of Software Engineering Radio with Wesley Beary making a guest appearance. They tackle both CLI and web APIs in their conversation.

Wesley has a long line of experience in ruby development. He has a lot of experience with cloud infrastructure startups and in this episode, he provides insight into his early work with Ruby fog and how they were finding ways to facilitate interoperability between different platforms in the early days of cloud computing.

One of the overarching themes of his podcast appearance is emphasising an agile style of development. He first likes to create a lightweight prototype of an API and then start modifying the details (or polishing the edges, as he put it) on the basis of feedback received from testing in an iterative manner.

Trying to create a perfect API design in your head before coding tends to be a waste of time as abstract planning usually cannot consider all the real-world usage patterns and extreme edge cases. Programmers at times have a hard time accounting for the point of view of their product's potential users who may not be as technologically savvy as the programmers themselves.

Wesley tells developers to be wary of using the CRUD model to do too much stuff. He talks about risks of the update method getting overloaded with overly complex parameters that can create undesired chain reactions behind the scenes. That is why Wesley recommends adding other capabilities on top of the necessary CRUD methods. In general he likes to make remove unnecessary complications from the process and he cites an example of this regarding his work with Anchor's CLI encryption certificate tool.

They talk about the principle of least surprise regarding API use. The main idea behind it is that a user should be able to instinctively figure out by using one part of the API the logic behind other sections. They use Git and OpenSSL as examples of inconsistent call names that leave the user confused as to the logic behind its use. The speakers advocate for APIs to communicate clearly to users as to why they have to use two different keywords to execute seemingly the same action in different objects. Examples of ambiguous error messages which do not help the user to solve their problem are also brought up to highlight the need for clear communication.

It is a challenge to comply with these principles when doing something innovative as users will have a hard time grasping something new instinctively. Wesley advocates

creating new patterns entirely that will use the same consistent logic throughout the platform, instead of just coming up with single new features that don't mesh with the rest of the interface.

Wesley is fond of OpenAPI and advocates for its use when designing APIs. He walks the listeners through the process of first creating a specification with it that is easily digestible by both humans and machines. The spec allows to have a living document that can be automatically updated when making changes to your API. You are able to avoid documentation depth when automating documentation according to your openAPI spec which can become a problem with versioning and updating your API.

There are tools that you can use to automatically create a mock up server based on your OpenAPI specification and the API endpoints that you have defined in it. You can use these user clients to implement CLI testing which will automatically raise errors if your API is not working as specified with OpenAPI. This facilitates debugging and error catching before you start to implement the backend servers for your software, which is when you do not want to be willing to go back and change fundamental design flaws in your structure.

Wesley is currently working for the start-up called Anchor. He describes the purpose of his current startup as providing companies with easier access to encryption certificates. Anchor has created an encryption certificate to help companies access encryption for their platforms in a seamless manner. The duo has an eye-opening discussion emphasizing security with instruction tutorials. Usually how-to tutorials regarding technology focus on providing the fastest possible solution to a given problem but tend to overlook security aspects of the process.