1. Titlu

2. Cine sunt eu ?

3. Cine sunteti voi ?

* Ce va aduce aici ?
* De ce sunteti interesati de subiect ?
* Cine a mai auzit de Spring Data ?
  + Nr de persoane
* Cine a mai lucrat cu Spring Data ?
  + Nr de persoane

4. Motivatie

1. Implementarea data access layer-ului propriu cu API-uri specifice tehnologiilor, iar uneori API-urile nu sunt chiar usor de folosit
2. Foarte multe modalitati de implementare a DAO-ului
3. Din aceste cauze ajungem la Boilerplate
4. Setarea asteptarilor

5. Solutii

1. Spring Data JPA ofc
2. Obiectivul sau este de a oferi o cale mai usoara de a dezvolta aplicatii care utilizeaza RDB sau chiar tehnologii mai noi
3. Nu e limitat la JPA. Are suport pentru MongoDB. Redis, Solr, REST si multe altele

6. Good (taiat), old (vazut) boilerplate – Modalitatile curente

1. Mecanismul de persistenta din EJB (heavyweight, configurari complicate XML, foarte mult boileplate)
2. JDBC API – deployable in orice servlet container – dar foarte mult boilerplate – trasnferarea domain modelului in queriuri si resultatele din nou in domain model (Prepared statements)
3. ORM – Hibernate – adesea o alegere buna deoarece ne scapa de scrierea de cod JDB care nu este necesar, dar inca e prezent boilerplate-ul – JPQL, HQL, Criteria API
4. JPA – mecanism standard pentru layerul de persistenta in RDB

7. Take off

1. Spring Data – proiect din portofoliul spring
   1. Faciliteaza accesul la date in multiple data store-uri
      1. Construim aplicatii si avem nevoie sa ne stocam informatiile. O facem in mod normal in RDM sau NoSQL si apoi aplicatia trebuie sa acceseze acele date. Aici e sweet spot-ul Spring Data.
   2. Furnizeaza o abordare simpla si consistenta care usureaza modalitatea de lucru cu persistenta datelor
   3. Proiect umbrela pentru mai multe module cum am spus si mai sus
      1. Deobicei un modul corespunde unui datastore
   4. Primul release a fost in 2010 la Spring One a avut originea intr-o sesiune de hacking a lui Rod Johnson (SpringSource) si Emif Eifrem (Neo Technologies) ce avusese loc un pic mai devreme in acel an. Au vrut sa integreze Neo4j graph database cu Spring Framework.
      1. Este un produs matur
      2. Se adauga module imediat cum alte datastore`urile apar
   5. Reduce efortul necesar pentru a pune Persistence Layerul in place.
      1. JDBC – Prepared statements, open/closing connections etc
   6. Da predictibilitatea codului

8. Cum folosim Spring Data Repositories

1. Ce spune Spring:
   1. The goal of Spring Data repository abstraction is to significantly reduce the amount of boilerplate code required to implement data access layers for various persistence stores.
2. Interfata centrala in abstractizarea repository-ului Spring Data este ... Repository