

то что мы написали -- с рутингом и т п -- это "микроскопический недофреймворк"

```
<form method="post" action="">
  <input type="hidden" name="action" value="register">
  <div class="field">
    <div class="name">
      <label for="login">Login</label>
    </div>
    <div class="value">
      <input id="login"/>
    </div>
  </div>
  <div class="field">
```

доменная сущность -- юзер

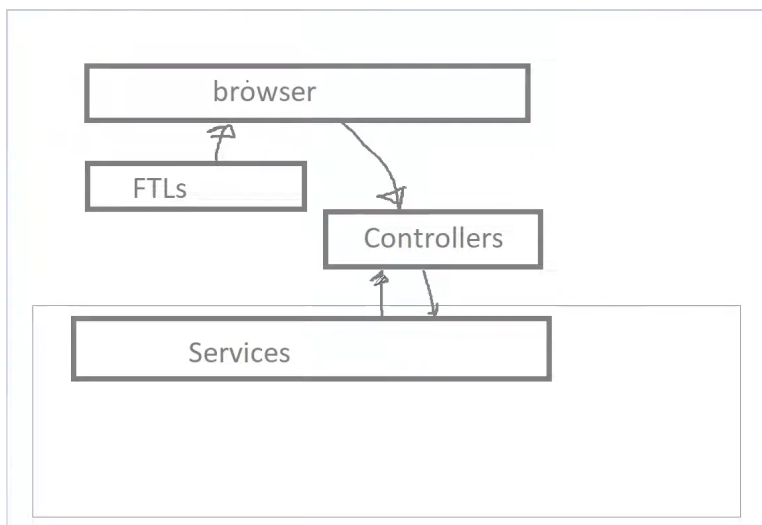
сервисы -- находятся в слое модели

хорошая идея делать сервисы domain-ориентированными

ну т е юзер -- юзерСервис

в целом стоило написать интерфейсы и их реализации а не сразу классы сервисов но мы торопимся так что ежжи

не путать сервисы и контроллеры!



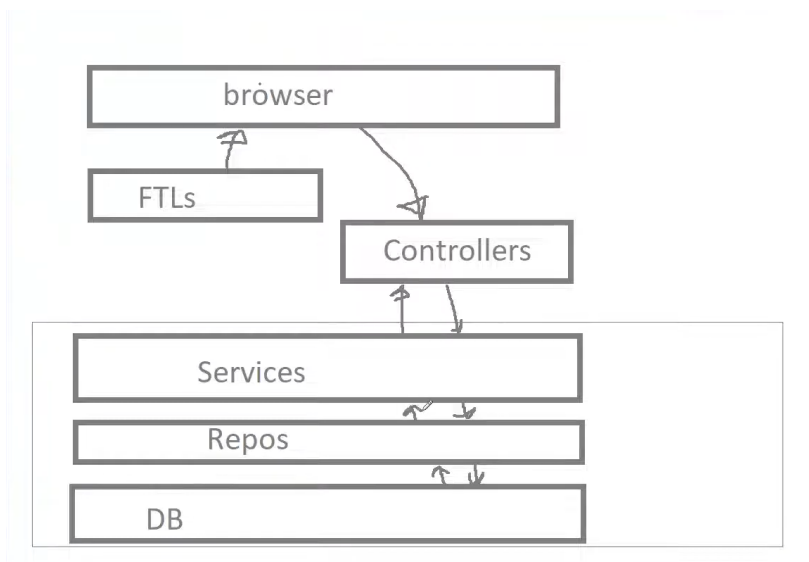
возникает понятие validation exception

библиотека guava

-- чтобы удобные утилитки для стрингов

работа с временем -- прерогатива бд, потому что томкатов у нас может быть запущено допизды а бд по идее одна (как минимум у нее один источник времени)

mariaDB



про соль в паролях

просто хэшить пароли не достаточно потому что

1. облегчается реверсинжиниринг (см радужные таблицы)
2. если кто-то вставляет пароль 12345 мы сразу увидим этот хэш
соль это что-то разное для разных паролей, например имя юзера
всегда стоит проектировать систему с расчетом на то что данные могут утечь

упражнение:

Почему не стоит использовать время регистрации как соль? (2 причины)

JDBC

[jdbc](#)

к нему нужен драйвер

бд

primary-ключ

индекс дерево поиска

всегда юзать prepared statement-ы