

```

@SuppressWarnings("unchecked")
@RequestMapping(value="/getproductlist",
method=RequestMethod.GET,produces="application/json")
public JSONArray getProductResponse(final HttpServletResponse
response,HttpServletRequest request) throws ParseException {

    JSONArray jsonArray=new JSONArray();

    String
value=CommonProperties.getUserName()+CommonProperties.getPassword();
// Kiosk kioskObj=null;
String user=commonUtils.generateEncryptedPwd(value);

    try {
        System.out.println("usewr"+user);
        URL url = new
URL("http://data.point24h.com/mobile_webservice/getProducts.php?user=944191bfb
c1d87ee4117b800c06fd967&test=1");
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");
        if (conn.getResponseCode() != 200) {
            throw new RuntimeException("Failed : HTTP error code : "+
conn.getResponseCode());
        }
        BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));
        String output;
        while ((output = br.readLine()) != null) {
            value= output;

        }
        System.out.println(value);
        conn.disconnect();
        jsonArray = kioskExternal(value);
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    return jsonArray;
}

private JSONArray kioskExternal(String value)
{
    JSONArray jsonArray=new JSONArray();
    try
    {
        Long kiosk=0;
        int z=0;
        Long kioskid=0;
        Long pi = 0;
        ProductKiosk pk = new ProductKiosk();
        JSONObject obj=(JSONObject) new JSONParser().parse(value);
    }
}

```

```

JSONArray array=(JSONArray) obj.get("machines");

Iterator<?> k = array.iterator();

for (int m = 0; m < array.size(); m++)
{
    JSONObject json3 = (JSONObject) k.next();
    System.out.println("aRRAY SIZE"+array.size());
    System.out.println("outter loop===== "+(m));

    // Kiosk kioskObj=new Kiosk();
    //
    Kiosk kioskObj =
kioskService.getKioskName((String)json3.get("name"));

    System.out.println(kioskObj);
    if(kioskObj != null)
    {
        kioskid = kioskObj.getId();
    }

    if(kioskObj==null)
    {
        Long mId = (Long)json3.get("id");
        String mName = (String)json3.get("name");
        kioskObj = new Kiosk();
        kioskObj.setKioskId(mId);
        kioskObj.setKioskName(mName);
        kioskObj.setCreated_on(new Date());
        kioskid = kioskService.registerKioskDetails(kioskObj);
        kioskObj = kioskService.getKioskById(kioskid);
        json3.put("name", mName);
        json3.put("id", mId);
        jsonArray.add(json3);
    }
    pk.setKiosk(kioskObj);
    if(json3.get("products")!=null){
        JSONArray jarray2=(JSONArray) json3.get("products");

        // Iterator<?> i = jarray2.iterator();
        // List<?> prodList = Arrays.asList(jarray2);

        List<JSONObject> prodList=(List<JSONObject>)
jarray2.stream().collect(Collectors.toList());

        Long iii= 0;

        for(int l=0;l<prodList.size();l++)

```

```

    {
        JSONObject json=new JSONObject();
        JSONObject json2 = (JSONObject) prodList.get(l);
        if(json2 != null) {
            String id = json2.get("id") != null ?
(String)json2.get("id"):null;
            String kioskId = id;
            String name = json2.get("name") != null ?
(String)json2.get("name"):null;
            float price = json2.get("price") != null ?
Float.parseFloat(json2.get("price").toString()):0;
            long qt = json2.get("qt") !=
null ?Long.parseLong(json2.get("qt").toString()):0;
            Product product=null;
            System.out.println("TWO");
            Product
productObj=productService.getProductDetails( (String)json2.get("name"),(float)j
son2.get("qt"), (String)json2.get("id"));
            if(productObj == null)
            {
                productObj=new Product();
                productObj.setProductId(id);
                productObj.setProductName(name);
                productObj.setPrice(price);
                productObj.setQuantity(qt);
                System.out.println("ONE");
                pi =productService.addNewProduct(productObj);
                productObj = productService.getById(pi);
            }
            pk.setProduct(productObj);
            json.put("name", name);
            json.put("id", id);
            json.put("price", price);
            json.put("qt", qt);
            jsonArray.add(json);
            System.out.println("KIOSK ID
"+pk.getKiosk().getId());
            System.out.println("PRODUCT
ID"+pk.getProduct().getId());
            ProductKiosk prk =
productKioskService.getProductKioskByProductandKioskId(pk.getKiosk().getId(),
pk.getProduct().getId());
            if(prk == null)
            {
                productKioskService.registerNewProductKiosk(pk);
            }
        }
    }
}
}
}
catch(Exception e)

```

```

    {
        e.printStackTrace();
    }
    return jsonArray;
}

@SuppressWarnings("unchecked")
@RequestMapping(value="/getcategorylist",
method=RequestMethod.GET,produces="application/json")
public JSONArray getCategoryResponse( final HttpServletResponse
response,HttpServletRequest request) throws ParseException, IOException {

    JSONArray jsonArray=new JSONArray();

    String
value=CommonProperties.getUserName()+CommonProperties.getPassword();

    String user=commonUtils.generateEncryptedPwd(value);

    try {
        URL url = new
URL("http://dati.erentcinema.it/mobile_webservice/getProducts.php?user="+user+
"&cat=1");
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");
        if (conn.getResponseCode() != 200) {
            throw new RuntimeException("Failed : HTTP error code : "+
conn.getResponseCode());
        }
        BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));
        String output;
        while ((output = br.readLine()) != null) {
            value= output;
        }
        conn.disconnect();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    JSONObject obj=(JSONObject) new JSONParser().parse(value);
    JSONArray array=(JSONArray) obj.get("categories");
    Iterator<?> k = array.iterator();
    while(k.hasNext()){
        JSONObject json3 = (JSONObject) k.next();

        ProductType PTObj =
productService.getProductTypeName((String)json3.get("name"));
        if(PTObj == null) {

```

```
String cName = (String)json3.get("name");  
String pic=(String)json3.get("pic");
```

```
byte[] encodedBytes = Base64.encodeBase64(pic.getBytes());  
String s=new String(encodedBytes, Charset.forName("UTF-8"));
```

```
System.out.println(s);
```

```
System.out.println(cName);  
json3.put("name", cName);  
json3.put("pic",pic);  
PTObj=new ProductType();
```

```
PTObj.setProductTypeName(cName);  
// PTObj.setProductTypeImage(pic);  
productService.addNewProductType(PTObj);  
jsonArray.add(json3);
```

```
}}
```

```
return jsonArray;
```

```
}
```