Představení backendu pro web kivu

Radek Muzika

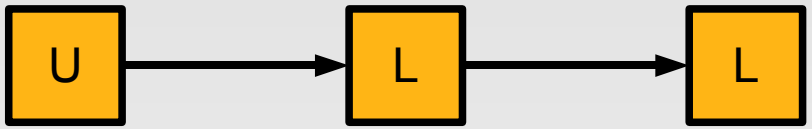
Spring JDBC

- podpora tvorby persistenční vrstvy (ORM,JDBC)

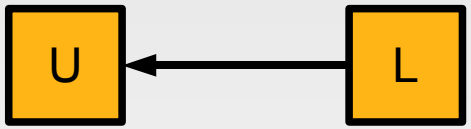
- hlavní přínos pro práci s jdbc:
  - méně kódu (odstraňuje TCFTC)
    - zjednodušení práce
  - odstraňuje „memory leak“ způsobený neuzavíráním přípojení do db
    - Rod Johnson - „Sackable offence“ - důvod k vyhazovu
  - ...

Spring IOC container

- implementace IOC návrhového vzoru
  - "don't call us, we will call you"
  - normal flow:



  - IOC flow:

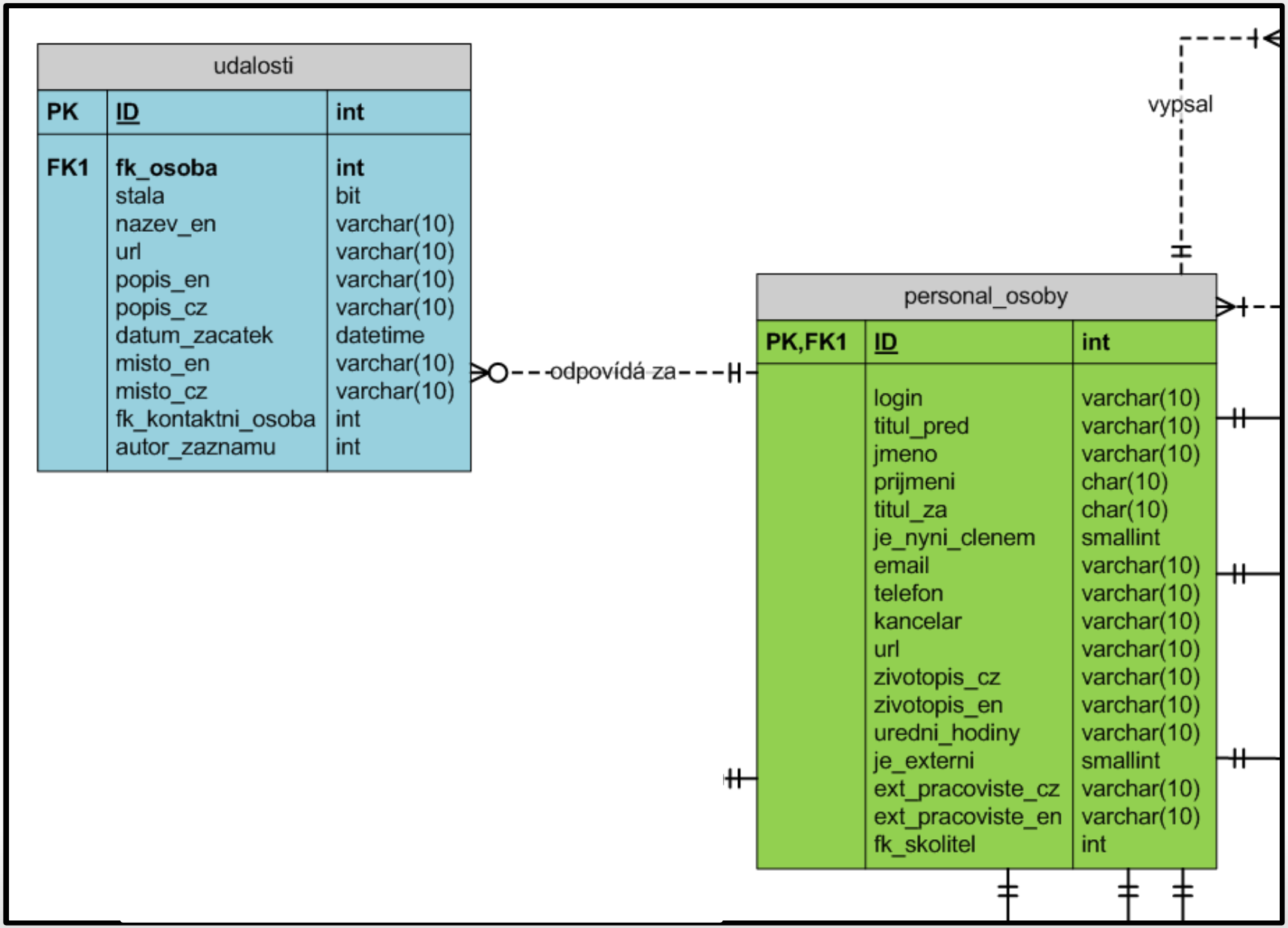


- přínos: vynucuje separaci vrstev, snažší testování (Mock objekty)

Architektura řešení

- doménový model
  - je otiskem datového modelu (POZOR: neobsahuje foreign keys)
- DAO rozhraní
- implementace DAO pro JDBC
  - lazy-load => nutnost přetížení doménových tříd
    - *cz.zcu.kiv.db.dao.jdbc.model*

Architektura řešení



Tutoriál – Jak získat data

1) získání aplikačního kontextu Springu
   - přístup k IOC kontejneru => přístup k beanám

```
ApplicationContext ctx =
        WebApplicationContextUtils.getWebApplicationContext(request.getSession().getServletContext());
```

Tutoriál – Jak získat data

2) získání DAO objektu UdalostDao

```
UdalostDao udalostDao = (UdalostDao) ctx.getBean("udalostDao");
```

- *udalostDao* je identifikátor beany v IOC cont.

```
@Repository("udalostDao")
public class UdalostDaoImpl implements UdalostDao {
        ...
}
```

Tutoriál – Jak uložit data

- ukládání/mazání dat přináší jednu nezvyklost
  - ➔ inicializaci dat, které nejsou z formuláře k dispozici
- příklad:

```
Udalost udalost = new Udalost();
udalost.setId(Integer.parseInt((String)request.getParameter("id")));
...
OsobaDao oDao = (OsobaDao) ctx.getBean("osobaDao");
Osoba o = oDao.getOsoba(Integer.parseInt((String)request.getParameter("fkOsoba")));
udalost.setOdpovednaOsoba(o);
UdalostDao uDao = (UdalostDao) ctx.getBean("udalostDao");
uDao.ulozUdalost(udalost);
```

Představení backendu pro web kivu

Děkuji za pozornost.