

Wap plus iPhone SDK

Оглавление

Содержимое пакета	3
Установка и настройка	4
Описание интерфейсов	5
<i>WPBannerRequestInfo</i>	5
<i>WPBannerVlew</i>	5
<i>WPBannerViewDelegate</i>	6
Примеры использования	7
1. Инициализация	7
2. Добавление баннера в приложение	7

Содержимое пакета

SDK

resources изображения, необходимые для отображения баннера

wp_banner_background.png

wp_banner_close.png

wp_banner_shild.png

src

WPBannerInfo.h хранение полной информации о баннере.

WPBannerInfo.m

WPBannerInfoLoader.h загрузка информации о баннере с сервера.

WPBannerInfoLoader.m

WPBannerInfoParser.h создание WPBannerInfo из полученного xml.

WPBannerInfoParser.m

WPBannerRequestInfo.h информация, необходимая для получения баннера.

WPBannerRequestInfo.m

WPBannerView.h отображение баннера

WPBannerView.m

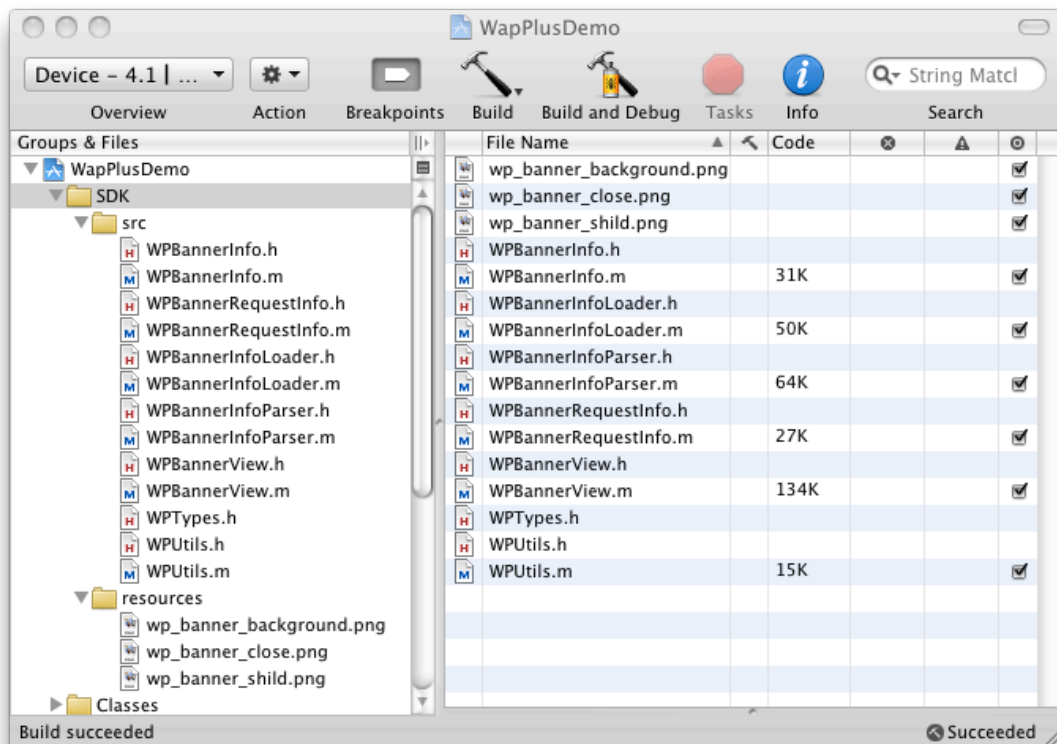
WPTypes.h типы, используемые в SDK.

WPUtils.h методы, необходимые для работы.

WPUtils.m

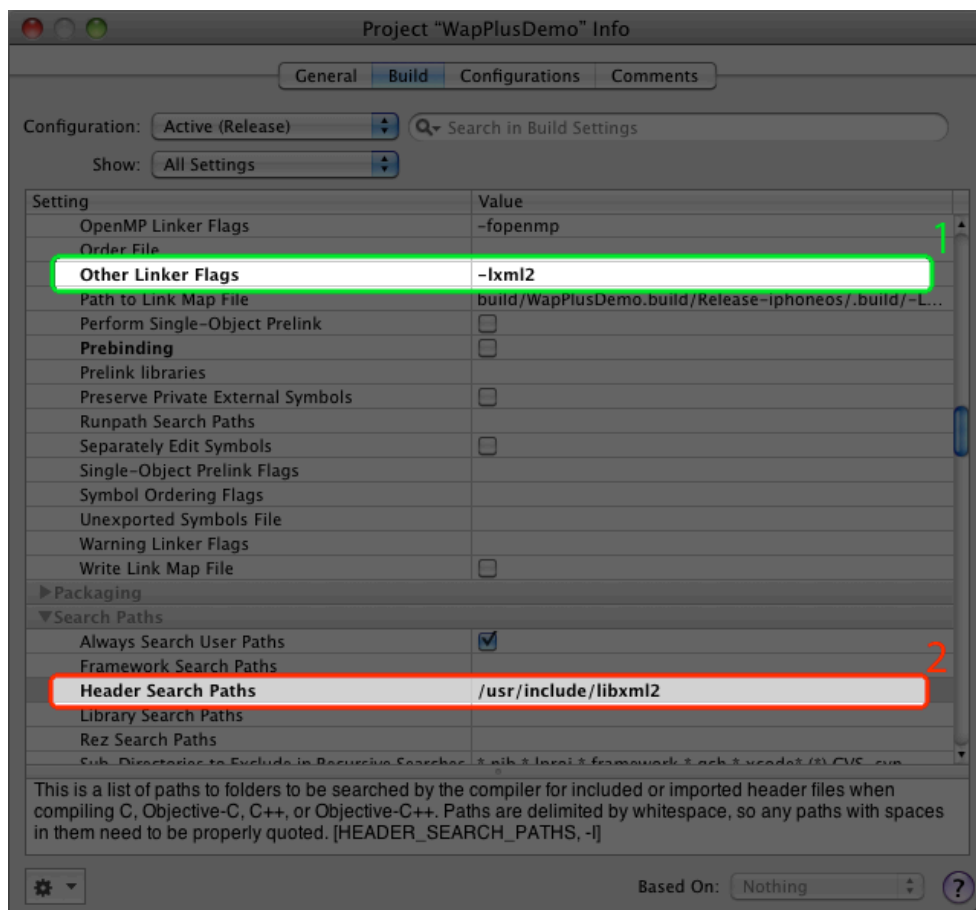
Установка и настройка

1. Необходимо скопировать все файлы, находящиеся в папках SDK/resources и SDK/src в свой проект.



2. Открыть «Project Info», перейти в раздел «Build» и добавить 2 параметра, необходимые для компиляции и работы приложения с этим SDK:

1. Добавить параметр «-lxml2» в «Other Linker Flags»
2. Добавить параметр «/usr/include/libxml2» в «Header Search Paths»



Описание интерфейсов

WPBannerRequestInfo

```
@interface WPBannerRequestInfo : NSObject

// ApplicationID – id вашего приложения
@property (nonatomic, assign) NSInteger applicationId;
// gender – пол текущего пользователя приложения (если известен)
@property (nonatomic, assign) WPGender gender;
// age – возраст текущего пользователя приложения (если известен)
@property (nonatomic, assign) NSInteger age;

@end
```

WPBannerView

```
@interface WPBannerView : UIView <WPBannerInfoLoaderDelegate>

// bannerInfo – информация о текущем баннере
@property (nonatomic, readonly) WPBannerInfo *bannerInfo;
// showCloseButton – показывать или нет кнопку закрытия баннера
@property (nonatomic, assign) BOOL showCloseButton;
// autoupdateTimeout – период авто обновления (в секундах). Если равно 0, то авто обновление баннера отключено.
@property (nonatomic, assign) CGFloat autoupdateTimeout;
// isMinimized – минимизирован баннер или нет
@property (nonatomic, assign) BOOL isMinimized;
// minimizedLabel – строка, которая отображается, когда баннер свернут (по умолчанию = "Открыть баннер")
@property (nonatomic, retain) NSString *minimizedLabel;
// hideWhenEmpty – показывать или нет баннер, когда информации с сервера нет
@property (nonatomic, assign) BOOL hideWhenEmpty;
// isEmpty – есть ли информация с сервера
@property (nonatomic, readonly) BOOL isEmpty;
// bannerHeight – высота баннера в данный момент.
@property (nonatomic, readonly) CGFloat bannerHeight;

// delegate – объект, который будет информироваться о текущем состоянии баннера (см. описание WPBannerViewDelegate)
@property (nonatomic, assign) id<WPBannerViewDelegate> delegate;

// Инициализация объекта
- (id) initWithBannerRequestInfo:(WPBannerRequestInfo *) requestInfo;

// Показать баннер сверху или снизу его superview
- (void) showFromTop:(BOOL) animated;
- (void) showFromBottom:(BOOL) animated;

// Скрыть баннер. Метод работает только в случае, если баннер находится сверху или снизу.
- (void) hide:(BOOL) animated;
```

```
// Получить заново информацию о баннере
- (void) reloadBanner;

// Минимизировать баннер
- (void) setIsMinimized:(BOOL)minimize animated:(BOOL) animated;

@end
```

WPBannerViewDelegate

```
@protocol WPBannerViewDelegate <NSObject>

// Вызывается при нажатии на баннер
- (void) bannerViewPressed:(WPBannerView *) bannerView;

@optional

// Вызывается при получении информации о баннере
- (void) bannerViewInfoLoaded:(WPBannerView *) bannerView;

// Вызывается после вызова метода hide: у WPBannerView
- (void) bannerViewDidHide:(WPBannerView *) bannerView;

// Вызывается, когда баннер сворачивается или разворачивается
- (void) bannerViewMinimizedStateChanged:(WPBannerView *) bannerView;

@end
```

Примеры использования

1. Инициализация

Для того, что бы инициализировать WPBannerView необходимо создать и сконфигурировать объект WPBannerRequestInfo.

```
@interface WPBannerRequestInfo : NSObject
{
    NSInteger    _applicationId;
    WPGender     _gender;
    NSInteger    _age;
}

@property (nonatomic, assign) NSInteger    applicationId;
@property (nonatomic, assign) WPGender     gender;
@property (nonatomic, assign) NSInteger    age;
@property (nonatomic, readonly) NSString *pageId;

- (id) initWithApplicationId:(NSInteger) applicationId;
- (id) initWithApplicationId:(NSInteger) applicationId gender:(WPGender)
gender;
- (id) initWithApplicationId:(NSInteger) applicationId age:(NSInteger) age;
- (id) initWithApplicationId:(NSInteger) applicationId gender:(WPGender) gender
age:(NSInteger) age;

@end
```

Он имеет 3 параметра:

- applicationId - обязательный параметр
- gender - не обязательный параметр, пол (WPGenderMale или WPGenderFemale)
- age - не обязательный параметр, возраст.

Не смотря на то, что параметры gender и age являются не обязательными, рекомендуется устанавливать их для более точного подбора отображаемого контента.

2. Добавление баннера в приложение

1. Необходимо добавить поддержку протокола WPBannerViewDelegate

```
#import <UIKit/UIKit.h>
#import "WPBannerView.h"

@interface UserViewController : UIViewController <WPBannerViewDelegate>
{
    WPBannerView *bannerView;
}

@end
```

2. Добавить создание WPBannerView в метод viewDidLoad

```
- (void) viewDidLoad
{
    [super viewDidLoad];

    WPBannerRequestInfo *requestInfo = [[WPBannerRequestInfo alloc]
initWithApplicationId:TEST_APPLICATION_ID];
```

```

        bannerView = [[WPBannerView alloc] initWithBannerRequestInfo:requestInfo];
        bannerView.showCloseButton = YES;
        bannerView.autoupdateTimeout = 10;
        bannerView.delegate = self;
        bannerView.hideWhenEmpty = YES;
        [bannerView reloadBanner];
        [self.view addSubview: bannerView];

        [requestInfo release];
    }

```

3. Имплементировать методы протокола WPBannerViewDelegate

```

- (void) bannerViewPressed:(WPBannerView *) bannerView
{
    if (bannerView.bannerInfo.responseType == WPBannerResponseWebSite)
        [[UIApplication sharedApplication] openURL:[NSURL
URLWithString:bannerView.bannerInfo.link]];

    [bannerView reloadBanner];
}

```