JavaScript has a native Sort method which can be used to sort arrays. However based on the nature of the array , the implementation varies.

1. data=["a","z","c","w","j"];
2. alert (data.sort( ));

The above sort method would sort the array without any issues as it is an array of string elements.  
  
JSON return type is an array of objects. Hence sort method cannot be used directly to sort the array. However we can use a comparer function as the argument of the ‘sort’ method to get the sorting implemented.  
  
Comparer function has the following format:

1. function compare(a, b) {
2. **if** (a **is** less than b by some ordering criterion) {
3. **return** -1;
4. }
5. **if** (a **is** greater than b by the ordering criterion) {
6. **return** 1;
7. }
8. // a must be equal to b
9. **return** 0;
10. }

Return value from the comparer function basically identifies the sort order of subsequent array elements. Summarized as the following:

* If ComparingFunction(a, b) is < 0, then a will be sorted to a lower index than b. Thus a will come first.
* If ComparingFunction (a, b) returns 0, then a and b will retain their respective positions. There won’t be any change in sort order.
* If ComparingFunction (a, b) is > then 0, then b will be sorted to a lower index than a. Thus b will come first.

The comparer function can be called to sort the JSON array as below:

1. var array = [{
2. "EmployeeName": "John",
3. "Experience": "12",
4. "Technology": "SharePoint"
5. }, {
6. "EmployeeName": "Charles",
7. "Experience": "9",
8. "Technology": "ASP.NET"
9. }, {
10. "EmployeeName": "Jo",
11. "Experience": "3",
12. "Technology": "JAVA"
13. }, {
14. "EmployeeName": "Daine",
15. "Experience": "7",
16. "Technology": "Sql Server"
17. }, {
18. "EmployeeName": "Zain",
19. "Experience": "6",
20. "Technology": "C#"
21. }];
22. //Comparer Function
23. function GetSortOrder(prop) {
24. **return** function(a, b) {
25. **if** (a[prop] > b[prop]) {
26. **return** 1;
27. } **else** **if** (a[prop] < b[prop]) {
28. **return** -1;
29. }
30. **return** 0;
31. }
32. }
34. array.sort(GetSortOrder("EmployeeName")); //Pass the attribute to be sorted on
35. document.write("Sorted Employee Names : ");
36. **for** (var item **in** array) {
37. document.write("<br>" + array[item].EmployeeName);
38. }
40. array.sort(GetSortOrder("Technology")); //Pass the attribute to be sorted on
41. document.write("<br><br> Sorted Technology Names : ");
42. **for** (var item **in** array) {
43. document.write("<br>" + array[item].Technology);
44. }

Output   
  
Sorted Employee Names:  
  
Charles  
Daine  
Jo  
John  
Zain  
  
Sorted Technology Names:  
  
ASP.NET  
C#  
JAVA  
SharePoint  
Sql Server  
  
This helper method comes in handy when AJAX calls return JSON array which has to be sorted and dynamically bound to the html DOM for end user display.   
  
You can play around with the [JSFiddle I have created here](https://jsfiddle.net/KSPriyaranjan/4Lzr0qux/) for better understanding