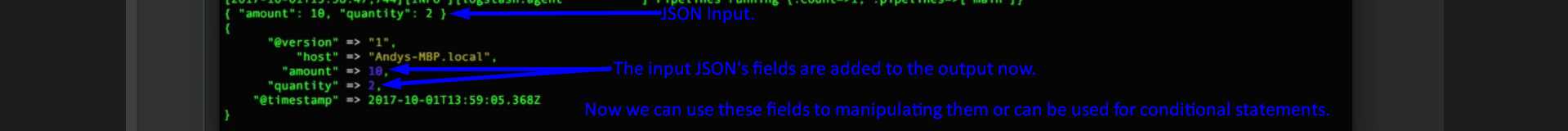
1. 
2. **Agenda**:
   1. Handling incoming JSON instead of arbitrary and unstructured text as we did in previous lecture where we give string as an input.
3. By using JSON, we could have different fields and therefore **structured data**.
4. An example could be that the JSON represents an order coming from E-Commerce App, for instance.
5. For now, we will be inputting the JSON directly from the terminal.  
   We already have a pipeline that let us input text, so let’s see what happens if we type JSON.
6. A picture containing text, screenshot, display, multimedia

   Description automatically generated
   1. Notice that the input JSON is put into the message field with the quotation marks escaped.   
      **NOTE: Jatin** 🡺 Don’t think about single quotation marks as they are not used to enclose fields and values.   
      Even if we use then that is not valid JSON😊
   2. We can’t see any new fields in the output.
   3. So, we have no way of actually doing something with the JSON as the JSON is being presented as String.
   4. That is not really useful but how to fix?
   5. **Solution**: By using a **Codec**.
7. Codecs can be used both for inputs and outputs.
8. When expecting input as JSON, we can use a **CODEC named JSON**.
9. Let’s change our pipeline configuration.
10. A screen shot of a computer

    Description automatically generated with medium confidence
11. Let’s restart the Logstash and enter the same JSON input.  
    
    1. This time around, the input JSON fields are added to the output.
    2. This is awesome as it allows us to do various things such as ( We will see later on)
       1. Manipulating them.
       2. Using them for conditional statements.
12. If you type in an array of objects as input, Logstash will generate an event for each object in the array.
13. A picture containing screenshot, multimedia software

    Description automatically generated
14. **But if we input an invalid JSON**?
15. A screen shot of a computer

    Description automatically generated with low confidence  
    Actually, the codec => json was set for stdin input plugin. So, we can input only JSON value otherwise above error.  
    We will discuss all this later on in the course.

Whether we want to send msg on SLACK, by email or drop the event, whatever be the case, we will see later in the course but basically, it involves checking if the event has the tagged with the value that we can see within the “tags” field now above in the slide.

1. Our Pipeline is capable handling only single line JSON input which means if we paste JSON spanning multiple lines, we would see the same error as you just saw.  
   That is because we must handle multi-line events a little differently.  
   That gets a little more complicated. So, we will get to this later in the course.  
   For the time, this pipeline **doesn’t support multi-line events**.
2. Let’s expand the pipeline further with some more cool stuff in the next lecture.