UNIT TESTING

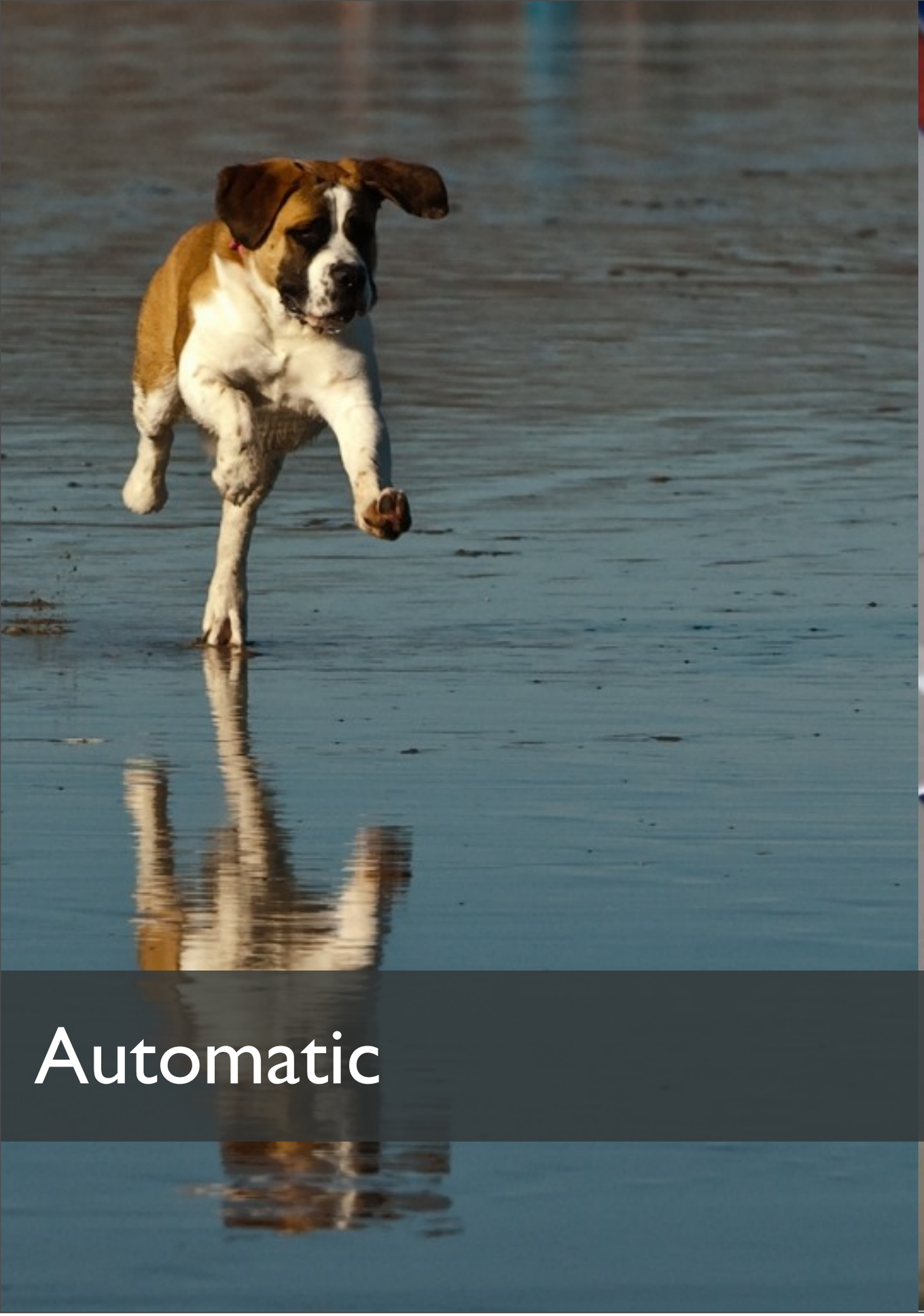
Cocoa Heads Stockholm 2011-03-07

“Unit tests are performed to prove that a piece of code does what the developer thinks it should do.”

Hunt &Thomas, Pragmatic Unit Testing

```
- (void)testIndexOfLargest_LargestElementFirst_ReturnsIndex0 {
    ArrayUtils *arrayUtils =
    [[[ArrayUtils alloc] init] autorelease];
    NSArray *array = [NSArray arrayWithObjects:
                      [NSNumber numberWithInt:9],
                      [NSNumber numberWithInt:8],
                      [NSNumber numberWithInt:7], nil];
    int indexOfLargest = [arrayUtils indexOfLargest:array];
    STAssertEquals(0, indexOfLargest,
                   @"Largest element should be at index 0");
}
```

Good Unit Tests



Automatic

Unit Testing Framework - OCUnit

Thorough

Repeatable

Fakes: Mocks and Stubs

Dependency Injection

Independent

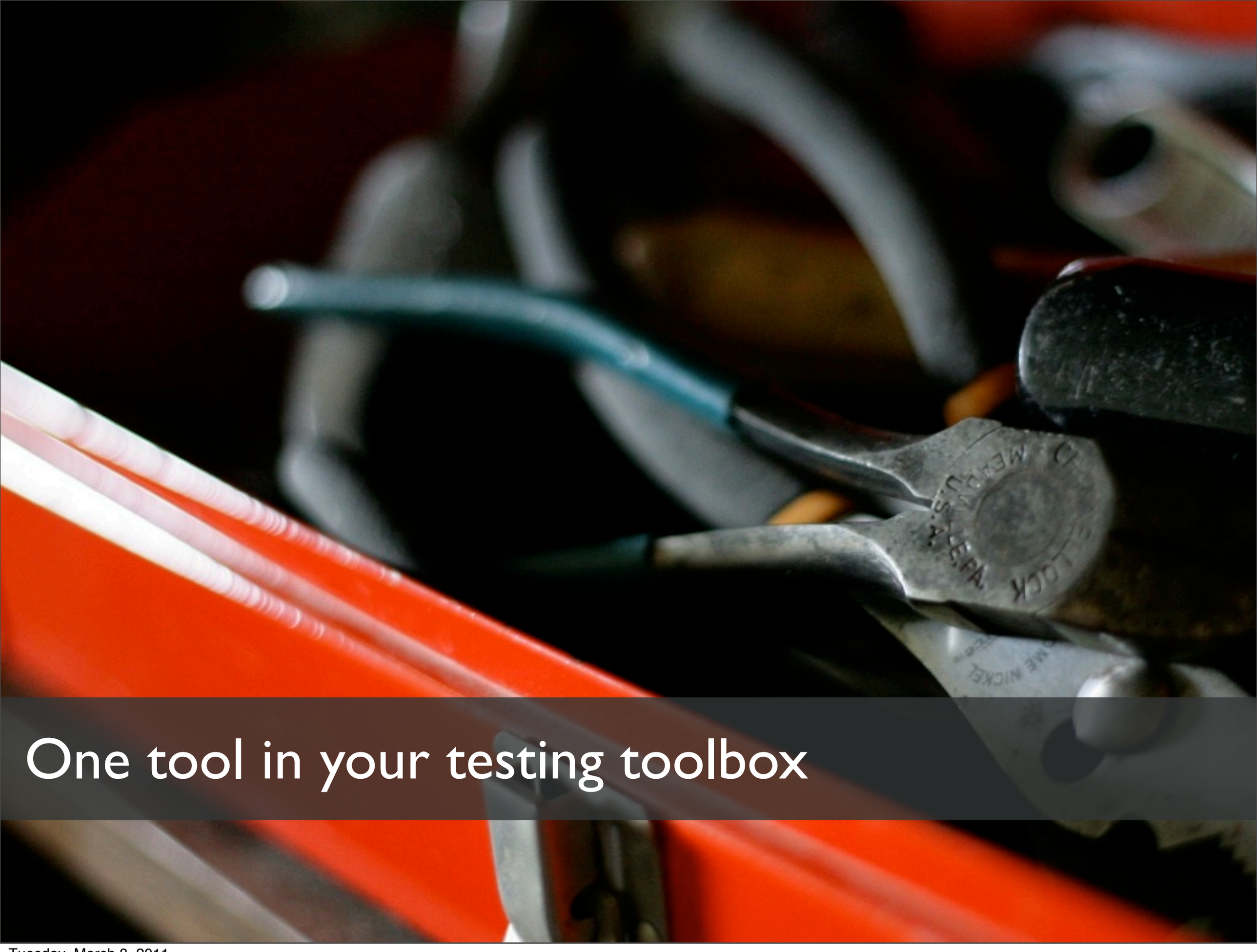
Professional

GOOD UNIT TEST

- Automatic
- Thorough
- Repeatable
- Independent
- Professional

GOOD UNIT TEST

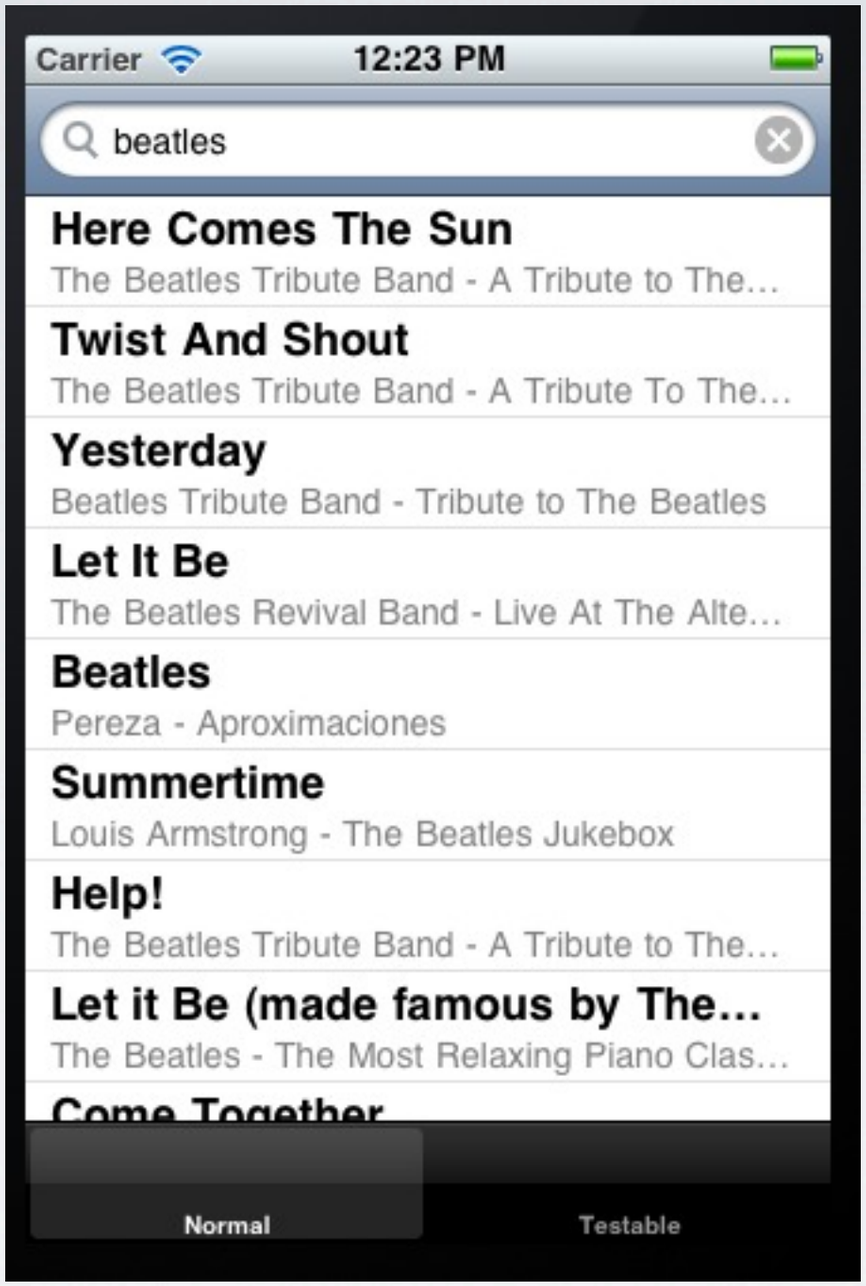
- **A**utomatic
- **T**horough
- **R**epeatable
- **I**ndependent
- **P**rofessional



One tool in your testing toolbox

DESIGN FOR TESTABILITY

- Making the assert
- A Unit Test is an API user
- TDD


Carrier
12:23 PM
beatles
Here Comes The Sun
The Beatles Tribute Band - A Tribute to The…
Twist And Shout
The Beatles Tribute Band - A Tribute To The…
Yesterday
Beatles Tribute Band - Tribute to The Beatles
Let It Be
The Beatles Revival Band - Live At The Alte…
Beatles
Pereza - Aproximaciones
Summertime
Louis Armstrong - The Beatles Jukebox
Help!
The Beatles Tribute Band - A Tribute to The…
Let it Be (made famous by The…
The Beatles - The Most Relaxing Piano Clas…
Come Together
Normal
Testable

Example

```
- (void)searchBarSearchButtonClicked:(UISearchBar *)searchBar {
  ...
}
```

```
- (void)configureCell:(UITableViewCell *)cell
          atIndexPath:(NSIndexPath *)indexPath {
  ...
}
```

```
- (void)searchBarSearchButtonClicked:(UISearchBar *)searchBar {
    NSString *urlString =
    [NSString stringWithFormat:@"http://ws.spotify.com/search/1/track.json?q=%@",
     [searchBar.text stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]];
    NSURL *url = [NSURL URLWithString:urlString];
    __block ASIHTTPRequest *request = [ASIHTTPRequest requestWithURL:url];
    __block __typeof__(self) blockSelf = self;
    [request setCompletionBlock:^ {
        NSDictionary *dict = [[CJSONDeserializer deserializer]
                              deserializeAsDictionary:request.responseData
                              error:nil];
        blockSelf.tracks = [dict objectForKey:@"tracks"];
        dispatch_async(dispatch_get_main_queue(), ^{
            [blockSelf.tableView reloadData];
        });
    }];
    [request startAsynchronous];
    [searchBar resignFirstResponder];
}
```

UITableViewController

```
- (void)searchBarSearchButtonClicked:(UISearchBar *)searchBar {
    [self.trackRequest searchForTracks:searchBar.text];
    [searchBar resignFirstResponder];
}

- (void)trackRequest:(SpotifyTrackRequest *)trackRequest
         foundTracks:(NSArray *)tracks {
    self.tracks = tracks;
    [self.tableView reloadData];
}
```

UITableViewController

```
- (void)configureCell:(UITableViewCell *)cell
          atIndexPath:(NSIndexPath *)indexPath {
    NSDictionary *track = [self.tracks objectAtIndex:indexPath.row];
    cell.textLabel.text = [track objectForKey:@"name"];

    NSDictionary *album = [track objectForKey:@"album"];
    NSString *albumName = [album objectForKey:@"name"];

    NSArray *artists = [track objectForKey:@"artists"];
    NSDictionary *artist = [artists objectAtIndex:0];
    NSString *artistName = [artist objectForKey:@"name"];
    cell.detailTextLabel.text = [NSString stringWithFormat:@"%@ - %@",
                                 artistName, albumName];
}
```

UITableViewController

```
- (void)configureCell:(UITableViewCell *)cell
          atIndexPath:(NSIndexPath *)indexPath {
    SpotifyTrack *track = [self.tracks objectAtIndex:indexPath.row];
    cell.textLabel.text = track.title;
    cell.detailTextLabel.text = [NSString stringWithFormat:@"%@ - %@",
                                 track.artist, track.album];
}
```

UITableViewController

```
- (NSString *)URLStringWithSearchTerm:(NSString *)searchTerm {
    return [NSString stringWithFormat:@"http://ws.spotify.com/search/1/track.json?q=%@",
            [searchTerm stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding]];
}
```

SpotifyTrackURL

```objc
- (NSArray *)tracksFromJSONData:(NSData *)jsonData {
    NSDictionary *dict = [[CJSONDeserializer deserializer]
                          deserializeAsDictionary:jsonData
                          error:nil];
    NSArray *tracks = [dict objectForKey:SPOTIFY_TRACKS_KEY];
    NSMutableArray *spotifyTracks = [NSMutableArray array];
    for (NSDictionary *trackDict in tracks) {
        SpotifyTrack *spotifyTrack = [self trackFromDictionary:trackDict];
        [spotifyTracks addObject:spotifyTrack];
    }
    return spotifyTracks;
}
```

SpotifyTrackParser

```
- (SpotifyTrack *)trackFromDictionary:(NSDictionary *)trackDict {
    SpotifyTrack *track = [[SpotifyTrack alloc] init];
    track.title = [trackDict objectForKey:SPOTIFY_TRACK_NAME_KEY];

    NSDictionary *album = [trackDict objectForKey:SPOTIFY_TRACK_ALBUM_KEY];
    track.album = [album objectForKey:SPOTIFY_TRACK_NAME_KEY];

    NSArray *artists = [trackDict objectForKey:SPOTIFY_TRACK_ARTISTS_KEY];
    if (artists.count > 0) {
        NSDictionary *artist = [artists objectAtIndex:0];
        track.artist = [artist objectForKey:SPOTIFY_TRACK_NAME_KEY];
    }

    return [track autorelease];
}
```

SpotifyTrackParser

```
@property (nonatomic, copy) NSString *title;
@property (nonatomic, copy) NSString *artist;
@property (nonatomic, copy) NSString *album;
```

SpotifyTrack

```
- (void)searchForTracks:(NSString *)text {
    NSURL *url = [NSURL URLWithString:[self.trackURL URLStringWithSearchTerm:text]];
    ASIHTTPRequest *request = [ASIHTTPRequest requestWithURL:url];
    request.delegate = self;
    [request startAsynchronous];
}

- (void)requestFinished:(ASIHTTPRequest *)request {
    NSData *responseData = [request responseData];
    NSArray *tracks = [self.trackParser tracksFromJSONData:responseData];
    [self.delegate trackRequest:self foundTracks:tracks];
}
```

SpotifyTrackRequest

READING LIST

- Pragmatic Unit Testing, Andrew Hunt, David Thomas
- The Art of Unit Testing, Roy Osherove
- Clean Code, Robert C. Martin