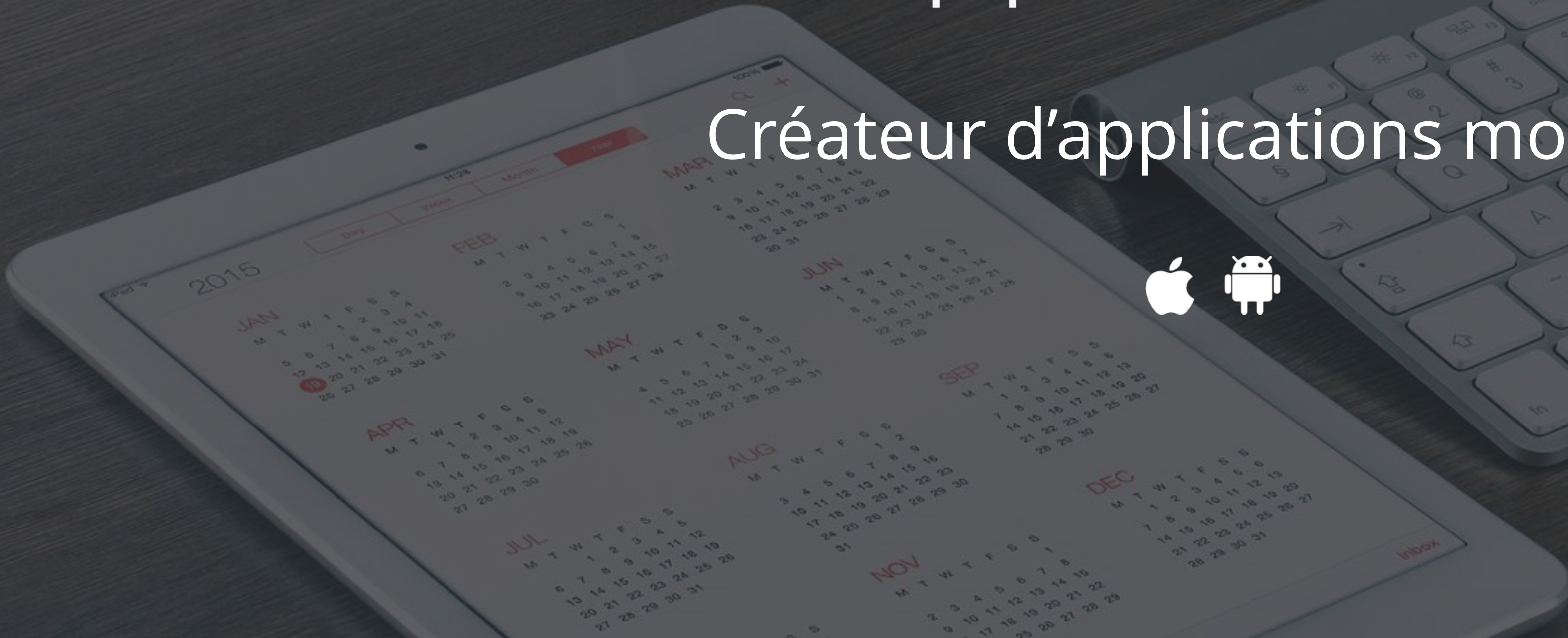




ripple motion

Créateur d'applications mobiles





# TECH TALK

GRATUITS & OUVERTS AU PUBLIC

Tech talk sous forme de courtes formations  
le **mercredi** de **17h** à **18h** !

N'hésitez pas à nous **contacter** afin de participer  
aux **prochaines sessions** au sein de nos locaux !



---

# TECH TALK

ANIMÉ PAR



**OLIVIER TABONE**

**CO FONDATEUR / DIRECTEUR**

**[olivier.tabone@ripplemotion.fr](mailto:olivier.tabone@ripplemotion.fr)**



TECH TALK #17 - 17.05.2017



---

# AUTOMATED TESTING

## FROM THE TRENCHES



# AUTOMATED TESTING FROM THE TRENCHES



**OLIVIER TABONNE**  
CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Essence

- Depuis 2008
- Environ 4M Téléchargements
- 
- Carburant + Lavage




**Aperçu d'iTunes**

**Essence – comparateur des prix des carburants**  
Par Ripple Motion

Ouvrez iTunes pour acheter et télécharger des apps.

**Description**  
Trouvez la station service la moins chère avec Essence. Plus de deux millions de personnes ont déjà téléchargé cette application. Les prix des carburants changent presque tous les jours, la station la moins chère également. Grâce à cette application, vous pouvez trouver la station la moins chère en quelques secondes.

**Assistance : Essence – comparateur des prix des carburants**

**Nouveautés de la version 3.2**  
Correction de bugs

**Captures d'écran**

**Gratuit**  
Catégorie : Navigation  
Mise à jour : 12 mai 2017  
Version : 3.2  
Taille : 127 Mo  
Langues : Français, Allemand, Anglais, Espagnol, Italien  
Éditeur : Ripple Motion  
© Ripple Motion  
Classé 4+

**Compatibilité** : Nécessite iOS 9.0 ou une version ultérieure. Compatible avec l'iPhone, l'iPad et l'iPod touch.

**Note**  
Nous n'avons pas reçu suffisamment de notes pour évaluer la moyenne de la version actuelle de cet article.  
Toutes les versions: ★★★★★ 1051 notes

**Classement des achats intégrés**

Station	Distance	Prix
AVIA RUEE SARL	7 km - 11 min	1,150
CARREFOUR ST HERBLAIN	3 km - 6 min	1,159
ELECLERC ATLANTIS	3 km - 7 min	1,159
SODIRETZ	8 km - 12 min	1,159
AUCHAN SAINT HERBLAIN	7 km - 10 min	1,159
Super U COUERON	8 km - 10 min	1,159
RELAIS ST HERBLAIN LES ECOBUTS	0.4 km - 2 min	1,162
RELAIS NANTES LE TERTRE	1.7 km - 6 min	1,162
RELAIS ST HERBLAIN LES ECOBUTS	0.4 km - 0.4 km, + 2 min	1,222
CASINO HERNIA	0.7 km - 1.2 km, + 4 min	1,185
LECLERC DRIVE TOURNEBRIDE	12 km (+ 0.5 km, + 1 min)	1,159
ELECLERC ATLANTIS	3 km (+ 0.6 km, + 7 min)	1,159
RELAIS ST HERBLAIN LES ECOBUTS	0.4 km (+ 0.4 km, + 2 min)	1,162



**OLIVIER TABONNE**  
CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

## AUTOMATED TESTING FROM THE TRENCHES

**CODE IS EASY TO WRITE,  
HARD TO MAINTAIN.**

**TESTING IS A TOOL TO  
MANAGE TECHNICAL DEBT.**



**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Shades of TestS

- **Unit Tests** (very fast, many)
- White box testing, ensure correctness of code at the method level
- **Functional Tests** (fast, many)
- Gray box testing, ensure correctness at subsystem level
- **UI Tests** (slow, few)
- Black box testing, ensure correctness at application level





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Anatomy of a test

- Given some preconditions
- When I perform some specific actions
- Then the system state has changed to an expected state.



# AUTOMATED TESTING FROM THE TRENCHES

## Anatomy of a test



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

```
13
14 class AutoWashDeliveryTests: MVMCTestCase {
15
16     /**
17     * Given any order that has been delivered
18     *
19     * When I Fetch it's related deliverable
20     *
21     * Then I get an autowash delivery
22     *   with a non empty code
23     */
24     func testRetrieveDelivery() {
25         XCTAssertNoThrow {
26             let (order, _) = try self.anyOrder(delivered: true)
27             let result = try order.fetchDelivery().get()
28             guard case let .autoWash(delivery) = result else {
29                 XCTFail("an autowash delivery was expected, got \(result)")
30                 return
31             }
32             expect(delivery.code).toNot(beEmpty())
33         }
34     }
35 }
```





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## What makes a good Test

- Automated
- Reliable
- Failure contains information
- Easy to read
- Fast
- Easy to write

## What makes a good Code Base

- Good Tests



**LET'S WRITE  
SOME TESTS**





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + PASSWORD

- **Given ...**
- **When ...**
- **Then ...**





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + PASSWORD

- **Given any email + password (that does not exist on the server)**
- **When ...**
- **Then ...**





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + PASSWORD

- **Given any email + password (that does not exist on the server)**
- **When I Create Account (POST /account/create with user & password params)**
- **Then ...**





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + PASSWORD

- **Given any email + password (that does not exist on the server)**
- **When I Create Account (POST /account/create with user & password params)**
- **Then I get a success (HTTP 201)**



# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + PASSWORD



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

```
37     /**
38     * Accounts can be created with email + password
39     *
40     * Given:
41     *   - any email that has not been registered on the server
42     *   - any valid password
43     *
44     * When I create account
45     *
46     * Then the call succeeds
47     */
48     func testRegisterNewAccount() {
49         XCTAssertNoThrow {
50             let email = "\(UUID().uuidString)@example.com"
51             let password = "totototo"
52             try self.api.create(account: .email(email: email, password: password)).get()
53         }
54     }
55
```





OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

## AUTOMATED TESTING FROM THE TRENCHES

**CTRL + ALT + CMD + G  
TO RUN LAST TEST  
CONFIGURATION.**

Public Service Announcement





**OLIVIER TABONNE**  
CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL + Empty PASSWORD

- Given any email (that does not exist on the server) and an empty password
- When I Create Account (POST /account/create with user & password params)
- Then I get an error (HTTP 400) indicating password is invalid

# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Account Creation WITH EMAIL + Empty PASSWORD

```
56  /**
57   * Accounts can't be created with an empty password
58   *
59   * Given:
60   *   - any email that has not been registered on the server
61   *   - an empty password
62   *
63   * When I create an account
64   *
65   * Then:
66   *   - an HTTP 400 error is raised
67   *   - the response contains an error message for the `password` field
68   */
69  func testRegisterFails_EmptyPassword() {
70    let email = "\(UUID().uuidString)@example.com"
71    let password = ""
72    let error = XCTAssertThrow {
73      try self.api.create(account: .email(email: email, password: password)).get()
74    }
75    expect(error).to(beHTTP(status: 400))
76    expect(error).to(containHTTPError(field: "password"))
77  }
```



# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## GiveAway: XCT Assert Throw

```

21 /**
22  * Function that will fail a test if `expression` does not raise an exception.
23  *
24  * This will fail with `exception not thrown` error message
25  *
26  * ...
27  * let error = XCTAssertThrow {
28  *     return
29  * }
30  * ...
31  *
32  * This will not fail, and error can be inspected
33  *
34  * ...
35  * let error = XCTAssertThrow {
36  *     throw SomeError
37  * }
38  * expect(error?.code) == 1 // inspect error
39  *
40  *
41  */
42 public func XCTAssertThrow(file: StaticString = #file, line: UInt = #line,
43     _ expression: () throws -> Void) -> Error? {
44     do {
45         try expression()
46         XCTFail("exception not thrown", file:file, line:line)
47     } catch {
48         return error
49     }
50     return nil
51 }

```

# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Account Creation WITH EMAIL + Empty PASSWORD

```
56  /**
57  *  Accounts can't be created with an empty password
58  *
59  *  Given:
60  *    - any email that has not been registered on the server
61  *    - an empty password
62  *
63  *  When I create an account
64  *
65  *  Then:
66  *    - an HTTP 400 error is raised
67  *    - the response contains an error message for the `password` field
68  */
69  func testRegisterFails_EmptyPassword() {
70      let email = "\(UUID().uuidString)@example.com"
71      let password = ""
72      let error = XCTAssertThrow {
73          try self.api.create(account: .email(email: email, password: password)).get()
74      }
75      expect(error).to(beHTTP(status: 400))
76      expect(error).to(containHTTPError(field: "password"))
77  }
```



# AUTOMATED TESTING FROM THE TRENCHES

## GiveAway: Custom Matchers



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

```
/**
 * Matcher function that will match for specific HTTP status.
 *
 * Example:
 * ...
 * expect(error).to(beHTTP(status: 400))
 *
 */
func beHTTP<T>(status: Int) -> MatcherFunc<T?> {
  return MatcherFunc { actualExpression, failureMessage throws -> Bool in
    failureMessage.postfixMessage = "have HTTP status <\(status)>"
    if let actualValue = try actualExpression.evaluate(),
       let httpError = actualValue as? HTTPError {
      return httpError.status == status
    }
    return false
  }
}
```



**OLIVIER TABONNE**  
CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Account Creation WITH EMAIL Already Used

- Given any email that already exists on the server, given a valid password
- When I Create Account (POST /account/create with user & password params)
- Then I get an error (HTTP 400) indicating the login is invalid



# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Account Creation WITH EMAIL Already Used

```
79  /**
80  *  Accounts can't be created if email already exists for an other account
81  *
82  *  Given:
83  *    - any email that has already been registered
84  *    - a valid password
85  *
86  *  When I create an account
87  *
88  *  Then:
89  *    - an HTTP 400 error is raised
90  *    - the response contains an error message for the `login` field
91  */
92  func testRegisterFails_DuplicateEmail() {
93      XCTAssertNoThrow {
94          let email = try anyEmailAlreadyRegistered()
95          let password = "toto"
96          let error = XCTAssertThrow {
97              try self.api.create(account: .email(email: email, password: password)).get()
98          }
99          expect(error).to(beHTTP(status: 400))
100         expect(error).to(containHTTPError(field: "login"))
101     }
102 }
103
```



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE



# GIVEAWAY : SYNCHRONOUS EVALUATION OF A PROMISE

```
4 extension Promise {
5
6     /**
7      * poll the runloop until the promise is resolved.
8      *
9      * - throws: promise's error (if any)
10     * - returns: promise's result
11     */
12     func get() throws -> T {
13         let rl = RunLoop.current
14
15         var result: Result<T>? = nil
16
17         self.then {
18             result = .result($0)
19         }.catch {
20             result = .error($0)
21         }
22         while result == nil {
23             rl.run(until: Date().addingTimeInterval(0.05))
24         }
25         switch result! {
26             case let .error(error): throw error
27             case let .result(r): return r
28         }
29     }
30 }
```





**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---



# AUTOMATED TESTING FROM THE TRENCHES

Same Apply To

- Login (aka obtain access token)
- Purchases (incl. Credit Card tokenization)
- **Consume REST resources**
- Unit tests,
- UI Tests



**OLIVIER TABONNE**

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

---

# AUTOMATED TESTING FROM THE TRENCHES

## Listing nearby auto wash

- Given any `document`
- When I request at most 5 auto wash near paris
- Then I get 5 results



# AUTOMATED TESTING FROM THE TRENCHES



**OLIVIER TABONNE**  
CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Listing nearby auto wash

```
255  /**
256  * Returns at most 5 autowash in a region
257  *
258  * Given:
259  *   - any document
260  *   - a region near paris
261  *
262  * When I list autowash in this region
263  *
264  * Then I get 5 results
265  */
266  func testNearParis() {
267      XCTAssertNoThrow {
268          let doc = try self.anyDocument()
269          let region = MKCoordinateRegionMakeWithDistance(Locations.Paris, 10_000, 10_000)
270          let instances = try doc.carWash.listAutoWash(in: region, limit: 5).get()
271          expect(instances.count) == 5
272      }
273  }
```

# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Managing Documents

```
46  /**
47  *
48  * - parameter account: Account for which document should be created.
49  * If nil, try to find a suitable account. Create an anonymous account on last resort.
50  *
51  * - returns: an opened document for given account.
52  */
53  func anyDocument(_ account: RMAccount? = nil) throws -> MVMCDocument {
54      let doc = MVMCDocument(account: try account ?? self.anyAccount())
55      deferOnTearDown {
56          _ = try doc.close().get()
57      }
58      return try doc.open().then {
59          return doc
60      }.get()
61  }
```



# AUTOMATED TESTING FROM THE TRENCHES



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

## Listing nearby auto wash

```
255  /**
256  * Returns at most 5 autowash in a region
257  *
258  * Given:
259  *   - any document
260  *   - a region near paris
261  *
262  * When I list autowash in this region
263  *
264  * Then I get 5 results
265  */
266  func testNearParis() {
267    XCTAssertNoThrow {
268      let doc = try self.anyDocument()
269      let region = MKCoordinateRegionMakeWithDistance(Locations.Paris, 10_000, 10_000)
270      let instances = try doc.carWash.listAutoWash(in: region, limit: 5).get()
271      expect(instances.count) == 5
272    }
273  }
```

# AUTOMATED TESTING FROM THE TRENCHES

## Using local fixtures for unit tests



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

```
{  
  "model" : "mvc.ecommerce.order",  
  "pk" : 1,  
  "fields" : {  
    "amount" : "4.000",  
    "creation_time" : "2017-03-02T15:55:58.686088Z",  
    "offer_type" : "autowash\//kleen",  
    "total_amount" : "4.000",  
    "vat_amount" : "0.000",  
    "currency" : "EUR",  
    "state" : "processing",  
    "payment_3ds" : {  
      "requires_verification": true,  
      "url": "http://example.com"  
    },  
    "offer_url": "https:\\//mvc-api-dev.ripplemotion.fr/api/1.0/carwash/offers/1234/"  
  },  
}
```

### Attributes

Attribute	Type
N amount	Decimal
D creationDate	Date
S currency	String
T ddds	Transformable
S deliveryURL	String
T error	Transformable
N identifier	Integer 64
S offerType	String
S offerURL	String
S state	String
N totalAmount	Decimal
N vatAmount	Decimal





# AUTOMATED TESTING FROM THE TRENCHES

## Using local fixtures for unit tests



OLIVIER TABONNE

CO FONDATEUR /  
DIRECTEUR TECHNIQUE

```
func testMapping_Order_1_3DS() {
    XCTAssertNoThrow {
        let order = try self.pipeline
            .add(stage: Stage { $0.first as! Order}) //swiftlint:disable:this force_cast
            .process(input: "Order-1-3DS.json")
        expect(order.amount) == NSDecimalNumber(string: "4.000")
        expect(order.offerType) == Order.OfferType.autoWashByKleen
        expect(order.state) == Order.State.processing
        expect(order.currency) == "EUR"
        expect(order.vatAmount) == NSDecimalNumber.zero
        expect(order.totalAmount) == NSDecimalNumber(string: "4.000")
        expect(order.creationDate) == RMISO8601DateFormatter().date(from: "2017-03-02T15:55:58.686088Z")
        expect(order.ddds?.url.absoluteString) == "http://example.com"
        expect(order.ddds?.requiresVerification) == true
    }
}
```

# WRAP UP



**MERCI DE VOTRE PARTICIPATION !**

---





ripple motion

Créateur d'applications mobiles

