IIT Madras
ONLINE DEGREE

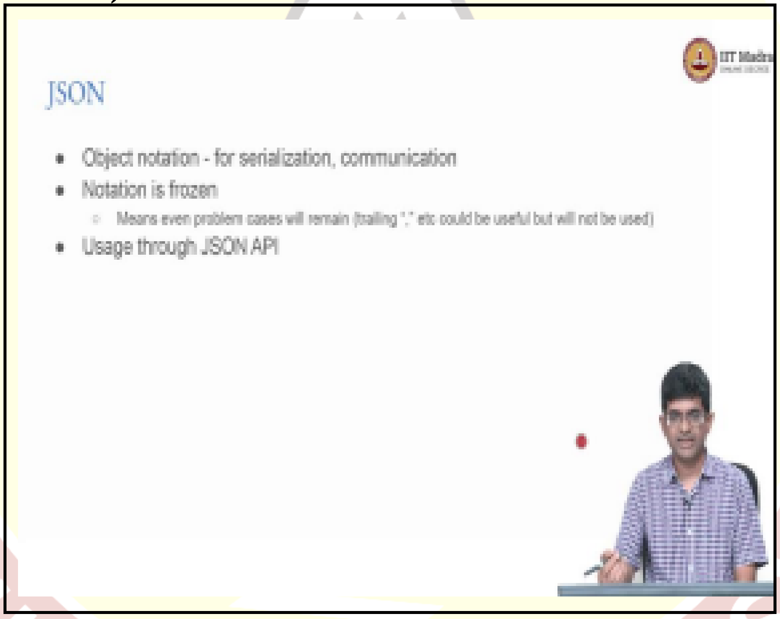
Modern Application Development II
Online Degree Programme

B. Sc in Programming and Data Science
Diploma Level
Prof. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology- Madras

Javascript - JSON API

Hello everyone welcome to modern application development part 2. Now in terms of our overall journey through the basics of Javascript we are mostly done with all the concepts that we want to look at there is one which is not really a Javascript concept as such but is something that is very useful in terms of transferring data around right. We have already sort of encountered this in the earlier modern application development part one course.

(Refer Slide Time: 00:41)



JSON

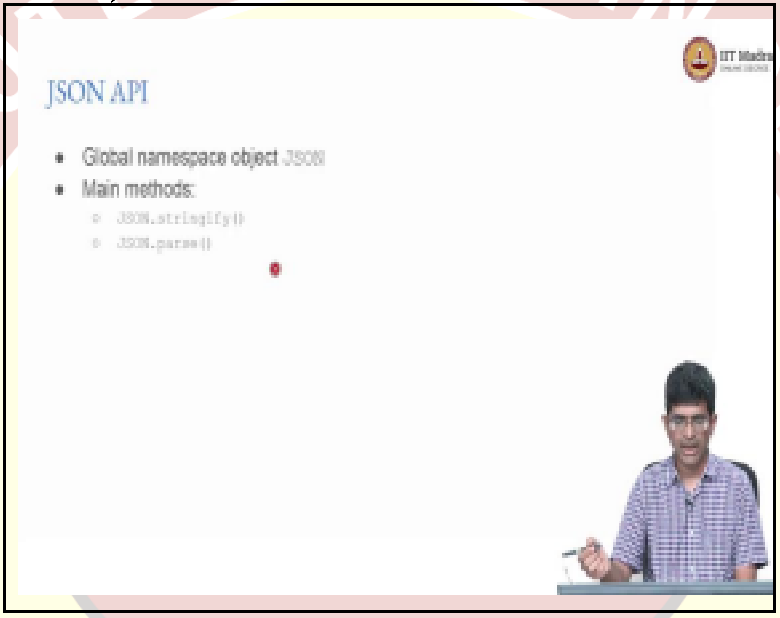
- Object notation - for serialization, communication
- Notation is frozen
    - Means even problem cases will remain (trailing "," etc could be useful but will not be used)
- Usage through JSON API

And essentially this is JSON right Javascript object notation and what do we mean by object notation? Basically means that we want to be able to sort of move objects around right we want to be able to transfer an object in some text format from one place to another. Now JSON was sort of informally put together as just some kind of a text based representation of what a Javascript object looks like internally.

It was used extensively and the interesting thing about it now is that this notation has been declared to be frozen okay and that is actually a very interesting decision that has been taken effectively. What they said is even if there are problems and there are you know for example certain complaints such as maybe one particular notation should have been done in a slightly different way in JSON.

The creators of the JSON spec basically said no we are not going to change it if it is a problem it remains a problem right at some point maybe we will have to replace JSON with something else. But for most use cases it is good enough and we will just stick with that because that way you know you can be sure that your api is never changed. In the
case of Javascript it is used through the so-called JSON api.

(Refer Slide Time: 01:56)



JSON API

- Global namespace object `JSON`
- Main methods:
  - `JSON.stringify()`
  - `JSON.parse()`

And the JSON api basically makes one namespace a global namespace object called JSON available to your scripts. So, you can straight away use JSON dot certain functions and the main methods that we are interested in are two of them one of them is JSON.stringify which basically takes an object and converts it into a string. And the other is parse which takes a string and can convert it back to an object it does not directly do.

So, but at least it parses it into something which can then be used in order to construct an object okay. So, why are we saying stringify rather than you know write to file or write to output or something like sort because very often JSON strings you want to create that string not in order to write it into a file but maybe directly to send it out over a network right. So, it may never actually make it into a actual file somewhere on disk.

So, stringify just takes care of that and says look what we are primarily interested in is making it a string that is what we are going to focus on okay.

(Video Start: 02:57)

Now we are going to take a look at how JSON can be used in order to implement both the stringify and the parse right. So, you recall the example that we used in order to implement classes right. So, we had this class cat that extends the other class called animal right. And we said that you know it has a constructor then that which in turn calls the super that is the parent constructor and it also sets the sound of the cat to be meow.

We created a new cat and we were able to log you know c dot describe and we got this. Now let us look at where JSON comes into the picture right what we are going to do here is essentially use the JSON api in order to stringify c okay. So, what does stringify c mean it basically means that we are going to take the object c which in this case we know is a cat and convert it into a string which will basically you know be put into this value p okay.

And as a result of that let us look at two things one is we want to look at console dot log what c looks like and we will also look at what console dot log p is going to look like. And if we run this we see that of course you know there are a few other console dog logs up there right but let me get rid of some of those to make things a bit clearer. So, when we run this what we will find is of course the first console log is going to be the c dot described which says tom makes a sound meow.

We already know where that came from that came from the described method corresponding to animal. Now I also log out console dot log c and what that does is that actually puts out this name cat and then it says name colon tom sound colon meow right. And I also print out what it looks like once it has been stringify in this case now notice a couple of things one is that the parameters right.

So, the object literals name and sound which were just raw literals over there have now been enclosed within quotes they have been sort of explicitly made strings the name also tom and meow have now been enclosed within double quotes all of this has basically make it made it standardized plus it has removed any unnecessary white space right. So, spaces etc have been removed from here that is not absolutely essential.

But it is you know why waste space right when you are just going to use this in order to send it to another machine or another program. So, this is good basically what it is telling us is that there is a clean way by which I could take an object and convert it into a unique string which is something that could then be passed around. But the interesting question becomes ok now how do I get back from there to the object right.

Unfortunately it is not that trivial because after all if you look at what this p contains right it does not tell you about what kind of object it is it does not tell you anything except the parameters themselves right. So, name is tom sound as meow was it a dog was it a cat was it you know just a dictionary right nothing right. You do not even know that there is such an a class as cat associated with this JSON string which means that if I want to be able to go back from the string to the object I need something in the class cat itself right.

And that is where this static from JSON function or method comes from it is a static method because it is a class level method it is not something that is there for each individual object right it is a class level method that can directly be invoked as cat dot from JSON okay. And what it does is that whenever I have a JSON string or rather a past JSON object I do not give it the JSON string I give it the JSON object in this case.

I could do the other way I could also choose to just give it a string and call the JSON dot parse inside this but I have decided that you know I will parse it first and then give the object directly to this and what it says is that you know it will create a new cat with whatever is the this object's name and set the sound to be the sound that was there in that object okay. Now if you had decided not to do this second step which was c dot sound equal to o dot sound.

Because you created it as a new cat in any case right it would have worked it would still have got meow as the correct result and you would pretty much have just ignored the fact that the meow was fed into you from the JSON string right because after all cats can only say meow right that ultimately means that there are some choice left to you in terms of how you would do it the execution of the from JSON or you know how you write that JSON interpreter.
So, let us see how that works in practice right. What I am going to do is I am creating a new variable cc and I am saying I am going to call catch dot from JSON remember what I said about from JSON being a static method and therefore a class level method and not an object method I do not have to create a new cat before I call from JSON. I can call cat dot from JSON and what should I give that as input I have to give it an object right.

A parsed object and how do I get that object i basically JSON dot parse this p this string that was created by stringifying the other object in the first place okay. So, p was created by stringifying the variable c that in turn gets parsed back into an object but just an object a dictionary that is in turn fed into the from JSON static method corresponding to the cat class. What does that method return it returns a new cat which is assigned into the variable cc.

And when I do console.log cc dot describe what will I get I should see that I have come back with tom makes the sound meow okay. So, JSON in other words has its roots in Javascript obviously right given the name but has been used in places that go well beyond Javascript it is even used as a generic sort of format for describing various notations and. So, on the reason it is been so successful is simply the fact that you know it is easy to describe and sort of very natural right even once you sort of do some indentation and so on it is very easy to read.

So, it is used well beyond what is there in Javascript but of course it plays very well with Javascript by itself. And therefore is going to be a very useful tool for us when we are trying to transfer data back and forth between different parts of a system. **(Video End: 10:23)**