MessageUI and Twitter Frameworks

Bonus Lecture 2
MessageUI Framework

- Provides standard composition interfaces for email messages and SMS text messages
- Allow the user to send email and SMS message without leaving your app
Mail Composition

- The class `MFMailComposeViewController` defines a view controller for writing and sending emails.

- Typically presented modally (but it is just a view controller - and you can present it however you like).
Can I send email?

- To check if the device is configured to send email, use 
  `[MFMailComposeViewController canSendMail]`

- If this returns **NO**, then *don't* try to display a mail view controller
Customizing the Mail Fields

- (void) setSubject: (NSString *) subject;
- (void) setToRecipients: (NSArray *) toRecipients;
- (void) setCcRecipients: (NSArray *) ccRecipients;
- (void) setBccRecipients: (NSArray *) bccRecipients;
- (void) setMessageBody: (NSString *) body
  isHTML: (BOOL) isHTML;
- (void) addAttachmentData: (NSData *) attachment
  mimeType: (NSString *) mimeType
  fileName: (NSString *) filename;
MKMailComposeViewControllerDelegate

- (void) mailComposeController: (MFMailComposeViewController *) controller
didFinishWithResult: (MFMailComposeResult) result
    error: (NSError *) error

enum MFMailComposeResult
{
    MFMailComposeResultCancelled,
    MFMailComposeResultSaved,
    MFMailComposeResultSent,
    MFMailComposeResultFailed
};
typedef enum MFMailComposeResult MFMailComposeResult;
SMS Composition

- The class `MFMessageComposeViewController` defines a view controller for writing and sending SMS messages
- Again, typically presented modally
Can I send SMS?

• To check if the device is capable of sending SMS messages,
  [MFMessageComposeViewController canSendText]

• If this returns NO, then don't try to display a message view controller
Customizing the Message

@property (nonatomic, copy) NSString *body;
@property (nonatomic, copy) NSArray *recipients;
MKMailComposeViewControllerDelegate

- (void) messageComposeViewController: (MFMessageComposeViewController *) controller
didFinishWithResult: (MessageComposeResult) result

enum MFMessageComposeResult
{
    MFMailComposeResultCancelled,
    MFMailComposeResultSent,
    MFMailComposeResultFailed
};
typedef enum MFMailComposeResult MFMailComposeResult;
Twitter Framework

• Two developer features
  • A framework for creating and sending Twitter API requests
  • The "tweet-sheet" for composing tweets
Twitter Requests

- The class **TWRequest** is used for making Twitter requests
- Very easy to use
- Basically a wrapper for the underlying HTTP requests
- Handles authentication
The Tweet Sheet

• The class `TWTweetComposeViewController` defines a view controller for writing and sending tweets

• Again, typically presented modally
Can I tweet?

• To check if the device is capable of sending tweets, `TWTweetComposeViewController canSendTweet`

• If this returns NO, then don't try to display a message view controller
Customizing the Tweet

- (BOOL) setInitialText: (NSString *) text;
- (BOOL) addImage: (UIImage *) image;
- (BOOL) addURL: (NSURL *) url;
- (BOOL) removeAllImages;
- (BOOL) removeAllURLs;

/* Note: Once the tweet sheet has been shown to the user, you can no longer customize the tweet! */
The Completion Handler

• In lieu of a delegate, the tweet sheet uses a completion handler to signal when the user has finished using the sheet

• The completion handler is a property that can be set on the sheet, and must be a block with the following signature:

```
typedef void (^TWTweetComposeViewControllerCompletionHandler)(TWTweetComposeViewControllerResult result);
```
Resources

- **MessageUI Framework Reference**

- **Twitter Framework Reference**