

Address Book & Messaging



Боян Лазов

Address Book



Address Book - intro

- База данни, съдържаща контакти на хора и друга лична информация
- Идеята е тази информация да бъде споделена и достъпна между приложенията
- Като интерфейс има два framework-а за работа с Address Book:
 - `AddressBook.framework` - дава функции за достъп до информацията в адресната книга
 - `AddressBookUI.framework` - user interface за показване на информацията

Address book - basics

Структура на информацията

- Address book обекти - дават достъп до базата; сочат към една и съща база
- Записи (records) - отговарят на хора или групи; като цяло са множество от свойства (properties)
- Свойства (properties) - 2 типа - single-value и multi-value. И двата типа имат име и една или повече стойности.

Пример:

first/last name - single-value свойства

телефонен номер - multi-value свойство

Address book - basics

Работа със свойства

- **single-value** свойства - просто get/set
 - `ABRecordSetValue(record, propName, value, &error);`
 - `ABRecordCopyValue(record, propName);`
 - **NB!** очаквано, резултата от `copyvalue` го притежаваме ние и трябва да бъде `release`-нат
- **multi-value** свойства - множество от (label, value, prop_id). Идентични `set/get` за всеки label
 - `ABMultiValueCopyLabelAtIndex/CopyValueAtIndex`
 - `ABMultiValueAddValueAndLabel`
 - `ABMultiValueReplaceValueAtIndex/ReplaceLabelAtIndex`
 - `ABMultiValueRemoveValueAndLabelAtIndex`

Address book - editing info

- **Добавяне/триене**
 - `ABAddressBookAddRecord/RemoveRecord`
- **Запазване на промените**
 - `ABAddressBookHasUnsavedChanges`
 - `ABAddressBookSave`
 - `ABAddressBookRevert`

Пример

```
ABRecordRef aRecord = ABPersonCreate(); CFErrorRef anError =
NULL; bool didSet; didSet = ABRecordSetValue(aRecord,
kABPersonFirstNameProperty,
                                CFSTR("Katie"), &anError); if (!
didSet) { /* Handle error here. */ } CFStringRef firstName;
firstName = ABRecordCopyValue
(aRecord, kABPersonFirstNamePr
operty);
```

Address book - user interaction

- Имаме стандартни view controller-и за показване и въвеждане/редактиране на данни:
 - `ABPeoplePickerNavigationController` - служи за избиране на запис на човек от адресната книга
 - `ABPersonViewController` - показва запис на човек и по избор позволява редактиране
 - `ABNewPersonViewController` - служи за създаване на нов запис
 - `ABUnknownPersonViewController` - служи за допълване на информацията на частичен запис, и по избор позволява добавянето му към адресната книга



Messaging



MessageUI.framework

Messaging - писане на email

- `MFMailComposeViewController`- стандартен интерфейс за писане на мейл
- Можем да настроим получателите, заглавието, тялото на съобщението и `attachment`-и, преди да го покажем:
 - - `setSubject:`
 - - `setToRecipients:`
 - - `setCcRecipients:`
 - - `setBccRecipients:`
 - - `setMessageBody:isHTML:`

Messaging - email - attachments

- Добавяме с помощта на метода - - (void)
addAttachmentData:(NSData*)*attachment* mimeType:
(NSString*)*mimeType* fileName:(NSString*)*filename*
- *attachment* са данните, най-често съдържанието на файл
- *mimeType* - MIME type (напр. `image/jpeg`); повече за тях на <http://www.iana.org/assignments/media-types/index.html>
- *filename* - предпочитано име, с което файла да бъдат асоциирани данните. **Трябва да бъде != nil**

Messaging - писане на SMS

- `MFMessageComposeViewController`- стандартен интерфейс за писане на SMS
- Подобно на пращането на email, можем да конфигурираме някои полета, преди да го покажем на потребителя:
 - `recipients` - получателите като телефонни номера
 - `body` - текст на съобщението
- Други настройки
 - `+ (BOOL) canSendText` - връща дали устройството може да праща текстови съобщения или не

Q&A



**Благодаря Ви за
вниманието!**