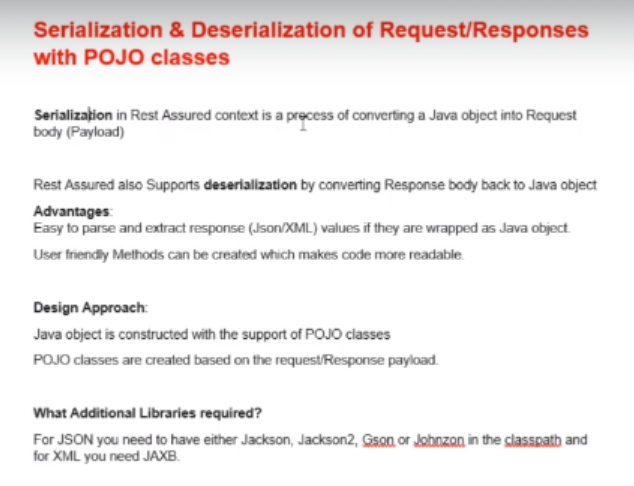
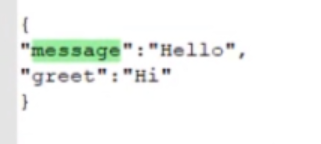
POJO : Plain object java classes

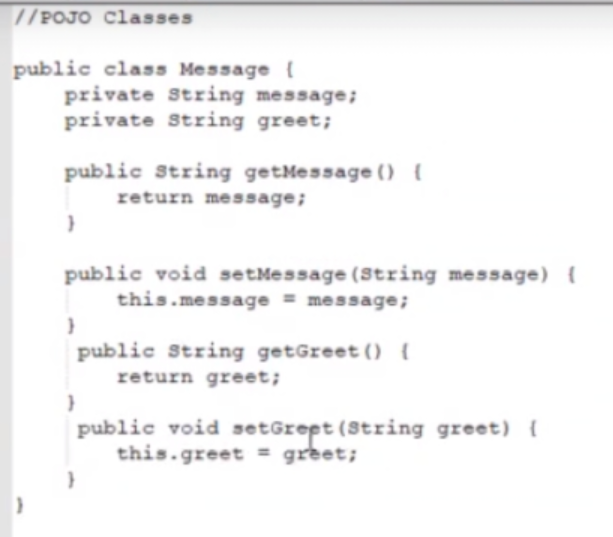


**SERIALIZATION :** Here we use setter method to set values for JSON fields

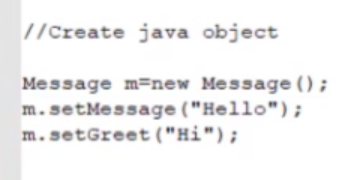
Sample JSON :



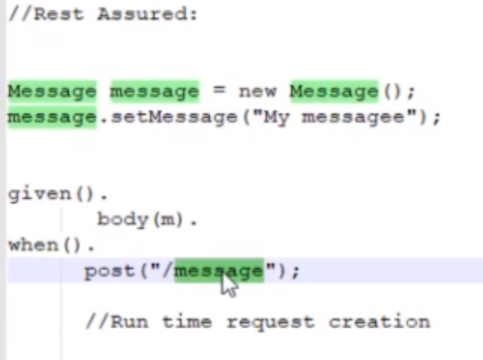
We create a Java class and in class we define each fields as Private member and we write public getter and setter method to access the fields. We created a class and methods as shown below :



Then we create object of the class and assign values to JSON fields



And this is how we send class object to RestAssured : Rest assured takes care of all rest of the thigs.



**DE-SERIALIZATION :**

Here we convert response to java object. It is exactly same as serialization implementation. Here we use getter method to retrieve values from JSON response.

1. To generate getter and setter methods :

Select private variable > ctrl+shift+s > get getter setter method. Here eclipse will generate getter and setter automatically.

1. For nested Json check that in <https://jsoneditoronline.org/>, if it shows [] means it is array and the we have to create a return type as a list for that node. All nested Jsons are not always indicated with array. But for every nested json we have to create a new class.

GetCourse gc =*given*().queryParam("access\_token",accessToken).expect().defaultParser(Parser.***JSON***)

// here we collect response in object gc of class GetCourse(POJO class)

// defaultParser(Parser.***JSON***): we tell to scan the response as JSON.

.when()

.get("https://rahulshettyacademy.com/getCourse.php").as(GetCourse.**class**);

//above line as(GetCourse.**class)** tells that we want to get response in to java object of GetCourse class

System.***out***.println(gc.getLinkedin());

System.***out***.println(gc.getInstructor());

System.***out***.println(gc.getCourses().getApi().get(1).getCourseTitle());

**Serialization** :

