**יצירת פרויקט web- ב IDE - - Eclipse כל הזכויות שמורות 0527670898**

1. להוריד שרת Tomcat Apache, לחלץ למיקום של כל התוכנות.
2. ניתן להרים את השרת ע"י קובץ startup.bat , אם נופל - יש להגדיר משתנה סביבה בשם JAVA\_HOME, למיקום שה JRE נמצא על המחשב ע"י פקודה ב CMD :

לדג: set JAVA\_HOME=C:\Program Files\Java\jre1.8.0\_201

(אפשר לדלג על השלב הזה ולבצע אותו מתוך הIDE )

1. פתיחת הפרויקט בIDE -eclipse

File – new – project –web –dynamic web project

(אם אין web נתקין : Help - install new software –

לבחור <http://download.eclipse.org/releases>,

אחרי שמתמלאת הרשימה לבחור : Web, XML, Java EE and OSGi Enterprise Development

ולהתקין.)

בפתיחת הפרויקט נגדיר את השרת שהורדנו שיהיה מחובר לפרויקט



1. בשביל לכתוב web service בחרנו להשתמש בספריה : [Jersey JAX-RS](https://jersey.java.net/download.html)

נוריד את הספריה  [Jersey JAX-RS 2.0 RI bundle](https://jersey.java.net/download.html) , לחלץ.

חיבור הספריה לפרויקט :

לחיצה ימנית על הפרויקט :java- build path – configure build path

Add Library – user library –

ליצור user library חדש : user libraries – new –

לתת לו שם כמו : JAX\_RS\_JERSY\_API

כשעומדים על הספריה שנוספה – Add External Jars

לבחור את כל הJARS שהורדנו מJERSY מתקיות api , ext ,lib

כעת כל הספריות נוספו לפרויקט.

1. מי שלא הגדיר בשלב הראשון משתנה סביבה JAVA\_HOME – יגדיר:

Java build path – Add variable – configure variabels

ולהוסיף את המשתנה הנ"ל

1. נגדיר שהספריה הזו תעבור ב DEPLOY :

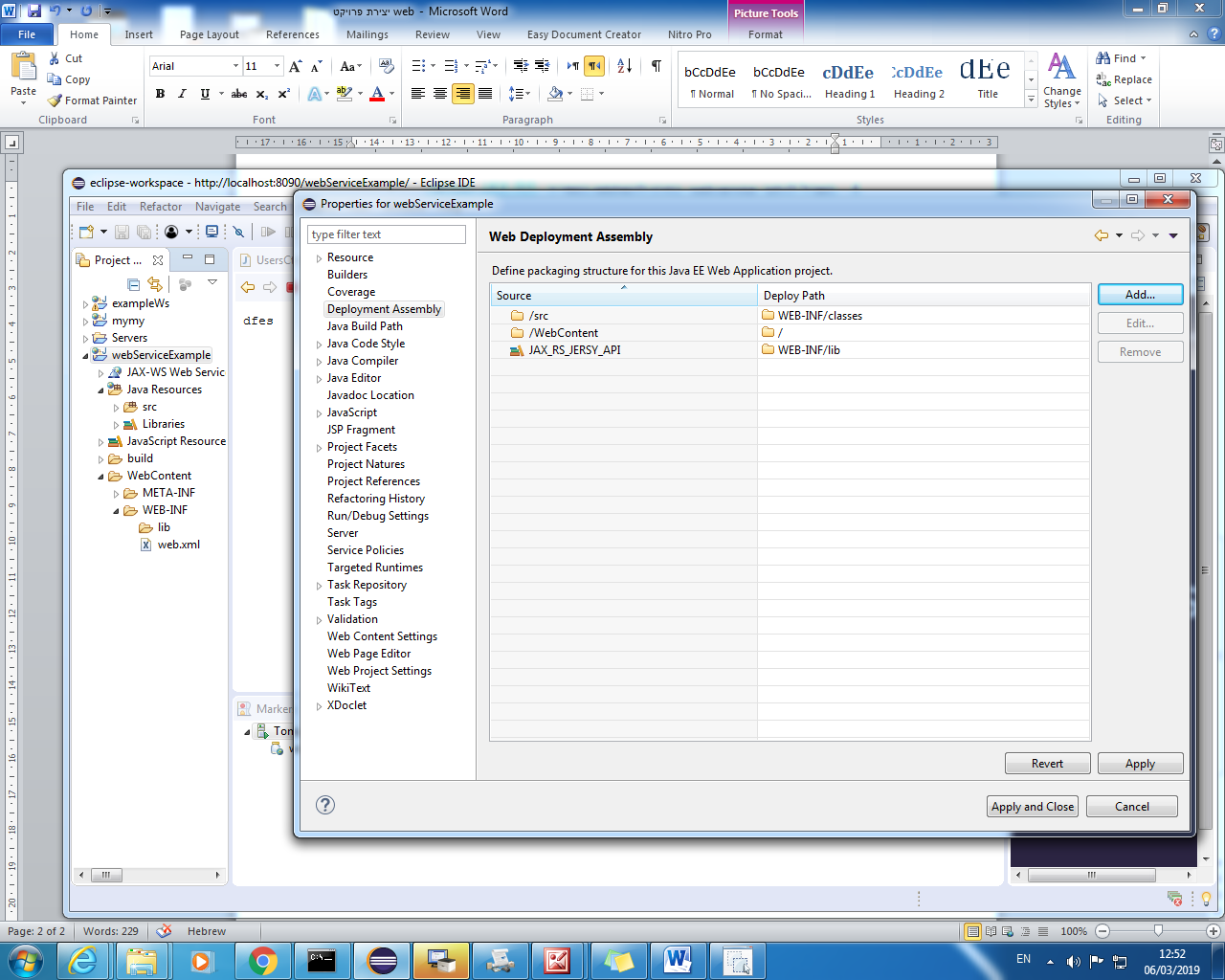
לחיצה ימנית על הפרויקט – properties – project Faces

לבחור את JAX-RS , ולמטה ללחוץ על Further configuration available…

בחלון שנפתח לבחור את ה user library שהוספנו בשלב קודם : JAX\_RS\_JERSY\_API

1. להגדיר שהפרויקט המקומפל יכיר את הספריות :

ב Deploymeny Assembly שיראה כך, להוסיף את הספריה בADD



1. כעת ניגש לכתוב את הקוד בתוך java resources – src

נשתמש באנוטציות@GET, @Produces @Path ,

דג:

1. package com.pegaxchange.java.web.rest;
2. import java.util.\*;
3. import javax.ws.rs.Consumes;
4. import javax.ws.rs.GET;
5. import javax.ws.rs.POST;
6. import javax.ws.rs.Path;
7. import javax.ws.rs.PathParam;
8. import javax.ws.rs.Produces;
9. import javax.ws.rs.QueryParam;
10. import javax.ws.rs.core.MediaType;
11. @Path("productcatalog")
12. public class ProductCatalogResource {
13. private static List productCatalog;
14. @GET
15. @Path("search/category/{category}")
16. @Produces(MediaType.APPLICATION\_TEXT)
17. public Product[] searchByCategory(@PathParam("category") String category) {
18. List products = new ArrayList();
19. for (Product p : productCatalog) {
20. if (category != null && category.equalsIgnoreCase(p.getCategory())) {
21. products.add(p);
22. }
23. }
24. return products.toArray(new Product[products.size()]);
25. }
27. @GET
28. @Path("search")
29. @Produces(MediaType.APPLICATION\_JSON)
30. public Product[] searchByName(@QueryParam("name") String name) {
31. List products = new ArrayList();
32. for (Product p : productCatalog) {
33. if (name != null && name.toLowerCase().startsWith(p.getName().toLowerCase())) {
34. products.add(p);
35. }
36. }
37. return products.toArray(new Product[products.size()]);
38. }
40. @POST
41. @Path("insert")
42. @Consumes(MediaType.APPLICATION\_JSON)
43. @Produces(MediaType.APPLICATION\_JSON)
44. public Status insert(Product product) {
45. productCatalog.add(product);
46. return new Status("SUCCESS", "Inserted " + product.getName());
47. }
48. יצירת קובץ web.xml

לחיצה ימנית על הפרויקט – java ee tools – generate ….

בתוך הקובץ שנוצר נוסיף:

<servlet>

<servlet-name>javax.ws.rs.core.Application</servlet-name>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>javax.ws.rs.core.Application</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

שיראה כך:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* version=*"3.1"*>

<display-name>webServiceExample</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>javax.ws.rs.core.Application</servlet-name>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>javax.ws.rs.core.Application</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

</web-app>

1. הרצת הפרויקט

* אופציה אחת דרך הIDE -

לחיצה ימנית על הפרויקט – run as – run on server

ואז השרת ירוץ בחלון למטה.

* אופציה שניה חיצוני-

לחיצה ימנית – Export –war file

ולקבוע יעד מיקום בתוך תקית ה apache tomcat /webapps, ולהרים את השרת דרך הקובץ הרצה startup.bat

1. עכשיו נלך ל browser כלשהו כמו chrome

וננסה לגשת ל : <http://localhost:8080> - נראה ש tomcat למעלה.

עכשיו ניגש לכתובות שנתנו בקוד לפונקציה ,

לדג לפי הקוד שנכתב למעלה:

https://localhost:8080/SampleWebProject /productcatalog/search/category/electronics

https://localhost:8080/SampleWebProject /productcatalog/search?name=Hammer

ונקבל בדף את מה שהפונקציה מחזירה

בהצלחה!!