Coding without Complexity

@ellenchisa
Agenda

Introduction

Complexity Today

Write Code

Removing Complexity
Use software.
Expectation

“This is a simple backend!”

Persistent Data Store

/checkins POST

/checkins GET

/emojicount GET
Reality

So many options.

- Open Cloud Functions, close it
- Need a backend...
- Get started on Rails
- API generator
- Check dependencies & versions
- Works locally
- Repo
- Set up Render to host the repo
- Need some sort of data storage
Receive data
Process data
Store/query data
Send data
No Silver Bullet
How many hours per week do you estimate developers at your company waste on maintenance (i.e. dealing with bad code / errors, debugging, refactoring, modifying)?

- **17.3 hours**
  Mean (all countries)

- **20.9 hours**
  Mean in **France** (highest)
Accidental complexity everywhere.
Accidental Complexity

- Infrastructure
- Deployment
- APIs
Coding without accidental complexity.
let range = List.range 1 100
List.map range \i → if i % 15 == 0 then "Fizzbuzz" else if i % 5 == 0 then "Buzz" else if i % 3 == 0 then "Fizz" else toString i
Time to write code!
No Accidental Complexity

- Compiled Infrastructure
- Deployless
- API focus
Infrastructure Complexity
```javascript
let user = {
  name: "John Doe",
  dob: Date.parse("1986-09-15T00:00:00Z"),
  email: "john@doe.com"
}

DB::set(user) DB::generateKey("Users")
```

```java
DB::getAllWithKeys("Users")
```

```javascript
DB::get("user")
```

```java
HTTP /api/users/:userid GET
```

```java
let user = DB::get("user")
sendEmail(user.email, "Your daily summary" messageFor user)
```
AWS Lambda

Upload your code to AWS Lambda or write code in Lambda's code editor

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

Lambda runs your code only when triggered, using only the compute resources needed

Just pay for the compute time you use
```swift
let comment = request.body.comment
let email = request.body.email
let host = request.body.sender
let name = request.body.name
let username = request.body.username
let password = DarkInternal:::upsertUser(username email name
let user = {
  email: email
  name: name
  username: username
}
let sendEmailKey = DB::add"{Jobs
  onUser: username
  task: "send-welcome-email"
  status: 0
  msg: ""
```
Cloudcraft

- RDS: Master
- Engine: MySQL
- Instance type: T2 - Burst
- Size: Micro
- Cost: $12.24 / mo

- mysql-db
  - Publicly accessible
  - Available
  - MySQL 5.6.27
  - 5 GB allocated, SSD
  - Not encrypted
  - US-east-1e zone
  - Multi-AZ Deployment
  - 1 Read replica
    - mysql-db-replica
Deployment Complexity
Make changes with feature flags

- Make change behind feature flag
- Integration test
- Code review

Gradual deploy with feature flags

- Enable feature flag for team
  - 10%
  - 50%
  - 100%

Remove feature flag
let _ = emit1 request.body "sendText"
let _ = emit1 request.body "sendDM"
DB::set1 request.body DB::generateKey 'Visits

flag name

Condition (run Case B if...)

Case B ✅
HTTP

/visit
POST

let _ = emit1 request.body "sendText"
let _ = emit1 request.body "sendDM"
DB::set1 request.body DB::generateKey Visits

also-send-email

Condition (run Case B if...)
request.headers.ff-send-email = true

Case B
let _ = emit1 request.body "sendText"
let _ = emit1 request.body "sendDM"
let _ = emit1 request.body "sendEmail"
DB::set1 request.body DB::generateKey Visit
HTTP

/visit

POST

let _ = emit1.request.body "sendText"
let _ = emit1.request.body "sendDM"
let _ = emit1.request.body "sendEmail"
DB::set1.request.body DB::generateKey "Visits"
API Complexity
The first API composition platform

Empowering developers to easily build apps in the expanding API ecosystem. **Focus on the fun, not the tedium.**

**Connect**
Instantly connect to APIs, and avoid the overhead of authentication, pagination, retries, and related grunt work.

**Compose**
Use SQL and JavaScript to explore data in real time, bringing the power of a relational database to your diverse data sources.

**Collaborate**
Build apps you can quickly share with teammates, collaborate on the code, safely share credentials, publish with zero-ops deployment.
Send text with `body` to phone number `phoneNumber` from number `fromNumber`, authenticated via `accountSID` and `authToken`.

```
let hostname = event.host
let host = DB::queryOne("Hosts
  name: hostname
")

let hostphone = host.phone
let message = event.visitor
  > (++) “is here to see you!”

Twilio::sendText(twilioSID, twilioToken, twilioFrom, hostphone, message)
```
```javascript
let checkins = DB::query2{
  deviceid: request.queryParams.device
}

let allemojis = List::foreach checkins
    checkin →
    checkin.emojis
    String::join ""
    String::toList1
    List::foreach
    a →
    toString a

let freq = List::fold allemojis Dict::empty
tracker mood →
match Dict::get1 tracker mood
  Just count → Dict::set tracker mood count + 1
  Nothing → Dict::set tracker mood 1

let emojis = Dict::keys freq
List::foreach emojis
  e →
  {
    emoji: e
    count: Dict::get freq e
  }

List::sortBy
  var →
  var.count
List::reverse
```

<html lang="en">
<head>
  <title>Hello!</title>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- import the webpage's stylesheet -->
  <link rel="stylesheet" href="/style.css">
  <!-- import the webpage's javascript file -->
  <script src="/script.js" defer></script>
</head>
<body>
  <h1>Hi there!!!</h1>
  I'm your cool new webpage. Made with <a href="https://glitch.com">Glitch</a>!
  <!- include the Glitch button to show what the webpage is about and to make it easier for folks to view source and remix -->
  <div class="glitchButton" style="position:fixed;top:20px;right:20px;"></div>
  <script src="https://button.glitch.me/button.js"></script>
</body>
</html>
```python
1) def factorial(n):
2)     if n == 0:
3)         return 1
4)     else:
5)         return n * factorial(n - 1)
6) print(factorial(7))
```
No Accidental Complexity

- Compiled Infrastructure
- Deployless
- APIs work like Functions
What problem am I solving?
What complexity can I remove?
thanks!

https://darklang.com || @ellenchisa