



10C: Inversion of control, a Central megfordítás / Hollywood elv; Ez a fontos elv úgynevezett levetíti a terhet a vállunkról, tehát nem az van, hogy a saját programom fogja látni az előre megírt dolgokat; hanem a leírt eredmények van az irányítás, és amikor az én időmre kerül a sor, akkor én fogom meg az eredményt.

Dependency Injection (DI). ~~A függő, de nem saját tulajdonság, hanem valaki másé, aki megadja a megoldást.~~

Azért nem a megírt uddon történet, hanem fordítva. Nem az üzleti logikánál történet a leírás a rendszer felé, hanem fordítva pl. Egy függvénynek megvalósítást az állapot, de csak utána ~~történet~~ hívódik meg meg a függvény.

Dependency Injection / Függőség-injektálás.

Az IoC speciális eset: az objektum létrehozásáért nem a mi kódunk felelős, hanem a framework. Technikaiilag: a framework eldönti, hogy melyik objektumra van szükség, és ezt a létrehozást elvégzi. Ez nem jelenti azt, hogy a framework nem tudja, hogy az objektum melyik osztályból kell létrehozni. A frameworknek van egy listája az objektumokról, és ez alapján tudja, hogy melyik objektumra van szükség. A frameworknek van egy listája az objektumokról, és ez alapján tudja, hogy melyik objektumra van szükség. A frameworknek van egy listája az objektumokról, és ez alapján tudja, hogy melyik objektumra van szükség.

② ~~HTML~~ HTML, HTML 5 front end, megjelenítés, minden alapja  
CSS: formázási lehetőségeket ad

JSON, JQuery front end fejlesztési eszközök

Back end: node.js, php, java, C++, C#, python; programozási nyelv, eszköz

MySQL, MongoDB: adatbázisokhoz eszközei

verziókövető rendszerek: git, SVN

Authentication: hogyan ellenőrizni, hogy ki akar és milyen erőforráshoz hozzáférni. Általában ~~egy~~ az egy felhasználó - jelszó kombinációval oldjuk meg. Ha meg volt az authentication, akkor azután lehet authorizationt elvégezni.

Authorization: miután bizonyítjuk a jelszót a felhasználónak, hogy mihez és hogyan nyúlhat hozzá, vagy sem, mit használhat, láthat stb...

login/logout: megvalósítható:

Form authentication

Digest authentication

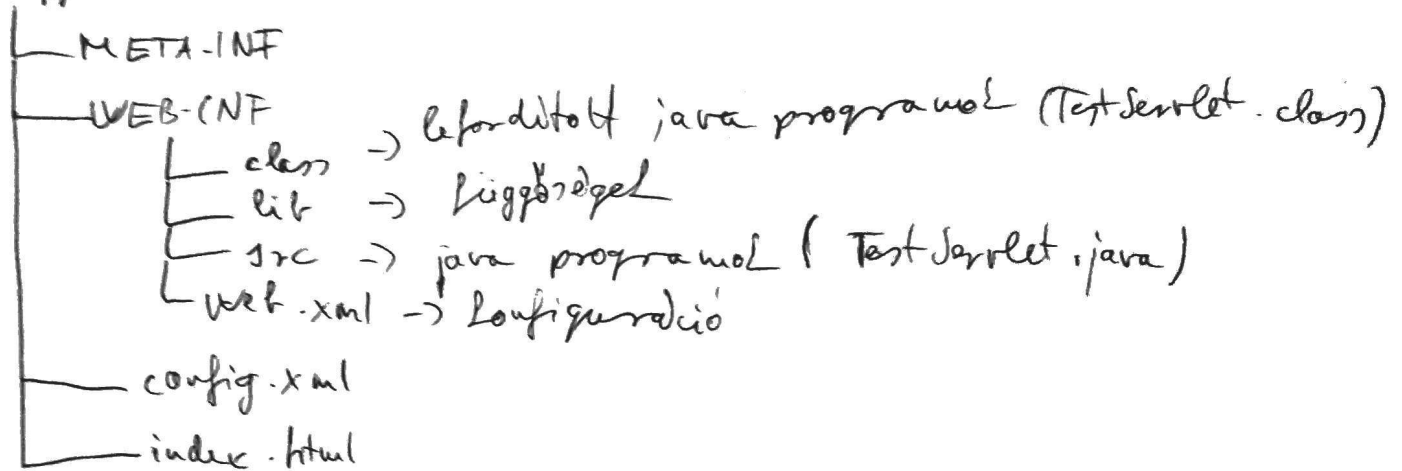
Basic authentication

In-memory authentication

ADBC authentication

## ③ Tomcat, servlet

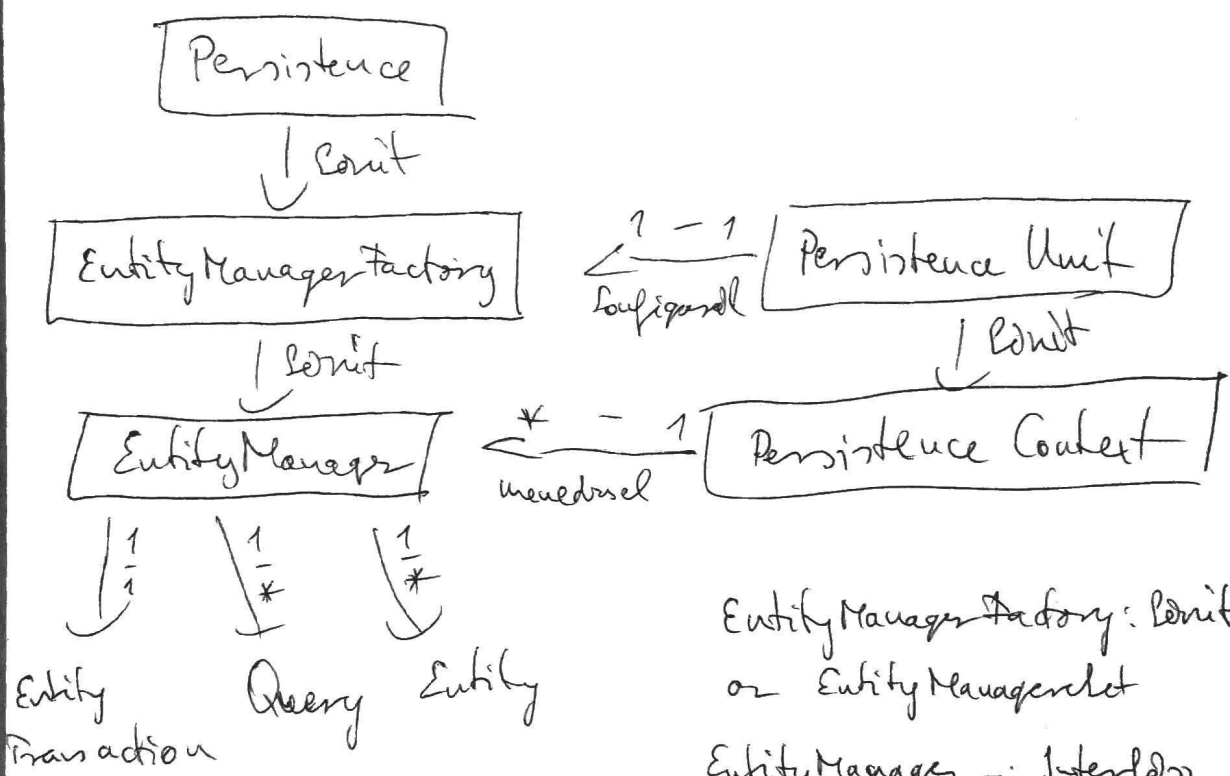
webapps



A tomcat lap egy requestet, ami után példányosítja a servletet az  
init() módszerrel

# ④ JPA

Java persistence API, a perszisztenciai foglalkozik, azaz a mechanizmusokkal, amik által a hirtelentott objektumok tartoznak az objektumok peldakapott procemelet. A JPA-val meg lehet adni hogy melyik objektum, is hogyan legyen perszisztens.



EntityManagerFactory: konit is menedzseli az EntityManagert

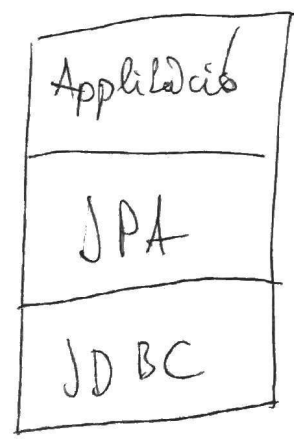
EntityManager: Interfész ami kereli az objektumok perszisztencia mndreletit

Entity: a perszisztencia objektum

EntityTransaction: mndreletet tartja felel

Query: relacio objektumokat listozik

Persistence: ontaly, statikus metódusokat tartalmaz



↑  
Relacio  
Adatbaze