

Tester l'app iPhone

capitaine train 🚂

Opérateur 

11:02



Aide

Trajets à venir

Trajets effectués

Octobre 2013

Dimanche 06 octobre

14:10 Digoin

17:57 Paris Gare-de-Bercy



Pierre

Samedi 05 octobre

07:53 Paris Gare-de-Lyon

10:48 Digoin



Pierre

Juillet 2013



Recherche



Panier



Billets

Opérateur 

11:02



 Billets Digoin → Paris

Pierre

Dimanche 06 octobre

35,70 €

14:10 Digoin

14:55 Moulins-sur-Allier

 2^{nde}



TER 16848 Placement libre

36 minutes de correspondance

15:31 Moulins-sur-Allier

17:57 Paris Gare-de-Bercy

 2^{nde}



IC 5974 Voiture 14, place 85

Billet classique QLLJZT accepté comme pièce comptable.

Architecture

Vues

View controllers

Description du modèle

Code métier

Réseau

Sérialisation

Modèle de données

Trois cibles

- Tests Logiques
- Tests UI
- Tests d'Intégration

Tests Logiques

- Ne dépendent pas de l'UI
- Potentiellement asynchrones
- Rapides

Code métier

Modèle de données

Sérialisation

Réseau

SenTestingKit

```
- (void) testDirectionWithSelectedOutwardFolder
{
    CATSearch *search = [CATSearch insertInManagedObjectContext:self.moc];
    search.selectedOutwardFolder = [CATFolder insertInManagedObjectContext:self.moc];
    search.selectedInwardFolder = nil;

    STAssertEqualObjects([search direction], CATFolderDirection.inward, nil);
}
```


OCMock

```
- (void) testRefreshSessionSendsCorrectRequest
{
    // Given
    id httpClientMock = [OCMockObject mockForClass:[CATHTTPClient class]];
    [[httpClientMock expect] getSessionWithCompletion:OCMOCK_ANY];
    _profile.httpClient = httpClientMock;
    // When
    [_profile refreshSession];
    // Then
    STAssertNoThrow([httpClientMock verify], @"A request should have been created");
    STAssertTrue(_profile.refreshingSession, @"The loading flag should be set");
}
```


RunLoop

```
- (void) testCancelledRequestCallsFailureCallback
{
    // Given
    __block BOOL requestDidFail = NO;
    [self.httpClient getPath:@"foo"
                      parameters:nil
                      completion:^(NSHTTPURLResponse *response, id receivedObject, NSError *error)
    {
        requestDidFail = (error != nil);
    }];
    // When
    [[self.httpClient operationQueue] cancelAllOperations];
    [self waitForCompletion:&requestDidFail];
    // Then
    STAssertTrue(requestDidFail,
                  @"Cancelled requests should call their completion handler with an error");
}
```


Tests UI

- Dépendents de l'UI
- Injectés dans l'application
- Rapides

Description du modèle

Combo

```
- (void) testSettingInitialHTMLFormSubmitsForm
{
    UIWebView *webView = [_challengeView webView];
    id webViewMock = [OCMockObject partialMockForObject:webView];
    [[webViewMock expect] submitFirstForm];

    _challengeView.initialHTMLForm = @"<form></form>";

    [self waitUntil:webViewMock finishes:@selector(hasAnyExpectations)];
    STAssertNoThrow([webViewMock verify],
                    @"Setting the initialHTMLForm should submit the form");
}
```


Tests d'Intégration

- Dépendents de l'UI
- Scénarios complets
- Lents

View controllers

Vues

CTT

```
- (void) testDeletePassenger
{
    // Show the Passengers list
    [CTT tap:@"Passengers"];
    // Show the Passenger details
    [CTT tap:@"More info, Pierre"];

    // Server Response
    self.responseStatusCode = 204;
    self.responseObject = nil;

    // Delete the Passenger
    [CTT tap:@"Remove the Passenger"]; // Button
    [CTT tap:@"Remove the Passenger"]; // Action Sheet
    // Check the Passenger has been deleted
    [CTT nomatch:@"Pierre"];
}
```


Limitations

- Pas d'estimation de la couverture
- Tests d'Intégration sensibles à l'environnement
- Vulnérable aux changements d'API

