Janis Gonzalez

WEB 420 RESTful APIS

Discussion 5.1 Organizing Data

April 13,2023

Bellevue University

In order to organize data for an API, one should consider how it might be easier for their company or data to be organized. If the data is better organized by each piece having its own URL, then it should be organized that way. The company or individual creating the API chooses which would be the best way to structure its API for its purpose and should be guided by the pre-existing RESTful API best practices (Donovan, 2023). Knowing the APIs goals and purpose will help individuals be able to prepare for testing the data for input and output. It helps define the verification approach to the API. This could include encryption, authorization, authentication, and identification (2022). Maintaining good security practices is also important since it receives and sends private information. Individuals shouldn’t be able to access more information than they should be allowed to. For example, a regular user should not be able to access information from another user or the data of admins. Feedback from users and other developers is important for API functionality. It should be readily available for users to submit feedback in order to make sure that the API is functioning to its capacity and correctly. There should also be different versions of an API if any changes are being made so that it doesn’t break any clients (2022). This way it also doesn’t force everyone to move to the new API at the same time and instead gradually phases them out (2022). Analyzing the APIs performance is also important because an API should be fast to provide an adequate experience. As soon as the performance indicators are known, one should start improving them and improving performance issues that might arise from large files (2022). Writing API documentation helps provide the necessary documentation for information to use the API. This API documentation includes authentication scheme; endpoints definition; supported HTTP requests and responses; all interfaces, classes, constructors, and exceptions; methods, structure, and accepted parameters for each URI; and error descriptions (2022).

References:

Donovan, R. (2023, February 6). *Best practices for REST API design*. Stack Overflow Blog. Retrieved from <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>

*How to build an API: 5-step guide – mindk blog*. Web and Mobile App Development Company - MindK.com. (2022, November 7). Retrieved from <https://www.mindk.com/blog/how-to-build-an-api/>