**Java Spring Standards**

1. לקרוא את [מסמך הסטנדרטים של טוויטר לכתיבה בג'אווה](https://github.com/twitter-archive/commons/blob/master/src/java/com/twitter/common/styleguide.md).
2. בכל מקום שטוויטר מציעים אך ורק אופציה אחת, אנחנו הולכים לפיהם. כאשר הם מציעים מספר אפשרויות, נציג כאן במסמך את האופציה שלפיה נפעל במדור.
3. אין הערות למעט אישור מיוחד.
4. במחלקה, הסדר הוא: שדות, בנאי ופונקציות.
5. במחלקה, הסדר הוא:
   1. Public static
   2. Private static
   3. Public
   4. Private

אך שימו לב שעדיין בכל מקרה שדות מגיעים לפני הבנאי, והבנאי לפני הפונקציות.

1. שורה רווח לפני ה-return, אלא אם זו השורה היחידה בפונקציה.
2. שורה רווח לפני ואחרי פתיחת בלוק (if, try…).
3. בהגדרה של פונקציה, אם השורה ארוכה מידי ויש צורך לרדת שורה, אז יורדים שורה לאחר כל פרמטר. דוגמה:

public String downloadAnInternet(Internet internet,

Tubes tubes,

Blogosphere blogs,

Amount<Long, Data> bandwidth) {

tubes.download(internet);

...

}

1. ארגומנטים של פונקציה תמיד נכתוב כ-final.
2. גם Local Variables נכתוב כ-final, אלא אם הם צריכים להשתנות. למשל משתנה של count, sum וכו' לא נכתוב כ-final, אך את כל השאר כן נכתוב בצורה הזו.
3. נדרש לכתוב בדיקות לכל פונקציה ברמת הרשאה public בשכבת הקונטרולר ובשכבת ה-BL.  
   בקונטרולר עושים בדיקות אינטגרציה מועטות כדי לבדוק שהתהליך הכללי עובד ויש חיבור בין כלל השכבות, כולל ה-DB.  
   בדיקות בשכבת ה-BL, עליהן להיות מעמיקות ולבדוק כמה שיותר מקרי קצה.  
   בשכבת ה-Repository, נעשה בדיקות רק במקרים של שליפות מיוחדות ובאישור מיוחד.
4. DTO – תמיד יוצרים מודלים שונים עבור in, out. הכלל של האם ליצור כמה שונים עבור in או out, הוא האם יש משמעות לוגית. למשל, האם שולחים רישיון ביקור לרש"פ ולרשב"ג, אז זה יהיה אותו DTO גם אם אחד מהם לא מעוניין באחד nהשדות שנשלף. לעומת זאת, כאשר יש אובייקט שונה עבור יצירת רישיון או עבור עדכון רישיון, אז כן נפריד.  
   בנוסף, כאשר מדובר במספר מועט של שדות, יכול להיות שנעדיף להעביר אותם כ-params ולא ליצור DTO.