

**Hatena**

# はてなにおける モダンiOSアプリ開発入門

Hatena Engineer Seminar #2



cockscomb

# cockscomb

株式会社はてな

アプリケーションエンジニア

はてなブログチーム/はてなスペースチーム

はてなサマーインターン2013参加後すぐに入社

はてなのアプリ





朝の日課と、最近気にな... 2ヶ月前

朝に近所を散歩するのが好きです。私の家の近くにはそこそこ大きな公園があって、そこでしば...



面白い記事を見つけた！ 3ヶ月前

第43回入賞作品 中学校の部 文部科学大臣奨励賞 -38°Cの日は暑いのに38°Cの風呂に入ると熱くないのはなぜか- これはすごい！ 一度出し...

サンプルエントリ 4ヶ月前

はてなブログでは、沢山の方々がそれぞれのスタイルで日記（ブログ）を更新しています。日常の...



1日目 Perl事前課題のフ... 4ヶ月前

はてなで主に利用されているプログラミング言語「Perl」の基本的な使い方を確認するために、事前課題が用意されています。課題では...





朝の日課と、最近気になる...

1時間前

朝に近所を散歩するのが好きです。  
私の家の近くにはそこそこ大きな...



面白い記事を見つけた！

1時間前

第43回入賞作品 中学校の部 文部科学大臣奨励賞  
-38°Cの日は暑いのに38°Cの風呂に入...

はてなブログ公式iPhoneア...

2ヶ月前

はてなブログにiPhoneアプリが登場しまし  
た。外出しているときも、近くにパソコン...

はてなブログへようこそ

7ヶ月前

自分の考えをまとめて発表するに  
は、書きやすく、編集しやすいブ...

★★★★



マイブログ



購読中のブログ



お知らせ

# はてなブックマーク iOS SDK



# アジェンダ

1. はてなにおけるアプリ開発
2. はてなサマーインターン2013
3. はてなのiOS開発を支える技術

# はてなにおける アプリ開発

# 開発

HTML5

Titanium

RubyMotion

ネイティブ + WebView

アプリの方向性を決定

ペーパープロトタイプリング

ユーザーテスト

社内のiOS端末ユーザーにTestFlight

アプリリリース

ユーザーの同意を得てトラッキング

# はてなブログアプリ

しっかりとブログを書けること

書きやすいこと

どこにいても書けること

フィードバックが得られること



マイブログ  
 cockscomb's blog >  
 scomb life >

マイブログ cockscomb's blog  
 あたらしいエントリー  
 記事の管理 >  
 下書き一覧 >  
 コメント >  
 設定 >

記事の管理  
 下書き  
 下書きの日記 100日前  
 日記は3週間ごとに、毎日朝起きて30分おき  
 公開中  
 今更かな日記 30日前  
 朝の日記をBGMで書いた。yuta  
 今更かな日記 15日前  
 映画のDVDをみた。絵の怪物のこころ  
 iPad  
 Apple  
 E... 3...のアプリをインストール  
 今日の記事 25日前  
 今日の記事はアップロードした。

下書き  
 下書きの日記 15日前  
 下書きマニアアアア。

コメント  
 承認済  
 vivreo  
 今更かな日記のコメント 10日前  
 おもしろくた。 >  
 承認済  
 nijunkun  
 今更かな日記のコメント 12日前  
 コメントは面白かった >

タイトル [RT]  
 [戻る] [進む] [検索] [共有]

記事を確認はるか  
 いいえ はい

投稿中...

下書き保存はした  
 開く 記事を書く 公開  
 タイトル:  
 本文を入力  
 カテゴリー:  
 投稿先: cockscomb's blog >  
 編集 / フォント

cockscomb's blog  
 scomb life

写真を撮る...  
 ライブラリから選択  
 キャンセル

保存しない  
 下書きを保存する  
 キャンセル

リンク先のURL  
 キャンセル OK

[チェックマーク]

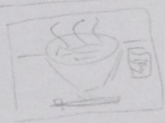
予期せぬエラーが発生しました  
 エラー5136  
 [X]

本文 [RT]  
 マイブログ: 検索 [戻る] [共有]  

あ	か	さ	[X]
た	な	は	スース
ま	や	り	実行
わ	れ	い	!!

[戻る] [進む] [検索] [共有]

アップロード中...

日記 cockscomb's blog  
 2016-08-29  
 E...の日記  
  
 ラーメンを食べた。

カテゴリ [RT]  
 日記  
 映画  
 音楽  
 プログラム  
 Apple  
 新規追加

親親カテゴリ名  
 キャンセル OK

Myコレクション [RT]  
 B! 8 @  
 お気に入り (4)  
 Hink: kate @cockscomb  
 まいれ3サイト

写真 [RT]  
 カラコロ-ル (192) >  
 フォトストリーム (138) >  
 アルバム (73) >





cell. KDDI 3G 14:35

戻る エントリー詳細 公開

タイトル: エントリーのタイトル

ブログ: chris4403's diary

編集 プレビュー

# 計測

<input type="checkbox"/>	イベントアクション ?
<input type="checkbox"/>	1. 見出し
<input type="checkbox"/>	2. 太字
<input type="checkbox"/>	3. リスト
<input type="checkbox"/>	4. 引用
<input type="checkbox"/>	5. 斜体
<input type="checkbox"/>	6. 数字リスト





# Web APIを利用するiOSアプリ開発

はてなサマーインターンシップ2013

# はてなインターン2013

はてなサマーインターン2013  
レポートサイト

はてな教科書

iOS講義サンプルコード



<https://github.com/hatena/>

“iOS 側ざっと見たけど、こういう  
綺麗なコードが小さいサンプルでなく  
一式揃っているのは本当に貴重だし  
有り難すぎる。”

*-id:murakaming*

“それなりに実戦経験がある人でも  
やってなさそうだったり知らなそうなことを  
サラッと見せてくれている”

*-id:griffin-stewie*

# 実践的なサンプルコード

*CocoaPods*

*AFNetworking*

*UIStoryboard*

*Auto Layout*

*NSLayoutConstraint*

*Key Value Observation*

*blocks*

*UITableViewController*

*NSNotificationCenter*

*isEqual: overriding*

*appledoc*

...

はてなのiOS開発を  
支える技術



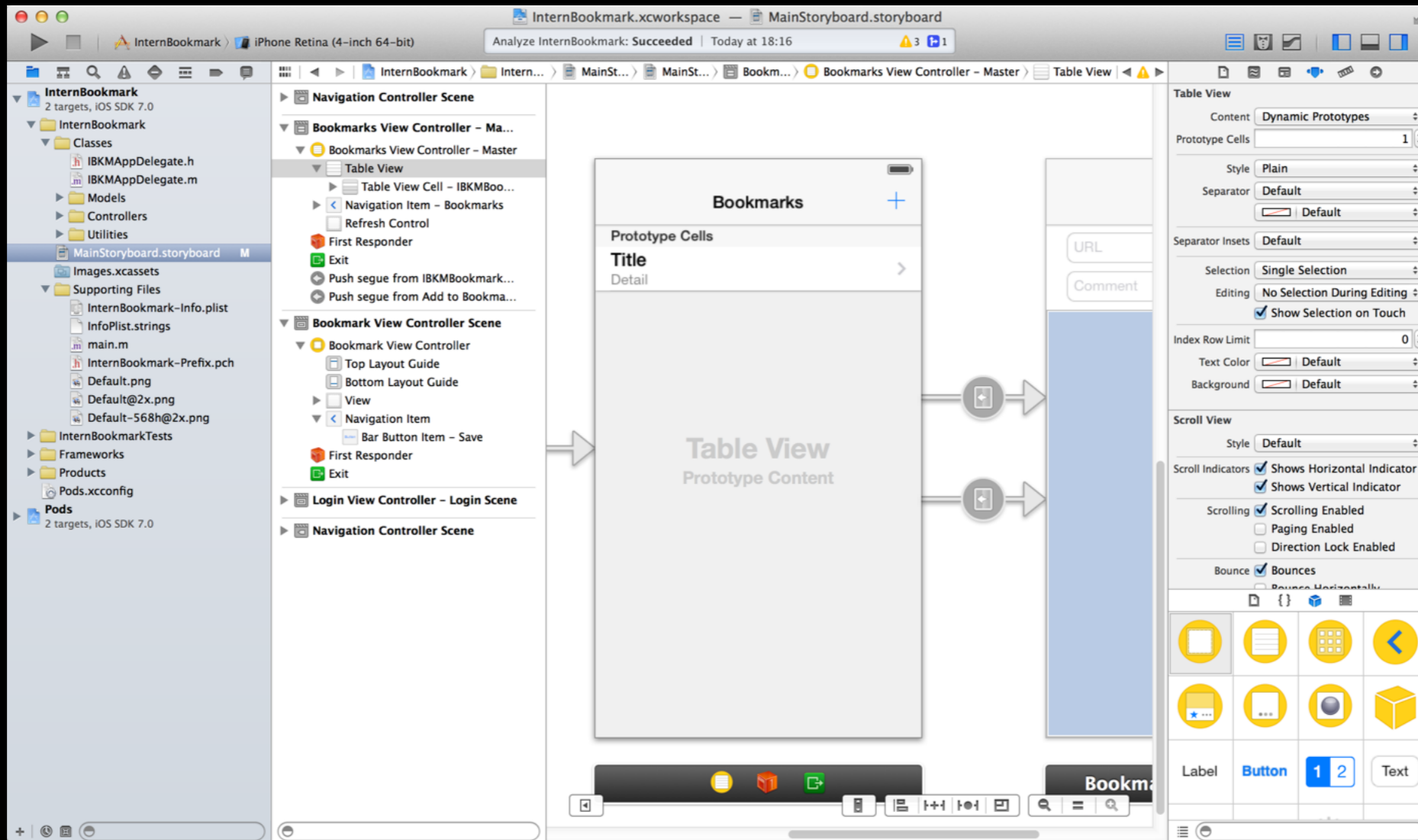
# CocoaPods

```
platform :ios, '7.0'
```

```
pod 'AFNetworking'
```

```
pod install
```

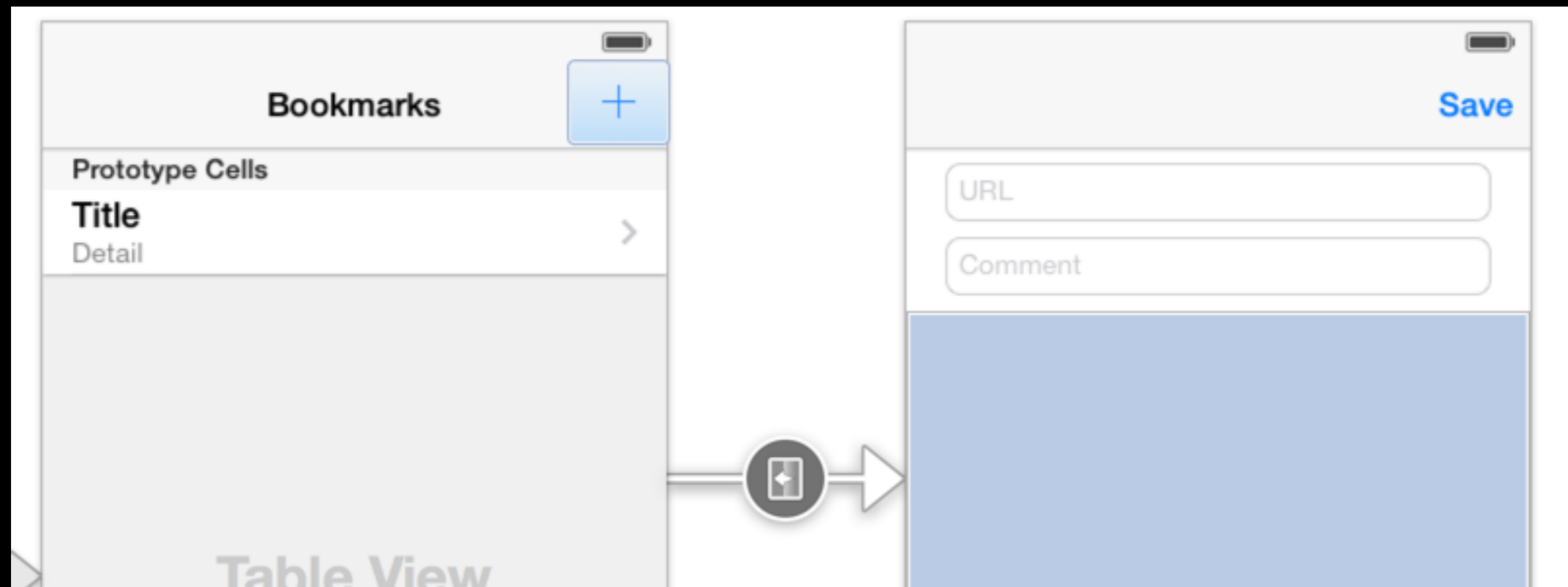
# UIStoryboard



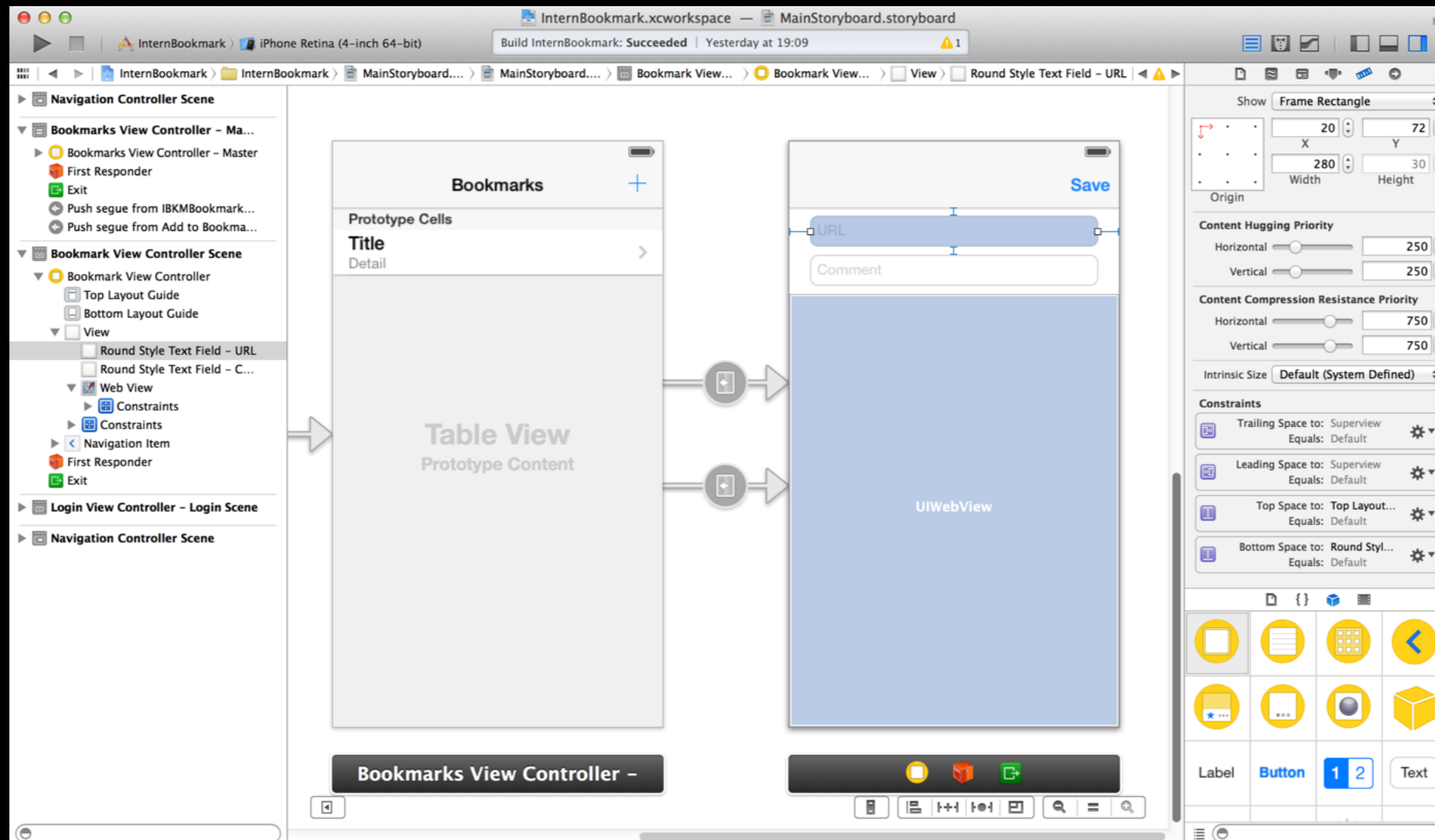
# Before

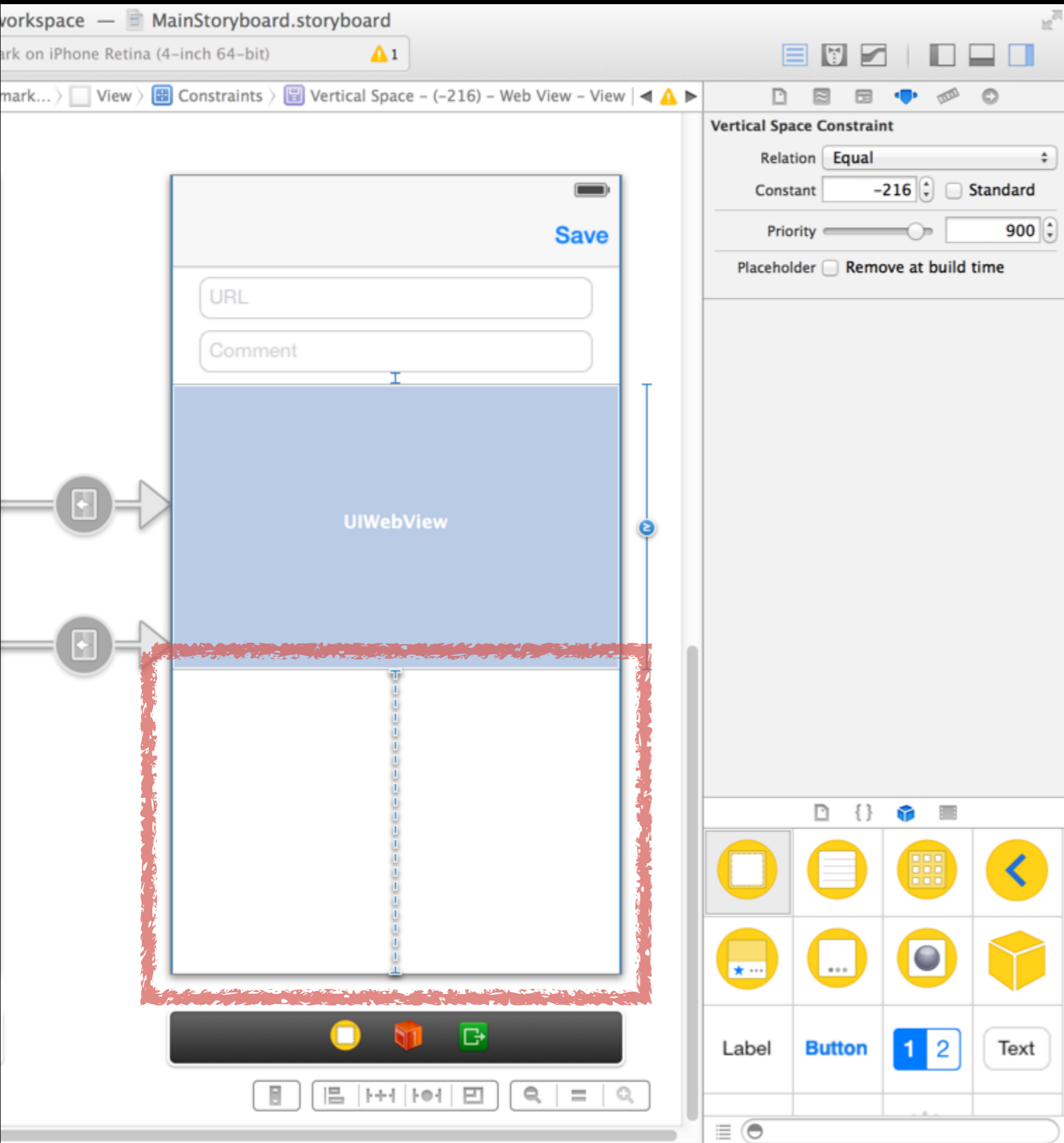
```
- (IBAction)newBookmark:(id)sender
{
    IBKMBookmarkViewController *bookmarkViewController
    = [[IBKMBookmarkViewController alloc] init];
    [self.navigationController
     pushViewController:bookmarkViewController
     animated:YES];
}
```

# After



# Auto Layout



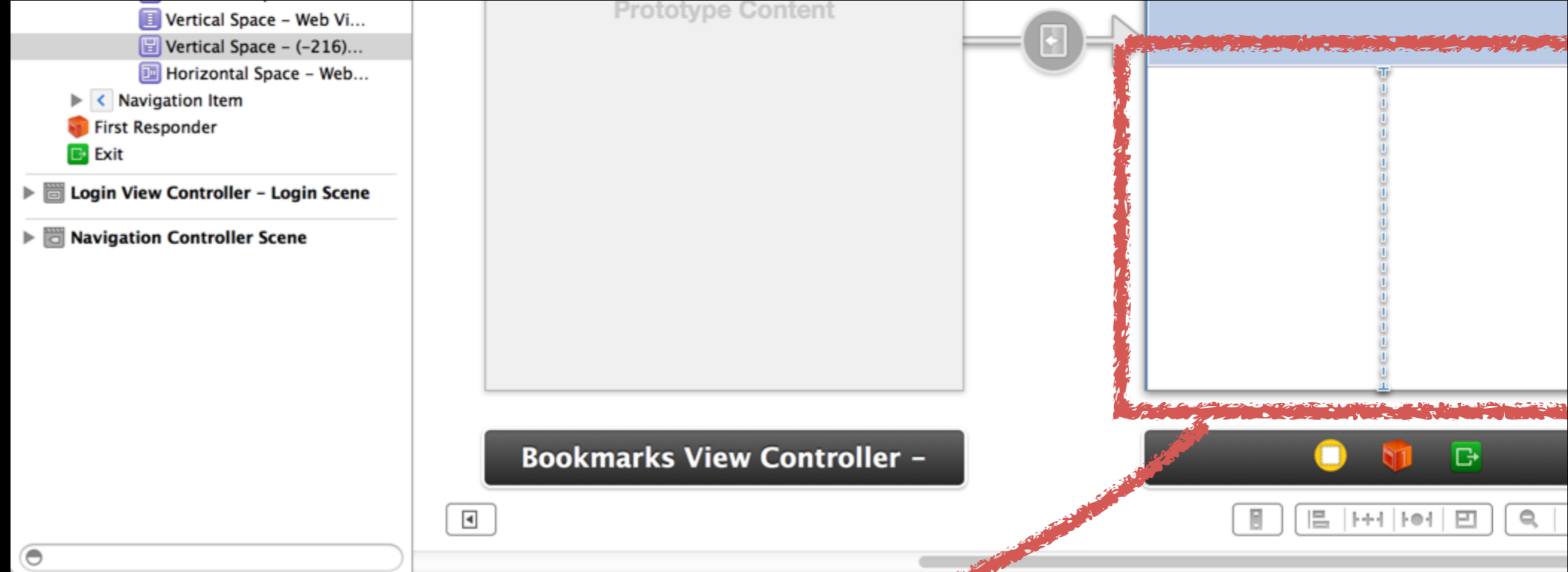


# Before

```
@property (weak, nonatomic) IBOutlet UITextView *textView;
```

```
– (void)keyboardWillChangeFrame:(NSNotification *)notification  
{  
    CGRect keyboardRect =  
        [notification.userInfo[UIKeyboardFrameEndUserInfoKey]  
         CGRectValue];  
    keyboardRect =  
        [self.view convertRect:keyboardRect fromView:nil];  
    double animationDuration =  
        [notification.userInfo[UIKeyboardAnimationDurationUserInfoKey]  
         doubleValue];  
  
    CGFloat keyboardHeight =  
        self.view.bounds.size.height – keyboardRect.origin.y;  
  
    CGRect newRect = self.view.bounds;  
    newRect.size.height -= keyboardHeight;  
  
    [UIView animateWithDuration:animationDuration animations:^(  
        self.textView.frame = newRect;  
    )];  
}
```

# After



```
@property (weak, nonatomic) IBOutlet NSLayoutConstraint *keyboardHeight;
```

```
-(void)keyboardWillChangeFrame:(NSNotification *)notification  
{  
    ...  
    CGFloat keyboardHeight =  
        self.view.bounds.size.height - keyboardRect.origin.y;  
    self.keyboardHeight.constant = -keyboardHeight;  
    [UIView animateWithDuration:animationDuration animations:^(  
        [self.view layoutIfNeeded];  
    )];  
}
```

# Key Value Observation



# ブックマーク一覧

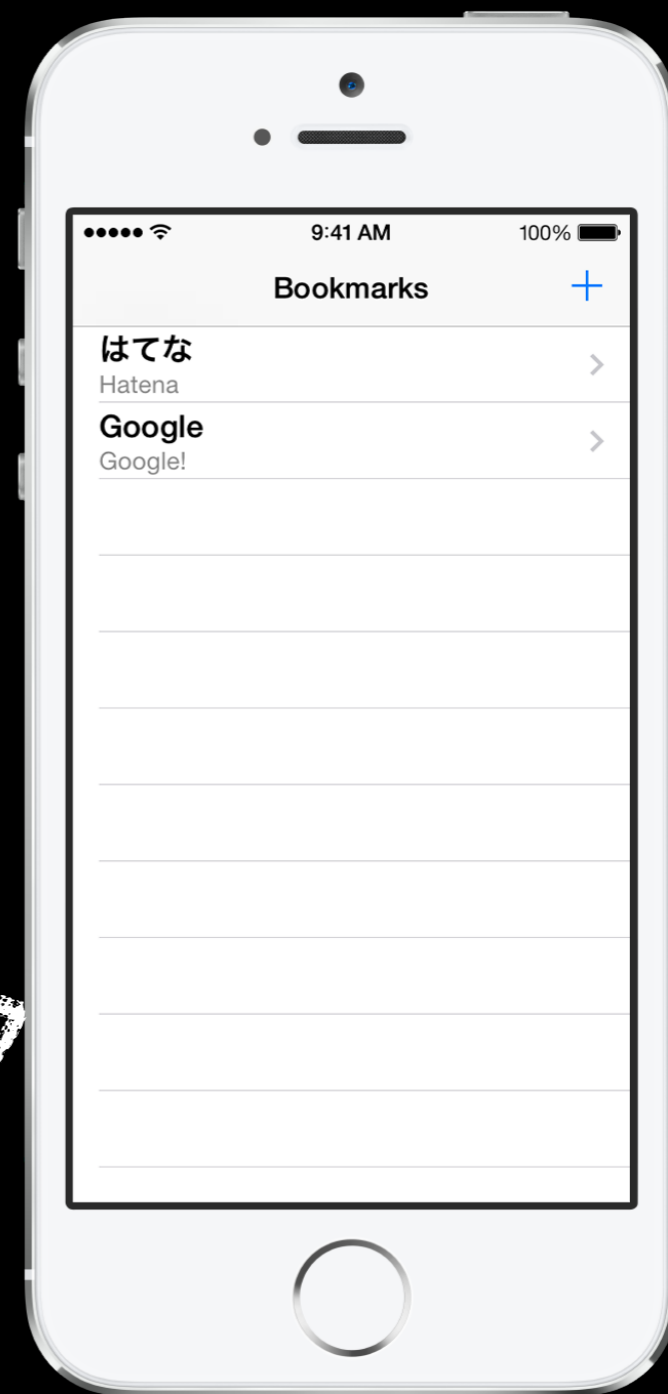
**BookmarkManager**  
– (NSArray \*)bookmarks

監視

通知

**TableViewController**

更新



# TableViewController

```
[[IBKMBookmarkManager sharedManager]
 reloadBookmarksWithBlock:^(NSError *error) {
    if (error) {
        NSLog(@"Error: %@", error);
    }
    [self.tableView reloadData];
}];
```

手で丁寧にUITableViewに反映させるのをやめたい

# TableViewController

```
[[IBKMBookmarkManager sharedManager]
 reloadBookmarksWithBlock:^(NSError *error) {
 if (error) {
 NSLog(@"Error: %@", error);
 }
 }];
```

# Key Value Observation

```
[ [IBKMBookmarkManager sharedManager]
  addObserver:self
  forKeyPath:@"bookmarks"
  options:NSKeyValueObservingOptionNew
  context:nil];
```

```

- (void)observeValueForKeyPath:(NSString *)keyPath
    ofObject:(id)object
    change:(NSDictionary *)change
    context:(void *)context
{
    if (object == [IBKMBookmarkManager sharedManager]
        && [keyPath isEqualToString:@"bookmarks"]) {
        NSIndexSet *indexSet = change[NSKeyValueChangeIndexesKey];
        NSKeyValueChange changeKind =
            (NSKeyValueChange)[change[NSKeyValueChangeKindKey] integerValue];

        NSMutableArray *indexPaths = [NSMutableArray array];
        [indexSet enumerateIndexesUsingBlock:^(NSUInteger index, BOOL *stop) {
            [indexPaths addObject:[NSIndexPath indexPathForRow:index inSection:0]];
        }];

        [self.tableView beginUpdates];
        if (changeKind == NSKeyValueChangeInsertion) {
            [self.tableView insertRowsAtIndexPaths:indexPaths
                withRowAnimation:UITableViewRowAnimationAutomatic];
        }
        else if (changeKind == NSKeyValueChangeRemoval) {
            [self.tableView deleteRowsAtIndexPaths:indexPaths
                withRowAnimation:UITableViewRowAnimationAutomatic];
        }
        else if (changeKind == NSKeyValueChangeReplacement) {
            [self.tableView reloadRowsAtIndexPaths:indexPaths
                withRowAnimation:UITableViewRowAnimationAutomatic];
        }
        [self.tableView endUpdates];
    }
}

```

# IBKMBookmarkManager

```
[[self mutableArrayValueForKey:@"bookmarks"]
    insertObjects:newBookmarks
    atIndexes:[NSSet
        indexSetWithIndexesInRange:
        NSRange(0, newBookmarks.count)]];
```

`self.bookmarks`を直接操作せず proxyを経由する

# ブックマーク一覧

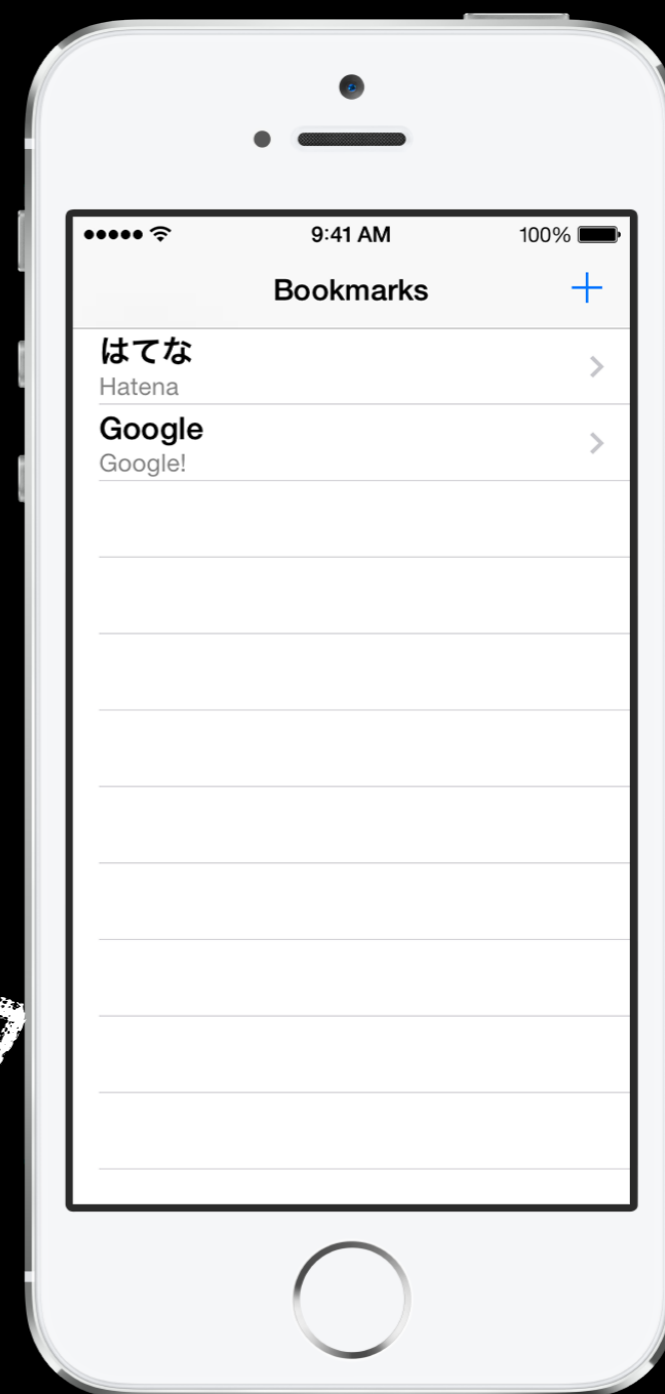
**BookmarkManager**  
– (NSArray) bookmarks

監視

通知

**TableViewController**

更新



テスト



# Test Frameworks

**OCUnit/XCTest**

**Kiwi**

BDD. テストを書きやすくするために使っている

**Nocilla**

ネットワークStub

**UIAutomation**

統合テストに使われている. KIFも使っていきたい

```
context(@"Fetching service data", ^{
    it(@"should receive data within one second", ^{
        __block NSData *fetchData = nil;

        NSURLRequest *request =
            [NSURLRequest requestWithURL:
             [NSURL URLWithString:@"http://www.hatena.ne.jp/"]];

        [NSURLConnection
         sendAsynchronousRequest:request
          queue:[NSOperationQueue mainQueue]
         completionHandler:^(NSURLResponse *response,
                              NSData *data,
                              NSError *connectionError)
         {
             fetchData = data;
         }
        ];

        [[expectFutureValue(fetchData) shouldEventually] beNonNil];
    });
});
```

```
stubRequest(@"GET", @"http://www.hatena.ne.jp/").  
andReturn(200);
```

# はてな

## スマートフォン重視

スマートフォンのトラフィックが無視できない

## 大きな裁量

企画から設計、スピード感ある開発

## 開発手法も日々変化

エンジニアの自主性でどんどん変えていける

**積極採用中**

<http://www.hatena.ne.jp/company/staff>

# Hatena

