# JSON vs XML

* **Less Verbose**: JSON has a more compact style than XML, and it is often more readable. The lightweight approach of JSON can make significant improvements in RESTful APIs working with complex systems.
* **Faster**: The[XML software parsing process](https://www.programmableweb.com/news/xml-vs.-json-primer/how-to/2013/11/07) can take a long time. One reason for this problem is the DOM manipulation libraries that require more memory to handle large XML files. JSON uses less data overall, so you reduce the cost and increase the parsing speed.
* **Readable**: The JSON structure is straightforward and readable. You have an easier time mapping to domain objects, no matter what programming language you're working with.
* **Structure Matches the Data**: JSON uses a map data structure rather than XML's tree. In some situations, key/value pairs can limit what you can do, but you get a predictable and easy-to-understand data model.
* **Objects Align in Code:** JSON objects and code objects match, which is beneficial when quickly creating domain objects in dynamic languages.
* **JSON Limitations:** The limitations in JSON actually end up being one of its biggest benefits. A common line of thought among developers is that XML comes out on top because it supports modeling more objects. However, JSON's limitations simplify the code, add predictability and increase readability.





RESTful APIs depend on easy, reliable and fast data exchanges. JSON fits the bill for each of these attributes, while XML is struggling to keep up. As more developers expand their API integration skills, the advantages of a simple data exchange become apparent.

# Advantages of NoSQL

The advantages of NoSQL include being able to handle:

* Large volumes of structured, semi-structured, and unstructured data
* They can handle huge amounts of data quickly then RDBMS.
* They do not generate bottlenecks (bottlenecks are caused by insufficient hardware resources).
* Agile sprints, quick iteration, and frequent code pushes
* Object-oriented programming that is easy to use and flexible
* Efficient, scale-out architecture instead of expensive, monolithic architecture

# Disadvantages of NoSQL

* Compatibility problems.- Unlike relational databases, which share certain standards, NoSQL databases have few rules in common. Each NoSQL database has its own API, the query interfaces are unique and have peculiarities. This lack of rules means that it is impossible to simply change from one provider to another, in case you are not satisfied with the service.
* They are not mature enough for some companies.- Despite their implementation in some large companies, the NoSQL databases still face an important credibility problem with many companies. Critics point to the lack of maturity of NoSQL and the possible problems of instability while citing the maturity, and great functionality and stability of the DBMS.
* Lack of experience.- NoSQL’s novelty means that there is not a large number of developers and administrators who know the technology -which makes it difficult for companies to find people with the appropriate technical knowledge. On the contrary, the SGBDR world has thousands of highly qualified people.
* The NoSQL databases being open source have a different support than the commercial companies offer to their products. The open source community, although it could respond to any problem, may possibly take longer than a commercial company with proprietary code.
* Limitations of Business Intelligence.- There are one or two questions about the Business Intelligence capabilities of NoSQL databases. Can these databases provide the kind of rigorous data mining that companies use with SGBDRs? How much is programming knowledge needed to do the ad hoc consultation and analysis? The answers are not exactly positive. NoSQL databases do not have many hooks for the general use of BI tools, while the simplest ad-hoc query and analysis involves pretty good programming knowledge. However, solutions are available. Quest Software, for example, has created Toad for databases in the cloud, which provides ad-hoc query capabilities for some NoSQL databases.
* No Stored Procedures