1. [Advantage and Disadvantages of Restfull web services](#advantages_disadvantages_restfull)

1. **Advantage and Disadvantages of Restfull web services**

<http://www.actionidentity.com/blog/post.cfm/messaging-protocols-soap-vs-rest-which-one-s-better>

**REST Advantages:**

One of REST’s primary advantages is its ease **of implementation on the client side of an application**. In addition to this idea of ease of implementation, the results of a REST call are also easily human readable. **Unlike SOAP, REST does not have a lot of extra XML markup allowing any implementation of it to be comparatively lightweight.** It only requires a HTTP stack to work, so there is an argument to be made that REST is more interoperable than SOAP; despite the fact that SOAP was designed with that exact thing in mind. With REST not only is the client and server implementation usually a more lightweight solution, but they also tend to use less bandwidth than their SOAP counterparts.

**REST Disadvantages**

The most glaring problem with RESTful web services is that they have no built-in standards for security or reliability. While it is completely possible to build this into a RESTful SOA solution, it is often complicated and requires an error-prone, time consuming “roll your own” approach. The key problem that arises with the lack of a standard method to handle messaging reliability is for operations that are sensitive on the amount of times they are performed. For example, if you want to perform an operation against a payroll service that adds a bonus to an employee’s upcoming paycheck, you would not want to accidently and unknowingly send the same message twice; causing a discrepancy in the amount they should be paid and the actual amount they are paid.