**Restful Web Services (JAX-RS)**

Restful web services (JAX-Rs) זהו framework (-תשתית) ב-JEE המספק אפשרות להקמת web services (web api) בארכיטקטורת Rest בצורה קלה ומהירה.

**Web Services**

Web services אלו תכניות הרצות בשרת ומקבלות בקשות מקליינט מרוחק ע"י פרוטוקול HTTP. קיימים 2 סוגים שלweb services : Soap Web Services, Rest Web Services.

נפרט על Rest Web Services:

**Rest - Representational State Transfer**

ארכיטקטורת Rest (עובד בד"כ יחד עם פרוטוקול http) מיועד למעבר קל משקל ומהיר של בקשות ותשובות מהקליינט לסרבר.   
הגישה לסרבר שיטפל בבקשה היא עפ"י: **URI** - Uniform Resource Identifiers – לינק הנכתב בהמשך לכתובת ה-URL הבסיסית של השרת.   
הנתונים בפרוטוקול Rest נכתבים בד"כ בפורמט json.

דוגמא לפורמט json:

{

"id":1,

"name":"Ester"

}

דוגמא לשליפת תלמיד עם מזהה 1 ב-Rest תראה כך:

<http://localhost:8080/pupils/get/1>

**Http Protocol**

זהו פרוטוקול שימושי לתעבורת נתונים ברשת. מאפיינים עיקריים:

1. הפניה לשרת תהיה ע"י שימוש ב-URL בהתאם לתבנית הבאה: <http://IP_ADDRESS:PORT>

סרבר פשוט בג'אווה רץ בד"כ ב- port 8080, לדוגמא:

<http://localhost:8080>

1. פרוטוקול http מכיל כמה סוגי מתודות:

|  |  |  |  |
| --- | --- | --- | --- |
| **שם המתודה** | **תפקיד** | **מעלות** | **חסרונות** |
| get | אחזור נתונים ללקוח | מהיר | הנתונים חשופים ב-URL – בעיה מבחינת אבטחה.  בנוסף גודל הפרמטרים מוגבל. לכן משמש לאחזור נתונים ולא לאתחול נתונים. |
| post | אתחול נתונים בשרת | הנתונים עוברים ב-body של הבקשה – כך אינם חשופים, וכן הגודל האפשרי גדול יותר. |  |
| put | עדכון נתונים בשרת | כמו ב-post | כמו ב-post |
| delete | מחיקת נתונים בשרת | כמו ב-get | כמו ב- get |

1. סטטוס הבקשה:כאשר הבקשה מטופלת וחוזרת ללקוח, נשלח יחד עם התשובה גם סטטוס המציין את מצב הבקשה.

סטטוסים נפוצים לדוגמא:

|  |  |  |
| --- | --- | --- |
| **סטטוס** | **תיאור באנגלית** | **הסבר בעברית** |
| 200 | OK | הבקשה עברה בהצלחה |
| 204 | No Content | הבקשה עברה בהצלחה, לא חזרו ערכים בתשובה |
| 400 | Bad Request | התחביר של הבקשה לא חוקי – השרת לא הצליח להבין את הבקשה |
| 403 | Forbidden | אין הרשאה לצפיה בדף המבוקש לקליינט הנוכחי |
| 404 | Not Found | הדף המבוקש לא נמצא |
| 415 | Unsupported Media Type | פורמט נתוני הבקשה לא נתמך ע"י השרת, ולכן השרת דחה את הבקשה |
| 500 | Internal Server Error | השרת זרק שגיאה – באג בשרת |

**Restful Resource Class (- Controller)**

המחלקות שמטפלות בבקשות הקליינט נקראות Resources או Controllers. ה-Controllers הם למעשה מחלקות ג'אווה פשוטות (POJO – plain old java objects) שיש עליהן את האנוטציה @Path- המציינת את ה-URI בו יטפל ה-controller הנוכחי.

לדוגמא:

@Path("pupils")

public class PupilController

ה-Controllers מכילים מתודות עבור תתי URI, כל מתודה יכולה להכיל אחת או יותר מהאנוטציות הבאות:

|  |  |  |
| --- | --- | --- |
| **שם האנוטציה** | **תפקיד** | **דוגמת שימוש** |
| @Path | מציינת את ה-URI בו המתודה תטפל | @Path("/get")  @Path("/get/{id}") |
| @GET | מתודה זו היא מסוג Http Get | @GET |
| @POST | מתודה זו היא מסוג Http Post | @POST |
| @PUT | מתודה זו היא מסוג Http Put | @PUT |
| @DELETE | מתודה זו היא מסוג Http Delete | @DELETE |
| @PathParam | המרה בין פרמטר שהתקבל ב-URI לפרמטר במתודה. אנוטציה זו נכתבת בסוגריים של קבלת הפרמטרים במתודה. | (@PathParam("id") int id) |
| @Consumes | הצהרה על סוג הנתונים שהמתודה תקבל – ואותם תמיר לאוביקטים של ג'אווה | @Consumes(MediaType.APPLICATION\_JSON) |
| @Produces | הצהרה על סוג הנתונים שהמתודה תחזיר – ע"י המרת האוביקטים של ג'אווה לטיפוס המצוין | @Produces(MediaType.APPLICATION\_JSON) |