CS 240: JSON Object Serialization Transcript

[00:00:00] So previously, we talked about a JSON Dom parser and how that was more convenient to use in most cases, more convenient than a streaming parser.

[00:00:11] Now, we want to talk about a Jason serialize, which is the third style of um parsing in generation you can do with, with JSON.

[00:00:22] And so the idea with serialization is that we as programmers, we deal with objects in our programs. We create objects in main memory, we call new.

Start visual description. The professor demonstrates how to serialize objects in JSON. He explains that serialization involves converting objects in main memory into an array of bytes, which can then be stored in a file, database, or transferred across a network. End visual description.

[00:00:33] We have objects, we build the objects into data structures, uh objects that point to each other essentially.

[00:00:40] And that's kind of the realm that we use or that we work in as programmers.

[00:00:45] But then we have this problem where we want to take our in memory data structures or object structures and we want to save them as JSON.

[00:00:54] And as we learned earlier, you can use a Dom parsing library to take your data structure, whatever it may be and convert it to Jason.

[00:01:04] But there's a certain amount of work you have to do because you have to build up this tree of objects that the library understands.

[00:01:11] And once you build up this tree of o of Jason objects, then you can save it to Jason. Well, that's perhaps more work than we want to do still. And so, um the notion of, of serializing and deserializing objects is even easier to use than a, than a Dom parser.

[00:01:28] So the word serialize is kind of strange. And so, I want to explain what that's all about.

Start visual description. The professor demonstrates the concept of serializing an object to an array of bytes and explains why this process is useful for saving objects outside of main memory. End visual description.

[00:01:35] So the idea was serializing an object is that I want to take my object that's in main memory. And I want to convert it to essentially an array of bytes.

[00:01:47] And that's why they call it a serialize is because I'm going to take an object graph essentially.

[00:01:52] And I'm going to convert that, that graph of objects to an array of bytes.

[00:01:58] Now, the bytes in the array are sequential and so they call it serializing the object.

[00:02:04] But that really just talks about converting an object to an array of bytes.

[00:02:09] Now, why would I want to convert an in-memory object to an array of bytes? Well, once I've converted the object to an array of bytes, then what I can do is I can take that array of bytes and I can store it in a file, I can store it in a database, I can transfer it across a network.

[00:02:27] Um So for example, if, if I have a an object data structure and I want to save it to a database, if I serialize my data structure, convert it to an array of bytes, I can store that array of bytes in my database.

[00:02:41] And then later I can read that array of bytes from my database.

[00:02:45] And then I can deserializing that byte array and recreate the original uh java object structure that uh that array of byte represents.

[00:02:57] And so that’s kind of the idea is you want to save objects in a, in a way that they can exist outside of main memory so that it can be saved or transferred.

[00:03:05] But later on, you want to be able to take those, those bytes and convert them back to the original object structure that was used to create them.

[00:03:13] So serializing is converting an object to an array of bytes and then de serializing is taking that array of bytes and converting it back to the original object structure that was used to create it.

[00:03:23] So that’s the problem we’re, we’re trying to solve.

[00:03:25] Now, this is something that’s going to come up a lot in the, the project for the class, the chess project because we’re going to need to take uh chess games for example and transfer them across the network between the server and the client.

Start visual description. The professor demonstrates how serialization will be used in the class project, specifically for transferring chess game objects across a network. He explains that the chess game object, which contains a board, pieces, and moves, will be serialized to an array of bytes. End visual description.

[00:03:38] And the way we’ll do that is we’ll take uh our chess game object which contains a board and pieces and moves and you know all the, all the things that are inside a chess game.

[00:03:48] And we’re going to serialize that chess game to an array of bytes. And then we can trans transfer it across um the network.

[00:03:55] Now, when I say an array of bytes, um in this case, we’re going to use an array of characters because we’re going to use JSON as our serialization format.

[00:04:03] So we’re going to convert or serialize our objects as JSON.

[00:04:07] So when I say array of bytes, it’s just really just an adjacent string or adjacent file.

Start visual description. The professor demonstrates the process of converting Java objects to JSON format using a serialization library. He shows how to create a list of CD objects, put them inside a catalog object, and then serialize the catalog object to a JSON string. End visual description.

[00:04:12] That’s what we’re talking about.

[00:04:13] Now, generally JSON is not the only data format you can use to serialize objects. There’s, there’s lots of uh formats in the industry that people use to, to serialize data. So, XML is one, JSON is another Java has a built in uh proprietary serialization format and so on and so forth. There’s, there’s lots of serialization libraries as well. So, although we’re using JSON, um this topic is not at all specific to Jason.

[00:04:44] OK.

[00:04:45] So the cool thing about serialization from our perspective is I can take any object I want, and I can easily convert it to a JSON string and then I can save that to a file or send it across the network. So, if we um let’s look at an example, let’s go back to our CD catalog example.

[00:05:05] So in this case, we, we have the same old CD catalog that has an array of C DS inside of it.

[00:05:12] Actually, the one we’re parsing is, is the other one. Let me open the other one.

[00:05:26] This one’s the one we’re parsing OK. So, we’ve got our CD catalog with array of C DS.

[00:05:32] And so let’s, let’s assume that I have a CD catalog in Java.

[00:05:41] And so if we look at the example code here, what we're going to do very first thing we do in this example is we create a list of C DS. These are Java objects. So, if we go to the CD factory, um if we look at that, you can see that, that uh the CD factory, all it does is it creates a list of CD objects and it just returns that, that list.

[00:06:17] And so first thing we do is we create a list of C DS and then we, we put those C DS inside the catalog object.

[00:06:24] And so now I've got my Java data structure, which is something that I as a program are very familiar with.

[00:06:32] But let's say I want to take this catalog object and save it to adjacent file.

[00:06:39] OK. So, we're going to take our catalog and we're going to save it or generate some JSON that saves it.

[00:06:50] And so down here in the generate method, we just pass in the catalog, we pass in the name of the file that we want to save it to.

[00:06:56] And then the first thing we do is we create a serialize object. Now, the library we're using here is called JSON.

[00:07:07] And so this JSON object is the, the object that knows how to serialize and de serialize java objects to and from JSON format.

[00:07:16] And so having created my serialize object named JSON, all I have to do is call the two JSON method on my serialize and pass it the Java object that I want to, to serialize or to save. So, I passed my catalog into the serialize.

Start visual description. The professor demonstrates how to use the JSON library to serialize and deserialize Java objects. He shows how to call the toJSON method to convert a catalog object to a JSON string and save it to a file. End visual description.

[00:07:34] And what it will do is it will take this catalog object and all the cds inside of it and all the objects that those C DS point to like strings and, and whatever it’s going to take this whole graph of objects and it’s going to convert it to Jason format and give me back a string that represents that, that catalog.

[00:07:53] And so with one method call, I can take that whole CD catalog and save it to Jason.

[00:07:59] Now that I’ve got my Jason string, I can do whatever I want with it.

[00:08:02] In this case, we save it to the file. So, I, I open the file and we write the string to the file and we close the file and, and that’s it.

[00:08:11] So just with essentially a couple of method calls here, I was able to take an entire java data structure and convert it to adjacent string.

[00:08:19] That’s a very powerful idea as a programmer because you’re often going to need to do that sort of thing.

[00:08:26] Now, let’s um let’s run this program.

[00:08:33] Let’s see. Let’s go look at the uh command line arguments.

[00:08:39] So I’ve, I’ve uh told, told it to save the uh the catalog to a file name, sir L dot JSON.

[00:08:49] And so if I go to my project and I look for that file, this is the output from JSON.

[00:09:00] So when I called the two JSON method, that’s what it produced.

[00:09:04] And what you can see here is that um what it essentially did is it just took the variable names in my Java code and use those variable names as the properties for the, the Jason objects.

[00:09:14] So if you look in the, the catalog class, for example, you can see that the CD list is stored in a variable name, catalog.

[00:09:21] And so for that reason, if we go over here, you can see that um this, this root object is just an object that has a CD property and that, that catalog name came from the code.

[00:09:34] And then of course, if you look in the catalog class, you see that uh it's a list of C DS and of course, the way you represent a list in JSON is with an array.

[00:09:44] And so it took and created an array and then it just iterated over all the CD objects that were in the array and it serialized each of those.

[00:09:53] And so if we go to the CD um class, you can see these are the variable names that are used in that class.

[00:09:59] And that's, that's where the property names came from.

[00:10:03] So that's what it means to serialize an object. It just um saves the object out to some data format and it's probably going to use the variable names in the code to uh to achieve that.

[00:10:17] So that um is how you serialize the file or an object. That's how you serialize an object to JSON using the JSON library.

[00:10:25] Now, let's look at the flip side where let's say later on, I want to be able to take this file and I want to read it back into my program and I want to convert this, this JSON file to a catalog ja object.

Start visual description. The professor demonstrates the deserialization process, where a JSON file is read back into the program and converted into a Java catalog object. He explains how to use the fromJSON method to achieve this. End visual description.

[00:10:38] So for this, we're going to go to the um the simple object de serialization example in the example code.

[00:10:48] And so in this case, we're doing the opposite of what we did before.

[00:10:51] In this case, we have a file name that's passed into the program and that files already got the JSON data in it. And what we want to do is get back a list of C DS.

[00:11:06] OK? So, we're going to call the parse method.

[00:11:09] And in the parse method, the first thing we do as usual is we open the file that we're parsing, wrap it in a buffered reader. So, it's more efficient.

[00:11:18] And then we go ahead and create our, our de serialize, which in this case is also the Gong class.

[00:11:25] So we're going to create an object that knows how to de serialize the data.

[00:11:29] And then once we've created the G Sun object. Um We'll call the from JSON method on it.

[00:11:34] Now, when we serialized, we called the, the two JSON method. But now that we're deserializing, we're calling the from JSON method on it.

[00:11:41] And what we do is we pass in the opened JSON file, and we also pass in the name of the Java class that we want to create from that data.

[00:11:51] So we're essentially saying JSON, please take this JSON data and from it, create a Java catalog class and it will do that, but it will not only create the Java the catalog object, but it will also create all the CD objects inside of it and put those in the catalog list and so forth.

[00:12:11] So with one method call here, I can basically take all that JSON data and convert it back into the catalog that it originally was.

[00:12:19] And once I’ve got that in this case, we extract the CD list from it and that’s what we return from the method.

[00:12:25] And so the whole idea here is that with, again, with like two lines of code, I can de serialize a Java object graph from, from a JSON file.

[00:12:38] And so object serialization and de serialization is a very powerful technique that programmers use all the time for storing objects or transferring objects outside of main memory.