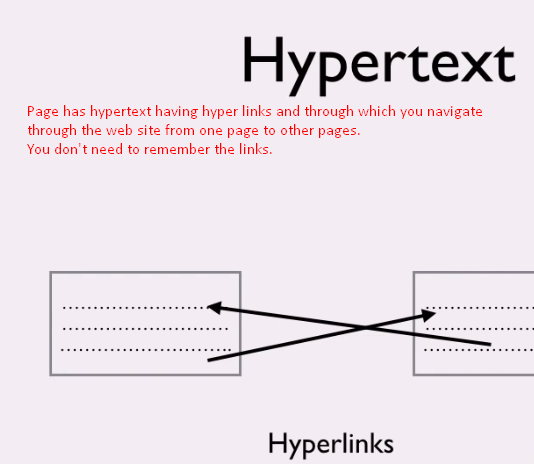
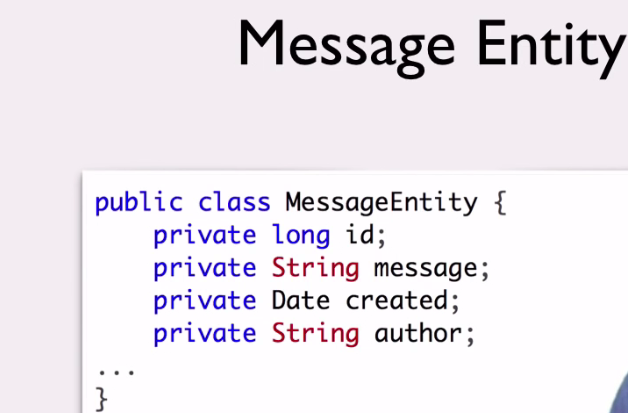
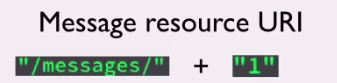
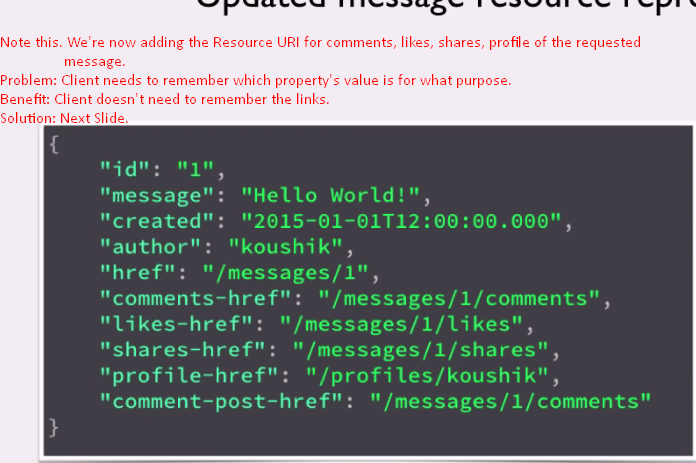
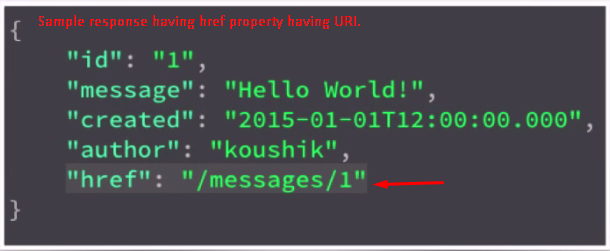
1. 
2. 
3.   
   As we talked already, there is no service definition specification for RESTful API. There is no formal document that documents the API itself. Most RESTful APIs have helpful pages that explain what the API URIs are and what the operations are supported.  
   We also mentioned that the best RESTful API doesn’t need any documentation. Let’s understand what that means by example. ☺

**Example**: We visit web site. Now tell me when we use documentation to visit a web site. Well never. You just go to the HOME PAGE where you get other links.   
Now, let’s think of the response that we returned from the REST API.   
What if I implement the same concept here. 

You say to your clients “Look, here are the other links to comments, shares, likes of the requested message”.   
So, the WS is being super helpful to the client. It’s like the Web Service with hyperlinks. Whether the client uses or not doesn’t matter.   
So, the client developer picks up a URI from the previous response and makes a subsequent call to those URI. When the response for the subsequent call comes in, that may have further URIs that client may be interested in.

1. **Advantages**: If you do this, you don’t let your clients know the URIs and hard-code this into their client app. So, by doing so, you basically let the hyperlinks in the response drive the client’s interaction with app state.  
   So, you could say that 🡺   
   The **H**ypertext/**H**ypermedia is sometimes called **a**s **t**he driver or the **e**ngine of the **a**pp **s**tate 🡸 HATEOS (Hypermedia As The Engine Of Application State).
2. Let’s start with a scenario to make the concept very clear.   
     
   See, the clients have now a list of messages.   
   Now, they want to make put request for message with message id 1.   
   They need first of all to construct the URI for update🡺 **“/messages/{messageId}**🡸 “/messages/”+”1” “/messages/”1”yellow they know and red they append.   
   They make this URI for update to make put request.   
   As a service implementer, here is the question. Why not to send the whole URL as we’re sending the message resource detail. 🡸 Solution
3. “**rel**” attribute is used in <link> to specify the link b/w the current document and the linked document. 
4. 
5. There are a couple of things to pay attention to.   
   **First**: While the concept of having URI in the response to achieve HEATEOS is well understood and agreed upon by all but the way it’s done could differ.  
   There are multiple ways to structure these URIs. This is one of them. There is no right and wrong but you have to choose what is best for your clients.  
   **Second**: “rel” attribute is the part of HTTP specification. So, there are only certain values it allows.   
   This link lists the allowed values.   
     
   Still, you can your any property name instead of “rel”.
6. **Summary**:  
   HEATEOS is a way to provide links to resources in the RESTful API response. So, the client doesn’t need to deal with URI construction and business flow. They make a request and the way to the next with URI is handed over to them with the response.  
   You can provide these links with attribute “href” and next you can provide the relationship b/w the linked resource with the current resource by the “rel” attribute.