# 524.https://stackoverflow.com/questions/70556920/how-to-extract-some-data-from-a-json-file

**T:**How to extract some data from a json file [closed]

**Q:**details or clarity(/help/closed-questions)Want to improve this question? Add details and clarify the problem by editing this post.  
  
Closed 12 months ago.  
  
  
  
 Improve this question  
  
 (/posts/70556920/edit)I would like to extract some data from a json file and insert it into another json file like this:  
  
 { "name": "File", "artist": "Andrew", "attributes": [ { "trait\_type": "Background", "value": "Black" }, { "trait\_type": "Base", "value": "White" }, { "trait\_type": "Eye Type", "value": "Eye" }, { "trait\_type": "Ear Type", "value": "Ear" }, { "trait\_type": "Tail Type", "value": "Tail" }, { "trait\_type": "Headwear", "value": "Hat" }, { "trait\_type": "Mouth Accessories", "value": "Cigarette" }, { "trait\_type": "Eye Accessories", "value": "Mask" }, { "trait\_type": "Ear Accessories", "value": "Headphones" }, { "trait\_type": "Details", "value": "Headband" } ], },{ "name": "File2", "artist": "Andrew", "attributes": [ { "trait\_type": "Background", "value": "White" }, { "trait\_type": "Base", "value": "Black" }, { "trait\_type": "Eye Type", "value": "Eye" }, { "trait\_type": "Ear Type", "value": "Ear" }, { "trait\_type": "Tail Type", "value": "Tail" }, { "trait\_type": "Headwear", "value": "Hat" }, { "trait\_type": "Mouth Accessories", "value": "Cigarette" }, { "trait\_type": "Eye Accessories", "value": "Mask" }, { "trait\_type": "Ear Accessories", "value": "Headphones" }, { "trait\_type": "Details", "value": "Headband" } ], },  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
to:  
  
[ { "Background": "Black", "Base": "White", "Eye Type": "Eye", "Ear Type": "Ear", "Tail Type": "Tail", "Headwear": "Hat", "Mouth Accessories": "Cigarette", "Eye Accessories": "Mask", "Ear Accessories": "Headphones", "Details": "Headband", "tokenId": 0 }, { "Background": "White", "Base": "Black", "Eye Type": "Eye", "Ear Type": "Ear", "Tail Type": "Tail", "Headwear": "Hat", "Mouth Accessories": "Cigarette", "Eye Accessories": "Mask", "Ear Accessories": "Headphones", "Details": "Headband", "tokenId": 1 },]  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The first json file has a lot of data, I only put two examples in order not to create a code that is too long.In the new file, there are constants which are "Background", "Base" ... and the last value which is "tokenId" which increases progressively.  
  
Thank you very much.

3 **Answer**

**A1:**const json = '[{"name":"File","artist":"Andrew","attributes":[{"trait\_type":"Background","value":"Black"},{"trait\_type":"Base","value":"White"},{"trait\_type":"Eye Type","value":"Eye"},{"trait\_type":"Ear Type","value":"Ear"},{"trait\_type":"Tail Type","value":"Tail"},{"trait\_type":"Headwear","value":"Hat"},{"trait\_type":"Mouth Accessories","value":"Cigarette"},{"trait\_type":"Eye Accessories","value":"Mask"},{"trait\_type":"Ear Accessories","value":"Headphones"},{"trait\_type":"Details","value":"Headband"}]},{"name":"File2","artist":"Andrew","attributes":[{"trait\_type":"Background","value":"White"},{"trait\_type":"Base","value":"Black"},{"trait\_type":"Eye Type","value":"Eye"},{"trait\_type":"Ear Type","value":"Ear"},{"trait\_type":"Tail Type","value":"Tail"},{"trait\_type":"Headwear","value":"Hat"},{"trait\_type":"Mouth Accessories","value":"Cigarette"},{"trait\_type":"Eye Accessories","value":"Mask"},{"trait\_type":"Ear Accessories","value":"Headphones"},{"trait\_type":"Details","value":"Headband"}]}]';const result = JSON.parse(json).map((parent, i) => { const val = parent.attributes.reduce((prev, curr) => ({...prev, [curr.trait\_type]: curr.value}), {}); val.tokenId = i; return val;})console.log(JSON.stringify(result))

**C1:**No.. I have to take the data from the first file and create a new one as in the second example.

**C2:**@Andrew update the answer to input and output strings, not objects. The data is processed correctly. If you need code to read a file, you can find it in other questions like this, as there is no info on is it a local file or a web requested. Is there anything that I am missing in the question?

**A2:**const dat = [ {"name": "File", "artist": "Andrew", "attributes": [{"trait\_type": "Background", "value": "Black"}, {"trait\_type": "Base", "value": "White"}, {"trait\_type": "Eye Type", "value": "Eye"}, {"trait\_type": "Ear Type", "value": "Ear"}, {"trait\_type": "Tail Type", "value": "Tail"}, {"trait\_type": "Headwear", "value": "Hat"}, {"trait\_type": "Mouth Accessories", "value": "Cigarette"}, {"trait\_type": "Eye Accessories", "value": "Mask"}, {"trait\_type": "Ear Accessories", "value": "Headphones"}, {"trait\_type": "Details", "value": "Headband"}], }, {"name": "File2", "artist": "Andrew", "attributes": [{"trait\_type": "Background", "value": "White"}, {"trait\_type": "Base", "value": "Black"}, {"trait\_type": "Eye Type", "value": "Eye"}, {"trait\_type": "Ear Type", "value": "Ear"}, {"trait\_type": "Tail Type", "value": "Tail"}, {"trait\_type": "Headwear", "value": "Hat"}, {"trait\_type": "Mouth Accessories", "value": "Cigarette"}, {"trait\_type": "Eye Accessories", "value": "Mask"}, {"trait\_type": "Ear Accessories", "value": "Headphones"}, {"trait\_type": "Details", "value": "Headband"}] }] const res = []dat.forEach(d => { let obj = {} d.attributes.forEach(m => { obj[m['trait\_type']] = m['value'] }) res.push(obj)})console.log(res)

**A3:**Keep in mind that JSON stands for JavaScript Object Notation.That means that any valid JSON is the stringified (or serialized) version of the JavaScript object.That means we can simply create an in memory object directly from the JSON string like so: (limited example to keep it small)  
  
let origJson = { "name": "File", "artist": "Andrew", "attributes": [ { "trait\_type": "Background", "value": "Black" } ]}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Note, you need this first one to be an array of these so you really need to wrap your first JSON in an outer set of [ ] so you have an array of those that you can iterate over.  
  
Like this...  
  
let origJson = [{ "name": "File", "artist": "Andrew", "attributes": [ { "trait\_type": "Background", "value": "Black" } ]},{//second object here...},{// ... more objects },]  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Once you've done that you need a class that represents your 2nd object -- the one that you are mapping to. You'll need to complete the object to insure every property can be initialized -- it was too much typing for me.Also notice that you have some property names that have spaces & I've removed those spaces. Spaces won't work for property names. This NewObject class is just a representation of your second JSON example.  
  
class NewObject{ constructor(Background, Base, EyeType, EarType, TailType, Headwear, MouthAccessories, EyeAccessories, EarAccessories, Details, tokenId) { this.Background =Background; this.Base = Base; this.EyeType = ... }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Once you have that, you can now iterate through the first object creating new instances of the NewObject. Something like the following.  
  
// set up a new array of NewObjectslet allNewObjects = [];origJson.forEach((item) => { allNewObjects.push (new NewObject(item.attributes[0].value, item.attributes[1].value, ...)); console.log(item.attributes[0].value, item.attributes[1].value);});  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Because I suggest the origJson be wrapped in outer [ ] it means that the loaded up variable origJson becomes an array of your original objects.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
JavaScript arrays have a .forEach() method which will call a function for each item in the array -- passing each item in the array as an argument.  
  
In my example I've implemented an arrow function that has two statements:  
  
 ● that maps to the new object  
  
 ● which outputs to the console.  
  
Here's a fully working example:  
  
 // define (minimal) target class class NewObject { constructor(Background, Base) { this.Background =Background; this.Base = Base; } } // load up your JSON into an array of objects let origJson = [{ "name": "File", "artist": "Andrew", "attributes": [ { "trait\_type": "Background", "value": "Black" },{ "trait\_type": "Base", "value": "White" }, ] }] // init array to hold all new objects let allNewObjects = []; // iterate over each original object and map to new origJson.forEach((item) => { allNewObjects.push (new NewObject(item.attributes[0].value, item.attributes[1].value)); console.log("attr 0: " + item.attributes[0].value, "attr 1: " + item.attributes[1].value); });