The primary tool used to support the web service APIs consists of good documentation implemented as a web application that can easily be centrally managed and kept up to date. In order to minimize the complexity and cost of support, the development and maintenance of client SDKs per supported programming language are to be avoided. As long as the API's themselves are universally compatible, well designed, logical and consistent, and the message syntax is clear and simple, good documentation is all that developers need in order to use the APIs effectively. Developers of each programming language are then free to create their own utilities that they design, build and maintain for themselves to suit their own specific requirements.

More importantly, this establishes a clear demarcation between client app development and web service API development so that developers of the API's are not responsible for any client application code. This greatly simplifies the amount of work required to support the APIs for users, clients and customers, while also minimizing support costs for the organization.

Requirements

1. Documentation web application
What: *A web application documenting the system is needed, containing sections for requirements, design, implementation, configuration, setup, maintenance, support, etc. plus an API dictionary that provides details of all API methods, input parameters and return values. This API dictionary, combined with the generic XML fragment API request and response messages eliminates the need for any client SDK's to be built and maintained.* Why: *Building and maintaining client SDKs for each supported programming language and platform is very expensive, and it also complicates support of the system in that the SDK's run remotely in the client applications using the APIs (merging system code and client application code). Basing support on documentation of the APIs and samples to guide developers avoids the cost of SDK development and maintenance, and also greatly simplifies support for the system by keeping client application code and system code cleanly separated.* Testing: *The test files for each API method should match the API method documentation in terms of input parameters and return values, so confirming that they are the same and then running tests using the test files is one of the best ways to prove that the API dictionary information is correct.*

© 2023 Alan Rahn *All Rights Reserved*

2. One or more software tools to enable user feedback

What: *Users of the system need to be able to report problems, issues, and suggestions to the people responsible for maintaining the system. This communication should be asynchronous, like a trouble ticket, forum, or email.*

Why: *The feedback tools are mainly a way for users to report bugs and/or requests for new features, but also as a backup mechanism for the monitoring tools to allow users to report other types of issues that adversely affect the normal operation of a software system.*

Testing: *The tools to submit bug reports, requests for new features, and operational issues are API based applications and are tested in the same way as other applications and API methods.*

3. System monitoring and observability

What: *Data about the ongoing state of the functionality of a system must be constantly collected and monitored so that problems are detected soon after they occur, eliminating the need for users to report problems themselves.*

Why: *The many continuous tests verify the correct functionality of all the web service APIs, but the use of those APIs by client applications will generally still need to be tested manually. Some edge cases can occasionally slip through the cracks of the manual UI testing. Also, suggestions to improve the usability of the applications from actual users of the system can also be beneficial.*

Testing: *The manual test scripts for UI's are typically checklists of steps to be performed by the tester as they systematically verify all of the functionality in each UI. The results of those tests verify the correct end-to-end functionality of the application calling the methods of the API's.*

© 2023 Alan Rahn *All Rights Reserved*