**Build és futtatás Mavennel**

**(start.spring.io-t - > Maven; mvn clean package; pom.xml -> spring-boot-maven-plugin;; feladata: jar állomány átcsomaglása úgy, hogy tartalmazza a webkonténert is; elindítás: mvn spring-boot:run -> elindul a Tomcat, 080ason figyel, böngészővel megnyitás és oldal lekérése; ha nincs a gépen Maven, elég egy sajátgépes JVM és a parancs parancs: java -jar név.jar)**

Build és futtatás parancssorból

**ha start.spring.io-t használnuk a** projektünk vázának legenerálásához, akkor **ki lehet választani, hogy Maven** vagy Grader legyen a build tool

Mavennel -> Mavenes könyvtárstruktúrával hozza létre és pom.xmlel a könyvtárban

parancssorból való cleanhez: **mvn clean package** -> **letölri a target könyvtárat ha létezik,** és a package hatására, a build lépésekez elvégezve **létrejön a .jar állomány**, maga az alkalmazás

a **pom.xml** állomáynban fog szereplni a **spring-boot-maven-plugin;** ennek a feladata, hogy a létrejött **.jar állományt átcsomagolja úgy, hogy magát a webkonténert(Tomcat) is belecsomagolja** -**> az eredeti -jar állományt átnevezi jar.original-re, és létrejön egy új nagy jar állomány a Tomcat containerrel**

**alkalmazás elindítása Mavennel: mvn spring-boot:run** -> a springboot **plugin run golja** fog meghívódni, ami elindítja az alkalmazást, **elindul a Tomcat**, **figyelni fog az alapértelmezett 808as porton**, az **alkalmazásunkat böngészővel meg tudjuk nyitni,** le tudunk kérni belőle egy oldalt

**ha nincs az adott gépen Maven telepítve**, úgy is lehet futtatni a .jart, ha van a gépen java virtuális gép, ilyenkor a **parancs: java -jar név.jar;** ilyenkor automatikusan a Tomcat fog elindulni és rajta a springes alkalmazás

**Build és futtatás Mavennel – gyakorlat**

Ha projektünk standard Mavenes project; ezért **terminálból menni fognak a mavenes parancsok:**

**mvnw** (Maven wrapper; egy script, ami letölti a Mavent; a start.spring.io legenerálja ezt a scriptet a projektünkbe és nem kell a Mavent külön telepíteni

parancs: **mwnw clean package** -> töröl és packagel (fordít, unit tesztek, becsomagol jarba erőforrásokkal majd a plugin átcsomagolja a fentiek szertint) -> employees.jar.original és a másik jar is létrejött

**alkalmazás indítása: az IDEAban először leállítjuk a futót, hogy ne foglalja a 8080as portot**

**parancs: mvnw spring-boot:run** -> elindul az alkalmazás és a Tomcat vár a 8080as porton; ha nyomok egy **refresht az oldalon akkor szépen kiszolgálja a böngészőt**

**terminalba ctrl cvel megszakítom a futást**, és megpróbáljuk a **jart direktbe elindítani**:

**átváltás target könyvtárra és java -jar név.jar**; ugyanúgy futott és ugyanúgy kiszolgált

**Kérdések**

*Springbootos template generálásakor Maven struktúra választása esetén milyen plugin kerül be a függőségke közé és mi a szerepe?*

*Hogy indítjuk el a springbootos alkalmazást Mavennel parancssorból?*

*Hogy indítjuk el a springbootos alkalmazást parancssorból simán JVM-el?*