Staying Ahead of the Curve

Trisha Gee, Java Developer at MongoDB
or:

New Technology is Dangerous, mmmkay
Let’s Learn Lambdas!
erm...
Let’s Go Parallel!
Wait a minute...
Why?
For The Business
Why?

- Customisable dashboards
- A new market squeezes your downtime window
- Trying to store different shaped data in a single table
<table>
<thead>
<tr>
<th>name</th>
<th>address</th>
<th>operator</th>
<th>vegan</th>
<th>weather</th>
<th>full name</th>
<th>inclusion</th>
<th>name_loc</th>
<th>opening_hours</th>
<th>toilets</th>
<th>access</th>
<th>currency</th>
<th>other</th>
<th>adt</th>
<th>postcode</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starbucks</td>
<td>12345</td>
<td>healthy</td>
<td>yes</td>
<td>sunny</td>
<td>Starbucks</td>
<td>yes</td>
<td>80000</td>
<td>Mo-Su 07:00-20:00</td>
<td>yes</td>
<td>5000</td>
<td>1234</td>
<td>no</td>
<td>80000</td>
<td>survey</td>
<td></td>
</tr>
<tr>
<td>Pizza Hut</td>
<td>67890</td>
<td>casual</td>
<td>yes</td>
<td>cloudy</td>
<td>Pizza Hut</td>
<td>yes</td>
<td>98765</td>
<td>Mo-Su 10:00-22:00</td>
<td>yes</td>
<td>7890</td>
<td>6789</td>
<td>no</td>
<td>67890</td>
<td>Survey</td>
<td></td>
</tr>
<tr>
<td>McDonald's</td>
<td>11223</td>
<td>fast food</td>
<td>yes</td>
<td>windy</td>
<td>McDonald's</td>
<td>yes</td>
<td>34567</td>
<td>Mo-Su 06:00-24:00</td>
<td>yes</td>
<td>5678</td>
<td>4567</td>
<td>no</td>
<td>5678</td>
<td>Personal</td>
<td></td>
</tr>
</tbody>
</table>

For more information, visit [Starbucks](http://www.starbucks.com) or [Pizza Hut](http://www.pizzahut.com) or McDonald's [website](http://www.mcdonalds.com).
Now What?

• Identify possible technology options, e.g:
  • HTML5/JavaScript
  • DevOps tools
  • NoSQL
Approaches

- Spikes / prototypes
- Use in tests
- Use in non-production code
  - Build / Deploy / Continuous Delivery Pipeline
- Use in internal tools
- Use on a new project
- Get the experts in (remember to share)
For The Team
Why?

- Mixed team used to different programming styles
- Releases are painful
- Databases schema changes take ages
- Testing is hard and inconsistent
@Test
public void testFindOne() {
    DBCollection c = collection;

   DBObject obj = c.findOne();
    assertEquals(obj, null);

    obj = c.findOne();
    assertEquals(obj, null);

    obj = c.findOne();
    assertEquals(obj, null);

    // Test that findOne works when fields is specified but no match is found
    // *** This is a Regression test for JAVA-411 ***
    obj = c.findOne(null, new BasicDBObject("_id", true));

    assertEquals(obj, null);

    DBObjct inserted = BasicDBObjectBuilder.start().add("x", 1).add("y", 2).get();
    c.insert(inserted);
    c.insert(BasicDBObjectBuilder.start().add("_id", 123).add("x", 2).add("z", 2).get());

    obj = c.findOne(123);
    assertEquals(obj.get("_id"), 123);
    assertEquals(obj.get("x"), 2);
    assertEquals(obj.get("z"), 2);

    obj = c.findOne(123, new BasicDBObject("x", 1));
    assertEquals(obj.get("_id"), 123);
    assertEquals(obj.get("x"), 2);
    assertEquals(obj.containsField("z"), false);

    obj = c.findOne(new BasicDBObject("x", 1));
    assertEquals(obj.get("x"), 1);
    assertEquals(obj.get("y"), 2);

    obj = c.findOne(new BasicDBObject("x", 1), new BasicDBObject("y", 1));
    assertEquals(obj.containsField("x"), false);
    assertEquals(obj.get("y"), 2);
Now What?

- Identify possible technology / process options, e.g:
  - Java 8, Groovy
  - DevOps
  - NoSQL
  - Modern Test Frameworks (e.g. Spock, Lambda Behave)
def 'should return null when findOne finds nothing'() {
    expect:
    collection.findOne([field: 'That Does Not Exist']) == null
}

def 'should return null when findOne finds nothing and a projection field is specified'() {
    given:
    collection.drop()

    expect:
    collection.findOne(null, [_id: true] as BasicDBObject) == null
}

@Unroll
def 'should return #result when performing findOne with #criteria'() {
    given:
    collection.insert([_id: 100, x: 1, y: 2] as BasicDBObject)
    collection.insert([_id: 123, x: 2, z: 2] as BasicDBObject)

    expect:
    result == collection.findOne(criteria)

    where:
    criteria | result
    123       | [_id: 123, x: 2, z: 2]
    [x: 1] as BasicDBObject | [_id: 100, x: 1, y: 2]
}

@Unroll
def 'should return #result when performing findOne with #criteria and projection #projection'() {
    given:
    collection.insert([_id: 100, x: 1, y: 2] as BasicDBObject)
    collection.insert([_id: 123, x: 2, z: 2] as BasicDBObject)

    expect:
    result == collection.findOne(criteria, projection)

    where:
    criteria | projection | result
    123       | [x: 1] as BasicDBObject | [_id: 123, x: 2]
    [x: 1] as BasicDBObject | [y: 1] as BasicDBObject | [_id: 100, y: 2]
Approaches

• Spikes / prototypes
• Use in tests
• Use in non-production code
  • Build / Deploy / Continuous Delivery Pipeline
• Use in internal tools
• Skunkworks projects / hack days
For You
Why?

- I want to have fun
- I want a new job
- I want to be a Technology Guru in this job
- I don’t want to be left behind
Now What?

• Any technology that looks awesome

• Buzz words from job adverts

• Any technology that is currently (or soon to be) used at work

• Any/all of the above. Or None.
Applet Viewer: Fortune

Tell My Fortune

Applet started.

Choose Color

Please choose a colour:

Red

Blue

Green

Yellow

OK
Search Facilities

1. Search for a title
2. Search for a format
3. Search for a subject
4. Search for a student
5. Search for a teacher
6. Search for overdue items
Approaches

• User Groups
• Online courses
• Pet project
• Join an open source project
• Find a project at work
• Get a new job!
Why?

• Because you want to
Being Proactive
Preventing Premature Implementation

• Hack days
• Skunkworks
• 20% time
• Conferences
• Brown Bags
• User Groups
Benefits

• Skilled Developers

• Ready to adopt the right technology at the right time

• Happy Developers
Summary of Approaches

For the business:
- Internal tools
  - Embedded in the team
  - Pairing
  - Brown bags
  - Training
  - Share the knowledge
    - Might be free
  - Business facing
- Use in non production code
  - Ops
  - Dev
  - Dev tools
- Use in non production code
- Tests
- Build
- Deploy
- Continuous Delivery pipeline

For the team:
- Spike / prototype
- Skunkworks projects
- 20% time

For you:
- "Sweat equity"
- New job!
- Startup
- Pet project
- Work project
- Online courses
- User groups
- Open source projects
- Pet project
- Work project
So What Can I Do?
Don’t Panic!
Find the Why

Select & Test the Technology

Does it address the problem?

Nope

Yes

Level up!
Lambdas: The Right Way
@trisha_gee