

Bound Breaker

Generated by Doxygen 1.8.9.1

Thu Apr 23 2015 11:27:15

Contents

1	Bound Breaker	1
2	Deprecated List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	9
4.1	Class List	9
5	File Index	11
5.1	File List	11
6	Class Documentation	15
6.1	AppDelegate Class Reference	15
6.1.1	Detailed Description	15
6.1.2	Property Documentation	15
6.1.2.1	window	15
6.2	AppDelegate() Category Reference	15
6.2.1	Detailed Description	15
6.3	BFApplink Class Reference	16
6.3.1	Detailed Description	16
6.3.2	Method Documentation	16
6.3.2.1	applinkWithSourceURL:targets:webURL:	16
6.3.3	Property Documentation	16
6.3.3.1	sourceURL	16
6.3.3.2	targets	17
6.3.3.3	webURL	17
6.4	BFApplinkNavigation Class Reference	17
6.4.1	Detailed Description	18
6.4.2	Method Documentation	18
6.4.2.1	defaultResolver	18
6.4.2.2	navigate:	18

6.4.2.3	navigateToAppLink:error:	18
6.4.2.4	navigateToURLInBackground:	18
6.4.2.5	navigateToURLInBackground:resolver:	18
6.4.2.6	navigationWithAppLink:extras:appLinkData:	18
6.4.2.7	resolveAppLinkInBackground:	18
6.4.2.8	resolveAppLinkInBackground:resolver:	18
6.4.2.9	setDefaultResolver:	18
6.4.3	Property Documentation	19
6.4.3.1	appLink	19
6.4.3.2	appLinkData	19
6.4.3.3	extras	19
6.5	<BFApplinkResolving> Protocol Reference	19
6.5.1	Detailed Description	19
6.5.2	Method Documentation	20
6.5.2.1	appLinkFromURLInBackground:	20
6.6	BFApplinkReturnToRefererController Class Reference	21
6.6.1	Detailed Description	21
6.6.2	Method Documentation	21
6.6.2.1	closeViewAnimated:	21
6.6.2.2	init	22
6.6.2.3	initWithDisplayAboveNavController:	22
6.6.2.4	removeFromNavController	22
6.6.2.5	showViewForRefererAppLink:	22
6.6.2.6	showViewForRefererURL:	22
6.6.3	Property Documentation	22
6.6.3.1	delegate	22
6.6.3.2	view	22
6.7	<BFApplinkReturnToRefererControllerDelegate> Protocol Reference	22
6.7.1	Detailed Description	23
6.7.2	Method Documentation	23
6.7.2.1	returnToRefererController:didNavigateToAppLink:type:	23
6.7.2.2	returnToRefererController:willNavigateToAppLink:	23
6.8	BFApplinkReturnToRefererView Class Reference	23
6.8.1	Detailed Description	24
6.8.2	Property Documentation	24
6.8.2.1	closed	24
6.8.2.2	delegate	24
6.8.2.3	includeStatusBarInSize	24
6.8.2.4	refererAppLink	24
6.8.2.5	textColor	24

6.9	<BFAppLinkReturnToRefererViewDelegate> Protocol Reference	24
6.9.1	Detailed Description	25
6.9.2	Method Documentation	25
6.9.2.1	returnToRefererViewDidTapInsideCloseButton:	25
6.9.2.2	returnToRefererViewDidTapInsideLink:link:	25
6.10	BFAppLinkTarget Class Reference	25
6.10.1	Detailed Description	26
6.10.2	Method Documentation	26
6.10.2.1	appLinkTargetWithURL:appStoreId:appName:	26
6.10.3	Property Documentation	26
6.10.3.1	appName	26
6.10.3.2	appStoreId	26
6.10.3.3	URL	26
6.11	BFExecutor Class Reference	26
6.11.1	Detailed Description	27
6.11.2	Method Documentation	27
6.11.2.1	defaultExecutor	27
6.11.2.2	execute:	27
6.11.2.3	executorWithBlock:	27
6.11.2.4	executorWithDispatchQueue:	27
6.11.2.5	executorWithOperationQueue:	28
6.11.2.6	immediateExecutor	28
6.11.2.7	mainThreadExecutor	28
6.12	BFMeasurementEvent Class Reference	28
6.12.1	Detailed Description	28
6.13	BFTask Class Reference	28
6.13.1	Detailed Description	29
6.13.2	Method Documentation	29
6.13.2.1	cancelledTask	29
6.13.2.2	continueWithBlock:	29
6.13.2.3	continueWithExecutor:withBlock:	30
6.13.2.4	continueWithExecutor:withSuccessBlock:	30
6.13.2.5	continueWithSuccessBlock:	30
6.13.2.6	taskForCompletionOfAllTasks:	30
6.13.2.7	taskForCompletionOfAllTasksWithResults:	31
6.13.2.8	taskFromExecutor:withBlock:	31
6.13.2.9	taskWithDelay:	31
6.13.2.10	taskWithError:	31
6.13.2.11	taskWithException:	31
6.13.2.12	taskWithResult:	31

6.13.2.13 waitUntilFinished	32
6.13.3 Property Documentation	32
6.13.3.1 cancelled	32
6.13.3.2 completed	32
6.13.3.3 error	32
6.13.3.4 exception	32
6.13.3.5 faulted	32
6.13.3.6 result	32
6.14 BFTaskCompletionSource Class Reference	33
6.14.1 Detailed Description	33
6.14.2 Method Documentation	33
6.14.2.1 cancel	33
6.14.2.2 setError:	33
6.14.2.3 setException:	34
6.14.2.4 setResult:	34
6.14.2.5 taskCompletionSource	34
6.14.2.6 trySetCancelled	34
6.14.2.7 trySetError:	34
6.14.2.8 trySetException:	34
6.14.2.9 trySetResult:	35
6.14.3 Property Documentation	35
6.14.3.1 task	35
6.15 BFURL Class Reference	35
6.15.1 Detailed Description	36
6.15.2 Method Documentation	36
6.15.2.1 URLWithInboundURL:sourceApplication:	36
6.15.2.2 URLWithURL:	36
6.15.3 Property Documentation	36
6.15.3.1 appLinkData	36
6.15.3.2 appLinkExtras	36
6.15.3.3 appLinkReferer	36
6.15.3.4 inputQueryParameters	36
6.15.3.5 inputURL	37
6.15.3.6 targetQueryParameters	37
6.15.3.7 targetURL	37
6.16 BFWebViewAppLinkResolver Class Reference	37
6.16.1 Detailed Description	37
6.16.2 Method Documentation	37
6.16.2.1 sharedInstance	38
6.17 Bolts Class Reference	38

6.17.1 Detailed Description	38
6.17.2 Method Documentation	38
6.17.2.1 version	38
6.18 Bound_BreakerTests Class Reference	38
6.18.1 Detailed Description	39
6.19 Game Class Reference	39
6.19.1 Detailed Description	40
6.19.2 Method Documentation	40
6.19.2.1 ballNotTouchingObstacle	40
6.19.2.2 ballTouchingObstacle	40
6.19.2.3 EasyDifficulty:	40
6.19.2.4 GameOver	40
6.19.2.5 HardDifficulty:	40
6.19.2.6 LeftLiftInside:	41
6.19.2.7 LeftLiftOutside:	41
6.19.2.8 LeftPress:	41
6.19.2.9 MediumDifficulty:	41
6.19.2.10 ObstaclesMoving	41
6.19.2.11 PlaceObstacles1	41
6.19.2.12 PlaceObstacles2	41
6.19.2.13 RightLiftInside:	41
6.19.2.14 RightLiftOutside:	42
6.19.2.15 RightPress:	42
6.19.2.16 Score	42
6.19.2.17 StartGame:	42
6.19.3 Member Data Documentation	42
6.19.3.1 Ball	42
6.19.3.2 EasyDifficultyButton	42
6.19.3.3 GameOver	42
6.19.3.4 HardDifficultyButton	42
6.19.3.5 longObstacle1	42
6.19.3.6 longObstacle2	42
6.19.3.7 longObstacle3	43
6.19.3.8 longObstacle4	43
6.19.3.9 MediumDifficultyButton	43
6.19.3.10 ObstacleMovement	43
6.19.3.11 ScoreLabel	43
6.19.3.12 TapLeft	43
6.19.3.13 TapRight	43
6.19.3.14 WelcomeLabel	43

6.20 Game() Category Reference	43
6.20.1 Detailed Description	43
6.21 leaderboardViewController Class Reference	44
6.21.1 Detailed Description	44
6.21.2 Property Documentation	44
6.21.2.1 score1	44
6.21.2.2 score10	45
6.21.2.3 score2	45
6.21.2.4 score3	45
6.21.2.5 score4	45
6.21.2.6 score5	45
6.21.2.7 score6	45
6.21.2.8 score7	45
6.21.2.9 score8	45
6.21.2.10 score9	45
6.21.2.11 username1	45
6.21.2.12 username10	45
6.21.2.13 username2	45
6.21.2.14 username3	46
6.21.2.15 username4	46
6.21.2.16 username5	46
6.21.2.17 username6	46
6.21.2.18 username7	46
6.21.2.19 username8	46
6.21.2.20 username9	46
6.22 leaderboardViewController() Category Reference	46
6.22.1 Detailed Description	46
6.23 LoginViewController Class Reference	46
6.23.1 Detailed Description	47
6.23.2 Method Documentation	47
6.23.2.1 loginButton:	47
6.23.2.2 registerAction:	47
6.23.3 Property Documentation	47
6.23.3.1 emailField	47
6.23.3.2 loginPasswordField	47
6.23.3.3 loginUsernameField	47
6.23.3.4 passwordField	48
6.23.3.5 reEnterPasswordField	48
6.23.3.6 userNameField	48
6.24 LoginViewController() Category Reference	48

6.24.1 Detailed Description	48
6.25 Parse Class Reference	48
6.25.1 Detailed Description	49
6.25.2 Method Documentation	49
6.25.2.1 enableLocalDatastore	49
6.25.2.2 getApplicationId	49
6.25.2.3 getClientKey	49
6.25.2.4 isLocalDatastoreEnabled	49
6.25.2.5 logLevel	49
6.25.2.6 setApplicationId:clientKey:	50
6.25.2.7 setLogLevel:	50
6.26 ParseCrashReporting Class Reference	50
6.26.1 Detailed Description	51
6.26.2 Method Documentation	51
6.26.2.1 enable	51
6.26.2.2 isCrashReportingEnabled	51
6.27 PF_Twitter Class Reference	51
6.27.1 Detailed Description	52
6.27.2 Method Documentation	52
6.27.2.1 authorizeInBackground	52
6.27.2.2 authorizeWithSuccess:failure:cancel:	52
6.27.2.3 signRequest:	52
6.27.3 Property Documentation	52
6.27.3.1 authToken	52
6.27.3.2 authTokenSecret	52
6.27.3.3 consumerKey	53
6.27.3.4 consumerSecret	53
6.27.3.5 screenName	53
6.27.3.6 userId	53
6.28 PFAcl Class Reference	53
6.28.1 Detailed Description	54
6.28.2 Method Documentation	54
6.28.2.1 ACL	54
6.28.2.2 ACLWithUser:	54
6.28.2.3 getPublicReadAccess	55
6.28.2.4 getPublicWriteAccess	55
6.28.2.5 getReadAccessForRole:	55
6.28.2.6 getReadAccessForRoleWithName:	55
6.28.2.7 getReadAccessForUser:	55
6.28.2.8 getReadAccessForUserId:	56

6.28.2.9	getWriteAccessForRole:	56
6.28.2.10	getWriteAccessForRoleWithName:	56
6.28.2.11	getWriteAccessForUser:	56
6.28.2.12	getWriteAccessForUserId:	57
6.28.2.13	setDefaultACL:withAccessForCurrentUser:	57
6.28.2.14	setPublicReadAccess:	57
6.28.2.15	setPublicWriteAccess:	57
6.28.2.16	setReadAccess:forRole:	58
6.28.2.17	setReadAccess:forRoleWithName:	58
6.28.2.18	setReadAccess:forUser:	58
6.28.2.19	setReadAccess:forUserId:	58
6.28.2.20	setWriteAccess:forRole:	58
6.28.2.21	setWriteAccess:forRoleWithName:	58
6.28.2.22	setWriteAccess:forUser:	59
6.28.2.23	setWriteAccess:forUserId:	59
6.29	PFAalytics Class Reference	59
6.29.1	Detailed Description	60
6.29.2	Method Documentation	60
6.29.2.1	trackAppOpenedWithLaunchOptions:	60
6.29.2.2	trackAppOpenedWithLaunchOptionsInBackground:block:	60
6.29.2.3	trackAppOpenedWithRemoteNotificationPayload:	60
6.29.2.4	trackAppOpenedWithRemoteNotificationPayloadInBackground:block:	61
6.29.2.5	trackEvent:	61
6.29.2.6	trackEvent:dimensions:	61
6.29.2.7	trackEventInBackground:block:	61
6.29.2.8	trackEventInBackground:dimensions:block:	62
6.30	PFAnonymousUtils Class Reference	62
6.30.1	Detailed Description	63
6.30.2	Method Documentation	63
6.30.2.1	isLinkedWithUser:	63
6.30.2.2	logInInBackground	63
6.30.2.3	logInWithBlock:	63
6.30.2.4	logInWithTarget:selector:	64
6.31	PFCloud Class Reference	64
6.31.1	Detailed Description	64
6.31.2	Method Documentation	64
6.31.2.1	callFunction:withParameters:	64
6.31.2.2	callFunction:withParameters:error:	65
6.31.2.3	callFunctionInBackground:withParameters:	66
6.31.2.4	callFunctionInBackground:withParameters:block:	66

6.31.2.5	callFunctionInBackground:withParameters:target:selector:	66
6.32	PFCollectionViewCell Class Reference	66
6.32.1	Detailed Description	67
6.32.2	Method Documentation	67
6.32.2.1	updateFromObject:	67
6.32.3	Property Documentation	67
6.32.3.1	imageView	67
6.32.3.2	textLabel	67
6.33	PFConfig Class Reference	68
6.33.1	Detailed Description	68
6.33.2	Method Documentation	68
6.33.2.1	currentConfig	68
6.33.2.2	getConfig	69
6.33.2.3	getConfig:	69
6.33.2.4	getConfigInBackground	69
6.33.2.5	getConfigInBackgroundWithBlock:	69
6.33.2.6	objectForKey:	69
6.33.2.7	objectForKeyedSubscript:	69
6.34	PFFacebookUtils Class Reference	70
6.34.1	Detailed Description	71
6.34.2	Method Documentation	72
6.34.2.1	facebookLoginBehavior	72
6.34.2.2	facebookLoginManager	72
6.34.2.3	handleOpenURL:	72
6.34.2.4	initializeFacebook	72
6.34.2.5	initializeFacebookWithApplicationLaunchOptions:	72
6.34.2.6	initializeFacebookWithUrlSchemeSuffix:	73
6.34.2.7	initializeWithApplicationId:	73
6.34.2.8	initializeWithApplicationId:urlSchemeSuffix:	73
6.34.2.9	isLinkedWithUser:	73
6.34.2.10	isLinkedWithUser:	74
6.34.2.11	linkUser:facebookId:accessToken:expirationDate:block:	74
6.34.2.12	linkUser:facebookId:accessToken:expirationDate:target:selector:	74
6.34.2.13	linkUser:permissions:	74
6.34.2.14	linkUser:permissions:block:	75
6.34.2.15	linkUser:permissions:target:selector:	75
6.34.2.16	linkUserInBackground:facebookId:accessToken:expirationDate:	75
6.34.2.17	linkUserInBackground:permissions:	75
6.34.2.18	linkUserInBackground:withAccessToken:	75
6.34.2.19	linkUserInBackground:withAccessToken:block:	76

6.34.2.20 linkUserInBackground:withPublishPermissions:	76
6.34.2.21 linkUserInBackground:withPublishPermissions:block:	76
6.34.2.22 linkUserInBackground:withReadPermissions:	76
6.34.2.23 linkUserInBackground:withReadPermissions:block:	77
6.34.2.24 logInInBackgroundWithAccessToken:	77
6.34.2.25 logInInBackgroundWithAccessToken:block:	77
6.34.2.26 logInInBackgroundWithPublishPermissions:	78
6.34.2.27 logInInBackgroundWithPublishPermissions:block:	78
6.34.2.28 logInInBackgroundWithReadPermissions:	78
6.34.2.29 logInInBackgroundWithReadPermissions:block:	78
6.34.2.30 logInWithFacebookId:accessToken:expirationDate:block:	79
6.34.2.31 logInWithFacebookId:accessToken:expirationDate:target:selector:	79
6.34.2.32 logInWithFacebookIdInBackground:accessToken:expirationDate:	79
6.34.2.33 logInWithPermissions:block:	79
6.34.2.34 logInWithPermissions:target:selector:	79
6.34.2.35 logInWithPermissionsInBackground:	80
6.34.2.36 reauthorizeUser:withPublishPermissions:audience:block:	80
6.34.2.37 reauthorizeUser:withPublishPermissions:audience:target:selector:	80
6.34.2.38 reauthorizeUserInBackground:withPublishPermissions:audience:	80
6.34.2.39 session	81
6.34.2.40 setFacebookLoginBehavior:	81
6.34.2.41 unlinkUser:	81
6.34.2.42 unlinkUser:error:	81
6.34.2.43 unlinkUserInBackground:	81
6.34.2.44 unlinkUserInBackground:	81
6.34.2.45 unlinkUserInBackground:block:	82
6.34.2.46 unlinkUserInBackground:block:	82
6.34.2.47 unlinkUserInBackground:target:selector:	82
6.35 PFFile Class Reference	82
6.35.1 Detailed Description	83
6.35.2 Method Documentation	83
6.35.2.1 cancel	83
6.35.2.2 fileWithData:	83
6.35.2.3 fileWithData:contentType:	84
6.35.2.4 fileWithName:contentsAtPath:	84
6.35.2.5 fileWithName:data:	84
6.35.2.6 fileWithName:data:contentType:	84
6.35.2.7 getData	85
6.35.2.8 getData:	85
6.35.2.9 getDataInBackground	85

6.35.2.10	getDataInBackgroundWithBlock:	85
6.35.2.11	getDataInBackgroundWithBlock:progressBlock:	85
6.35.2.12	getDataInBackgroundWithTarget:selector:	86
6.35.2.13	getDataStream	86
6.35.2.14	getDataStream:	86
6.35.2.15	getDataStreamInBackground	86
6.35.2.16	getDataStreamInBackgroundWithBlock:	86
6.35.2.17	getDataStreamInBackgroundWithBlock:progressBlock:	87
6.35.2.18	save	88
6.35.2.19	save:	88
6.35.2.20	saveInBackground	88
6.35.2.21	saveInBackgroundWithBlock:	88
6.35.2.22	saveInBackgroundWithBlock:progressBlock:	88
6.35.2.23	saveInBackgroundWithTarget:selector:	89
6.35.3	Property Documentation	89
6.35.3.1	isDataAvailable	89
6.35.3.2	isDirty	89
6.35.3.3	name	89
6.35.3.4	url	89
6.36	PGeoPoint Class Reference	89
6.36.1	Detailed Description	90
6.36.2	Method Documentation	90
6.36.2.1	distanceInKilometersTo:	90
6.36.2.2	distanceInMilesTo:	90
6.36.2.3	distanceInRadiansTo:	91
6.36.2.4	geoPoint	92
6.36.2.5	geoPointForCurrentLocationInBackground:	92
6.36.2.6	geoPointWithLatitude:longitude:	92
6.36.2.7	geoPointWithLocation:	92
6.36.3	Property Documentation	92
6.36.3.1	latitude	92
6.36.3.2	longitude	93
6.37	PImageView Class Reference	93
6.37.1	Detailed Description	93
6.37.2	Method Documentation	93
6.37.2.1	loadInBackground	93
6.37.2.2	loadInBackground:	93
6.37.2.3	loadInBackground:progressBlock:	94
6.37.3	Property Documentation	94
6.37.3.1	file	94

6.38	PFInstallation Class Reference	94
6.38.1	Detailed Description	95
6.38.2	Method Documentation	95
6.38.2.1	currentInstallation	95
6.38.2.2	query	95
6.38.2.3	setDeviceTokenFromData:	96
6.38.3	Property Documentation	96
6.38.3.1	badge	96
6.38.3.2	channels	96
6.38.3.3	deviceToken	96
6.38.3.4	deviceType	96
6.38.3.5	installationId	96
6.38.3.6	timeZone	96
6.39	PFLoginView Class Reference	96
6.39.1	Detailed Description	97
6.39.2	Method Documentation	97
6.39.2.1	__attribute__	97
6.39.2.2	__attribute__	98
6.39.2.3	initWithFields:	98
6.39.3	Property Documentation	98
6.39.3.1	dismissButton	98
6.39.3.2	emailAsUsername	98
6.39.3.3	facebookButton	98
6.39.3.4	fields	98
6.39.3.5	loginButton	98
6.39.3.6	logo	99
6.39.3.7	passwordField	99
6.39.3.8	passwordForgottenButton	99
6.39.3.9	presentingViewController	99
6.39.3.10	signUpButton	99
6.39.3.11	twitterButton	99
6.39.3.12	usernameField	99
6.40	PFLoginViewController Class Reference	99
6.40.1	Detailed Description	100
6.40.2	Property Documentation	100
6.40.2.1	delegate	100
6.40.2.2	emailAsUsername	100
6.40.2.3	facebookPermissions	100
6.40.2.4	fields	101
6.40.2.5	loginView	101

6.40.2.6	signUpController	101
6.41	<PFLoginViewControllerDelegate> Protocol Reference	101
6.41.1	Detailed Description	102
6.41.2	Method Documentation	102
6.41.2.1	loginViewController:didFailToLoginWithError:	102
6.41.2.2	loginViewController:didLoginUser:	102
6.41.2.3	loginViewController:shouldBeginLoginWithUsername:password:	102
6.41.2.4	loginViewControllerDidCancelLogin:	102
6.42	PFNetworkActivityIndicatorManager Class Reference	103
6.42.1	Detailed Description	103
6.42.2	Method Documentation	103
6.42.2.1	decrementActivityCount	103
6.42.2.2	incrementActivityCount	103
6.42.2.3	sharedManager	104
6.42.3	Property Documentation	104
6.42.3.1	enabled	104
6.42.3.2	networkActivityCount	104
6.42.3.3	networkActivityIndicatorVisible	104
6.43	PFObjec Class Reference	104
6.43.1	Detailed Description	108
6.43.2	Method Documentation	108
6.43.2.1	addObject:forKey:	108
6.43.2.2	addObjectsFromArray:forKey:	108
6.43.2.3	addUniqueObject:forKey:	108
6.43.2.4	addUniqueObjectsFromArray:forKey:	108
6.43.2.5	allKeys	108
6.43.2.6	delete	109
6.43.2.7	delete:	109
6.43.2.8	deleteAll:	109
6.43.2.9	deleteAll:error:	109
6.43.2.10	deleteAllInBackground:	109
6.43.2.11	deleteAllInBackground:block:	110
6.43.2.12	deleteAllInBackground:target:selector:	111
6.43.2.13	deleteEventually	111
6.43.2.14	deleteInBackground	111
6.43.2.15	deleteInBackgroundWithBlock:	111
6.43.2.16	deleteInBackgroundWithTarget:selector:	111
6.43.2.17	fetch	111
6.43.2.18	fetch:	111
6.43.2.19	fetchAll:	112

6.43.2.20 fetchAll:error:	112
6.43.2.21 fetchAllIfNeeded:	112
6.43.2.22 fetchAllIfNeeded:error:	112
6.43.2.23 fetchAllIfNeededInBackground:	112
6.43.2.24 fetchAllIfNeededInBackground:block:	113
6.43.2.25 fetchAllIfNeededInBackground:target:selector:	114
6.43.2.26 fetchAllInBackground:	114
6.43.2.27 fetchAllInBackground:block:	114
6.43.2.28 fetchAllInBackground:target:selector:	114
6.43.2.29 fetchFromLocalDatastore	114
6.43.2.30 fetchFromLocalDatastore:	114
6.43.2.31 fetchFromLocalDatastoreInBackground	114
6.43.2.32 fetchFromLocalDatastoreInBackgroundWithBlock:	115
6.43.2.33 fetchIfNeeded	115
6.43.2.34 fetchIfNeeded:	115
6.43.2.35 fetchIfNeededInBackground	115
6.43.2.36 fetchIfNeededInBackgroundWithBlock:	115
6.43.2.37 fetchIfNeededInBackgroundWithTarget:selector:	115
6.43.2.38 fetchInBackground	115
6.43.2.39 fetchInBackgroundWithBlock:	116
6.43.2.40 fetchInBackgroundWithTarget:selector:	117
6.43.2.41 incrementKey:	117
6.43.2.42 incrementKey:byAmount:	117
6.43.2.43 init	117
6.43.2.44 initWithClassName:	117
6.43.2.45 isDataAvailable	117
6.43.2.46 isDirty	118
6.43.2.47 isDirtyForKey:	118
6.43.2.48 object	118
6.43.2.49 objectForKey:	118
6.43.2.50 objectForKeyedSubscript:	118
6.43.2.51 objectWithClassName:	118
6.43.2.52 objectWithClassName:dictionary:	119
6.43.2.53 objectWithoutDataWithClassName:objectId:	119
6.43.2.54 objectWithoutDataWithObjectId:	119
6.43.2.55 pin	120
6.43.2.56 pin:	120
6.43.2.57 pinAll:	120
6.43.2.58 pinAll:error:	121
6.43.2.59 pinAll:withName:	121

6.43.2.60 pinAll:withName:error:	122
6.43.2.61 pinAllInBackground:	122
6.43.2.62 pinAllInBackground:block:	122
6.43.2.63 pinAllInBackground:withName:	123
6.43.2.64 pinAllInBackground:withName:block:	123
6.43.2.65 pinInBackground	124
6.43.2.66 pinInBackgroundWithBlock:	124
6.43.2.67 pinInBackgroundWithName:	124
6.43.2.68 pinInBackgroundWithName:block:	125
6.43.2.69 pinWithName:	125
6.43.2.70 pinWithName:error:	125
6.43.2.71 query	126
6.43.2.72 queryWithPredicate:	126
6.43.2.73 registerSubclass	126
6.43.2.74 relationForKey:	126
6.43.2.75 relationforKey:	127
6.43.2.76 removeObject:forKey:	127
6.43.2.77 removeObjectForKey:	127
6.43.2.78 removeObjectsinArray:forKey:	127
6.43.2.79 save	127
6.43.2.80 save:	127
6.43.2.81 saveAll:	128
6.43.2.82 saveAll:error:	128
6.43.2.83 saveAllInBackground:	128
6.43.2.84 saveAllInBackground:block:	128
6.43.2.85 saveAllInBackground:target:selector:	129
6.43.2.86 saveEventually	129
6.43.2.87 saveEventually:	129
6.43.2.88 saveInBackground	129
6.43.2.89 saveInBackgroundWithBlock:	129
6.43.2.90 saveInBackgroundWithTarget:selector:	130
6.43.2.91 setObject:forKey:	130
6.43.2.92 setObject:forKeyedSubscript:	131
6.43.2.93 unpin	131
6.43.2.94 unpin:	131
6.43.2.95 unpinAll:	132
6.43.2.96 unpinAll:error:	132
6.43.2.97 unpinAll:withName:	132
6.43.2.98 unpinAll:withName:error:	132
6.43.2.99 unpinAllInBackground:	133

6.43.2.100	unpinAllInBackground:block:	133
6.43.2.101	unpinAllInBackground:withName:	133
6.43.2.102	unpinAllInBackground:withName:block:	134
6.43.2.103	unpinAllObjects	134
6.43.2.104	unpinAllObjects:	134
6.43.2.105	unpinAllObjectsInBackground	135
6.43.2.106	unpinAllObjectsInBackgroundWithBlock:	135
6.43.2.107	unpinAllObjectsInBackgroundWithName:	135
6.43.2.108	unpinAllObjectsInBackgroundWithName:block:	135
6.43.2.109	unpinAllObjectsWithName:	135
6.43.2.110	unpinAllObjectsWithName:error:	136
6.43.2.111	unpinInBackground	136
6.43.2.112	unpinInBackgroundWithBlock:	136
6.43.2.113	unpinInBackgroundWithName:	136
6.43.2.114	unpinInBackgroundWithName:block:	137
6.43.2.115	unpinWithName:	137
6.43.2.116	unpinWithName:error:	137
6.43.3	Member Data Documentation	138
6.43.3.1	dirty	138
6.43.3.2	estimatedData	138
6.43.3.3	operationSetQueue	138
6.43.4	Property Documentation	138
6.43.4.1	ACL	138
6.43.4.2	createdAt	138
6.43.4.3	objectId	138
6.43.4.4	parseClassName	138
6.43.4.5	updatedAt	138
6.44	PFObject(Subclass) Category Reference	138
6.44.1	Detailed Description	139
6.44.2	Method Documentation	139
6.44.2.1	init	139
6.44.2.2	object	139
6.44.2.3	objectWithoutDataWithObjectId:	139
6.44.2.4	query	139
6.44.2.5	queryWithPredicate:	140
6.44.2.6	registerSubclass	140
6.45	PFProduct Class Reference	140
6.45.1	Detailed Description	141
6.45.2	Property Documentation	141
6.45.2.1	downloadName	141

6.45.2.2	icon	141
6.45.2.3	order	141
6.45.2.4	productIdentifier	141
6.45.2.5	subtitle	141
6.45.2.6	title	142
6.46	PFPProductTableViewController Class Reference	142
6.46.1	Detailed Description	142
6.46.2	Method Documentation	142
6.46.2.1	initWithStyle:	142
6.47	PFPurchase Class Reference	143
6.47.1	Detailed Description	143
6.47.2	Method Documentation	143
6.47.2.1	addObserverForProduct:block:	143
6.47.2.2	assetContentPathForProduct:	143
6.47.2.3	buyProduct:block:	143
6.47.2.4	downloadAssetForTransaction:completion:	144
6.47.2.5	downloadAssetForTransaction:completion:progress:	144
6.47.2.6	restore	144
6.48	PFPurchaseTableViewCell Class Reference	144
6.48.1	Detailed Description	145
6.48.2	Property Documentation	145
6.48.2.1	priceLabel	145
6.48.2.2	progressView	145
6.48.2.3	state	145
6.49	PFPush Class Reference	146
6.49.1	Detailed Description	147
6.49.2	Method Documentation	147
6.49.2.1	clearExpiration	147
6.49.2.2	expireAfterTimeInterval:	147
6.49.2.3	expireAtDate:	148
6.49.2.4	getSubscribedChannels:	148
6.49.2.5	getSubscribedChannelsInBackground	148
6.49.2.6	getSubscribedChannelsInBackgroundWithBlock:	148
6.49.2.7	getSubscribedChannelsInBackgroundWithTarget:selector:	148
6.49.2.8	handlePush:	148
6.49.2.9	push	149
6.49.2.10	sendPush:	149
6.49.2.11	sendPushDataToChannel:withData:error:	149
6.49.2.12	sendPushDataToChannelInBackground:withData:	149
6.49.2.13	sendPushDataToChannelInBackground:withData:block:	149

6.49.2.14	sendPushDataToChannelInBackground:withData:target:selector:	150
6.49.2.15	sendPushDataToQuery:withData:error:	150
6.49.2.16	sendPushDataToQueryInBackground:withData:	150
6.49.2.17	sendPushDataToQueryInBackground:withData:block:	150
6.49.2.18	sendPushInBackground	151
6.49.2.19	sendPushInBackgroundWithBlock:	151
6.49.2.20	sendPushInBackgroundWithTarget:selector:	151
6.49.2.21	sendPushMessageToChannel:withMessage:error:	151
6.49.2.22	sendPushMessageToChannelInBackground:withMessage:	151
6.49.2.23	sendPushMessageToChannelInBackground:withMessage:block:	152
6.49.2.24	sendPushMessageToChannelInBackground:withMessage:target:selector:	153
6.49.2.25	sendPushMessageToQuery:withMessage:error:	153
6.49.2.26	sendPushMessageToQueryInBackground:withMessage:	153
6.49.2.27	sendPushMessageToQueryInBackground:withMessage:block:	153
6.49.2.28	setChannel:	153
6.49.2.29	setChannels:	154
6.49.2.30	setData:	154
6.49.2.31	setMessage:	154
6.49.2.32	setPushToAndroid:	154
6.49.2.33	setPushToIOS:	154
6.49.2.34	setQuery:	155
6.49.2.35	storeDeviceToken:	155
6.49.2.36	subscribeToChannel:error:	155
6.49.2.37	subscribeToChannelInBackground:	155
6.49.2.38	subscribeToChannelInBackground:block:	155
6.49.2.39	subscribeToChannelInBackground:target:selector:	156
6.49.2.40	unsubscribeFromChannel:error:	156
6.49.2.41	unsubscribeFromChannelInBackground:	156
6.49.2.42	unsubscribeFromChannelInBackground:block:	156
6.49.2.43	unsubscribeFromChannelInBackground:target:selector:	156
6.50	PFQuery Class Reference	157
6.50.1	Detailed Description	159
6.50.2	Method Documentation	159
6.50.2.1	addAscendingOrder:	159
6.50.2.2	addDescendingOrder:	159
6.50.2.3	cancel	160
6.50.2.4	clearAllCachedResults	160
6.50.2.5	clearCachedResult	160
6.50.2.6	countObjects	160
6.50.2.7	countObjects:	160

6.50.2.8	countObjectsInBackground	160
6.50.2.9	countObjectsInBackgroundWithBlock:	160
6.50.2.10	countObjectsInBackgroundWithTarget:selector:	161
6.50.2.11	findObjects	161
6.50.2.12	findObjects:	161
6.50.2.13	findObjectsInBackground	161
6.50.2.14	findObjectsInBackgroundWithBlock:	161
6.50.2.15	findObjectsInBackgroundWithTarget:selector:	161
6.50.2.16	fromLocalDatastore	161
6.50.2.17	fromPin	162
6.50.2.18	fromPinWithName:	162
6.50.2.19	getFirstObject	163
6.50.2.20	getFirstObject:	163
6.50.2.21	getFirstObjectInBackground	163
6.50.2.22	getFirstObjectInBackgroundWithBlock:	163
6.50.2.23	getFirstObjectInBackgroundWithTarget:selector:	164
6.50.2.24	getObjectInBackgroundWithId:	164
6.50.2.25	getObjectInBackgroundWithId:block:	164
6.50.2.26	getObjectInBackgroundWithId:target:selector:	164
6.50.2.27	getObjectOfClass:objectId:	164
6.50.2.28	getObjectOfClass:objectId:error:	165
6.50.2.29	getObjectWithId:	165
6.50.2.30	getObjectWithId:error:	165
6.50.2.31	getUserObjectWithId:	166
6.50.2.32	getUserObjectWithId:error:	166
6.50.2.33	hasCachedResult	166
6.50.2.34	ignoreACLs	166
6.50.2.35	includeKey:	167
6.50.2.36	initWithClassName:	168
6.50.2.37	orderByAscending:	168
6.50.2.38	orderByDescending:	168
6.50.2.39	orderBySortDescriptor:	168
6.50.2.40	orderBySortDescriptors:	168
6.50.2.41	orQueryWithSubqueries:	169
6.50.2.42	PARSE_DEPRECATED	169
6.50.2.43	queryWithClassName:	169
6.50.2.44	queryWithClassName:predicate:	169
6.50.2.45	selectKeys:	170
6.50.2.46	whereKey:containedIn:	170
6.50.2.47	whereKey:containsAllObjectsInArray:	170

6.50.2.48 whereKey:containsString:	171
6.50.2.49 whereKey:doesNotMatchKey:inQuery:	171
6.50.2.50 whereKey:doesNotMatchQuery:	171
6.50.2.51 whereKey:equalTo:	171
6.50.2.52 whereKey:greaterThan:	172
6.50.2.53 whereKey:greaterThanOrEqualTo:	172
6.50.2.54 whereKey:hasPrefix:	172
6.50.2.55 whereKey:hasSuffix:	172
6.50.2.56 whereKey:lessThan:	173
6.50.2.57 whereKey:lessThanOrEqualTo:	173
6.50.2.58 whereKey:matchesKey:inQuery:	173
6.50.2.59 whereKey:matchesQuery:	173
6.50.2.60 whereKey:matchesRegex:	174
6.50.2.61 whereKey:matchesRegex:modifiers:	174
6.50.2.62 whereKey:nearGeoPoint:	174
6.50.2.63 whereKey:nearGeoPoint:withinKilometers:	175
6.50.2.64 whereKey:nearGeoPoint:withinMiles:	175
6.50.2.65 whereKey:nearGeoPoint:withinRadians:	175
6.50.2.66 whereKey:notContainedIn:	176
6.50.2.67 whereKey:notEqualTo:	176
6.50.2.68 whereKey:withinGeoBoxFromSouthwest:toNortheast:	176
6.50.2.69 whereKeyDoesNotExist:	176
6.50.2.70 whereKeyExists:	177
6.50.3 Property Documentation	177
6.50.3.1 cachePolicy	177
6.50.3.2 limit	177
6.50.3.3 maxCacheAge	177
6.50.3.4 parseClassName	177
6.50.3.5 skip	178
6.50.3.6 trace	178
6.51 PFQueryCollectionViewController Class Reference	178
6.51.1 Detailed Description	179
6.51.2 Method Documentation	179
6.51.2.1 clear	179
6.51.2.2 collectionView:cellForItemAtIndexPath:object:	179
6.51.2.3 collectionViewReusableViewForNextPageAction:	180
6.51.2.4 initWithClassName:	180
6.51.2.5 initWithCollectionViewLayout:className:	180
6.51.2.6 loadNextPage	180
6.51.2.7 loadObjects	181

6.51.2.8	loadObjects:clear:	181
6.51.2.9	NS_REQUIRES_SUPER	181
6.51.2.10	objectAtIndexPath:	181
6.51.2.11	objectsDidLoad:	181
6.51.2.12	queryForCollection	181
6.51.3	Property Documentation	181
6.51.3.1	loading	181
6.51.3.2	loadingViewEnabled	182
6.51.3.3	objects	182
6.51.3.4	objectsPerPage	182
6.51.3.5	paginationEnabled	182
6.51.3.6	parseClassName	182
6.51.3.7	pullToRefreshEnabled	182
6.52	PFQueryTableViewController Class Reference	182
6.52.1	Detailed Description	183
6.52.2	Method Documentation	184
6.52.2.1	clear	184
6.52.2.2	initWithClassName:	184
6.52.2.3	initWithStyle:className:	184
6.52.2.4	loadNextPage	184
6.52.2.5	loadObjects	184
6.52.2.6	loadObjects:clear:	184
6.52.2.7	objectAtIndexPath:	185
6.52.2.8	objectsDidLoad:	185
6.52.2.9	objectsWillLoad	185
6.52.2.10	queryForTable	185
6.52.2.11	tableView:cellForNextPageAtIndexPath:	185
6.52.2.12	tableView:cellForRowAtIndexPath:object:	186
6.52.3	Property Documentation	186
6.52.3.1	imageKey	186
6.52.3.2	loading	186
6.52.3.3	loadingViewEnabled	186
6.52.3.4	objects	186
6.52.3.5	objectsPerPage	186
6.52.3.6	paginationEnabled	187
6.52.3.7	parseClassName	187
6.52.3.8	placeholderImage	187
6.52.3.9	pullToRefreshEnabled	187
6.52.3.10	textKey	187
6.53	PFRelation Class Reference	187

6.53.1 Detailed Description	188
6.53.2 Method Documentation	188
6.53.2.1 addObject:	188
6.53.2.2 query	188
6.53.2.3 removeObject:	188
6.53.3 Property Documentation	188
6.53.3.1 targetClass	188
6.54 PFRole Class Reference	188
6.54.1 Detailed Description	189
6.54.2 Method Documentation	189
6.54.2.1 initWithName:	189
6.54.2.2 initWithName:acl:	189
6.54.2.3 roleWithName:	190
6.54.2.4 roleWithName:acl:	190
6.54.3 Property Documentation	190
6.54.3.1 name	190
6.54.3.2 roles	190
6.54.3.3 users	190
6.55 PFSession Class Reference	191
6.55.1 Detailed Description	191
6.55.2 Method Documentation	191
6.55.2.1 getSessionInBackground	191
6.55.2.2 getSessionInBackgroundWithBlock:	191
6.55.3 Property Documentation	192
6.55.3.1 sessionToken	192
6.56 PFSignUpView Class Reference	192
6.56.1 Detailed Description	193
6.56.2 Method Documentation	193
6.56.2.1 initWithFields:	193
6.56.3 Property Documentation	193
6.56.3.1 additionalField	193
6.56.3.2 dismissButton	193
6.56.3.3 emailAsUsername	193
6.56.3.4 emailField	193
6.56.3.5 fields	193
6.56.3.6 logo	194
6.56.3.7 passwordField	194
6.56.3.8 presentingViewController	194
6.56.3.9 signUpButton	194
6.56.3.10 usernameField	194

6.57	PFSignUpViewController Class Reference	194
6.57.1	Detailed Description	195
6.57.2	Property Documentation	195
6.57.2.1	delegate	195
6.57.2.2	emailAsUsername	195
6.57.2.3	fields	195
6.57.2.4	minPasswordLength	195
6.57.2.5	signUpView	195
6.58	<PFSignUpViewControllerDelegate> Protocol Reference	196
6.58.1	Detailed Description	196
6.58.2	Method Documentation	196
6.58.2.1	signUpViewController:didFailToSignUpWithError:	196
6.58.2.2	signUpViewController:didSignUpUser:	197
6.58.2.3	signUpViewController:shouldBeginSignUp:	197
6.58.2.4	signUpViewControllerDidCancelSignUp:	197
6.59	<PFSubclassing> Protocol Reference	197
6.59.1	Detailed Description	198
6.59.2	Method Documentation	198
6.59.2.1	object	198
6.59.2.2	objectWithoutDataWithObjectId:	198
6.59.2.3	parseClassName	198
6.59.2.4	query	198
6.59.2.5	queryWithPredicate:	199
6.59.2.6	registerSubclass	199
6.60	PFTableViewCell Class Reference	199
6.60.1	Detailed Description	199
6.60.2	Property Documentation	200
6.60.2.1	imageView	200
6.61	PFTextField Class Reference	200
6.61.1	Detailed Description	200
6.61.2	Method Documentation	200
6.61.2.1	initWithFrame:separatorStyle:	200
6.61.3	Property Documentation	201
6.61.3.1	separatorStyle	201
6.61.3.2	UI_APPEARANCE_SELECTOR	201
6.62	PFTwitterUtils Class Reference	201
6.62.1	Detailed Description	202
6.62.2	Method Documentation	202
6.62.2.1	initWithConsumerKey:consumerSecret:	202
6.62.2.2	isLinkedWithUser:	202

6.62.2.3	linkUser:	203
6.62.2.4	linkUser:block:	203
6.62.2.5	linkUser:target:selector:	203
6.62.2.6	linkUser:twitterId:screenName:authToken:authTokenSecret:block:	203
6.62.2.7	linkUser:twitterId:screenName:authToken:authTokenSecret:target:selector:	203
6.62.2.8	linkUserInBackground:	204
6.62.2.9	linkUserInBackground:twitterId:screenName:authToken:authTokenSecret:	204
6.62.2.10	loginInBackground	204
6.62.2.11	loginWithBlock:	204
6.62.2.12	loginWithTarget:selector:	205
6.62.2.13	loginWithTwitterId:screenName:authToken:authTokenSecret:block:	205
6.62.2.14	loginWithTwitterId:screenName:authToken:authTokenSecret:target:selector:	205
6.62.2.15	loginWithTwitterIdInBackground:screenName:authToken:authTokenSecret:	205
6.62.2.16	twitter	205
6.62.2.17	unlinkUser:	205
6.62.2.18	unlinkUser:error:	206
6.62.2.19	unlinkUserInBackground:	206
6.62.2.20	unlinkUserInBackground:block:	206
6.62.2.21	unlinkUserInBackground:target:selector:	206
6.63	PFUser Class Reference	207
6.63.1	Detailed Description	208
6.63.2	Method Documentation	208
6.63.2.1	become:	208
6.63.2.2	become:error:	208
6.63.2.3	becomeInBackground:	209
6.63.2.4	becomeInBackground:block:	209
6.63.2.5	becomeInBackground:target:selector:	209
6.63.2.6	currentUser	209
6.63.2.7	enableAutomaticUser	209
6.63.2.8	enableRevocableSessionInBackground	210
6.63.2.9	enableRevocableSessionInBackgroundWithBlock:	210
6.63.2.10	isAuthenticated	210
6.63.2.11	loginWithUsername:password:	210
6.63.2.12	loginWithUsername:password:error:	211
6.63.2.13	loginWithUsernameInBackground:password:	211
6.63.2.14	loginWithUsernameInBackground:password:block:	211
6.63.2.15	loginWithUsernameInBackground:password:target:selector:	212
6.63.2.16	logout	212
6.63.2.17	logoutInBackground	212
6.63.2.18	logoutInBackgroundWithBlock:	212

6.63.2.19	requestPasswordResetForEmail:	212
6.63.2.20	requestPasswordResetForEmail:error:	212
6.63.2.21	requestPasswordResetForEmailInBackground:	213
6.63.2.22	requestPasswordResetForEmailInBackground:block:	213
6.63.2.23	requestPasswordResetForEmailInBackground:target:selector:	213
6.63.2.24	signUp	213
6.63.2.25	signUp:	214
6.63.2.26	signUpInBackground	214
6.63.2.27	signUpInBackgroundWithBlock:	214
6.63.2.28	signUpInBackgroundWithTarget:selector:	214
6.63.2.29	user	214
6.63.3	Property Documentation	215
6.63.3.1	email	215
6.63.3.2	isNew	215
6.63.3.3	password	215
6.63.3.4	sessionToken	215
6.63.3.5	username	215
6.64	ProfileViewController Class Reference	215
6.64.1	Detailed Description	216
6.64.2	Method Documentation	216
6.64.2.1	deleteButton:	216
6.64.2.2	logoutButton:	216
6.64.2.3	saveChanges:	216
6.64.3	Property Documentation	216
6.64.3.1	emailLabel	216
6.64.3.2	highScoreLabel	216
6.64.3.3	userNameField	216
6.64.3.4	userNameLabel	217
6.65	ProfileViewController() Category Reference	217
6.65.1	Detailed Description	217
6.66	ViewController Class Reference	217
6.66.1	Detailed Description	217
6.67	ViewController() Category Reference	218
6.67.1	Detailed Description	218
7	File Documentation	219
7.1	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLink.h File Reference	219
7.1.1	Variable Documentation	219
7.1.1.1	BFAppLinkVersion	219

7.2	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLinkNavigation.h File Reference	219
7.2.1	Function Documentation	220
7.2.1.1	NS_ENUM	220
7.3	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLinkResolving.h File Reference	220
7.4	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLinkReturnToRefererController.h File Reference	220
7.5	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLinkReturnToRefererView.h File Reference	220
7.5.1	Typedef Documentation	221
7.5.1.1	BFIncludeStatusBarInSize	221
7.5.2	Enumeration Type Documentation	221
7.5.2.1	BFIncludeStatusBarInSize	221
7.6	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFAppLinkTarget.h File Reference	221
7.7	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFExecutor.h File Reference	221
7.8	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFMeasurementEvent.h File Reference	222
7.8.1	Variable Documentation	222
7.8.1.1	BFAppLinkNavigateBackToReferrerEventName	222
7.8.1.2	BFAppLinkNavigateInEventName	222
7.8.1.3	BFAppLinkNavigateOutEventName	222
7.8.1.4	BFAppLinkParseEventName	222
7.8.1.5	BFMeasurementEventArgsKey	223
7.8.1.6	BFMeasurementEventNameKey	223
7.8.1.7	BFMeasurementEventNotificationName	223
7.9	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFTask.h File Reference	223
7.9.1	Typedef Documentation	223
7.9.1.1	BFContinuationBlock	223
7.9.2	Variable Documentation	223
7.9.2.1	BFTaskErrorDomain	223
7.9.2.2	BFTaskMultipleExceptionsException	224
7.10	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFTaskCompletionSource.h File Reference	224
7.11	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFURL.h File Reference	224
7.12	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BFWebViewAppLinkResolver.h File Reference	224
7.13	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/Bolts.h File Reference	224
7.13.1	Variable Documentation	225

7.13.1.1	kBFMultipleErrorsError	225
7.14	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔ Headers/BoltsVersion.h File Reference	225
7.14.1	Macro Definition Documentation	225
7.14.1.1	BOLTS_VERSION	225
7.15	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ AppDelegate.h File Reference	225
7.16	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ AppDelegate.m File Reference	225
7.17	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ Game.h File Reference	225
7.17.1	Variable Documentation	226
7.17.1.1	HighScoreNumber	226
7.17.1.2	Left	226
7.17.1.3	randomLongObstacle1Placement	226
7.17.1.4	randomLongObstacle2Placement	226
7.17.1.5	randomLongObstacle3Placement	226
7.17.1.6	randomLongObstacle4Placement	226
7.17.1.7	Right	226
7.17.1.8	ScoreNumber	227
7.18	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ Game.m File Reference	227
7.18.1	Variable Documentation	227
7.18.1.1	ballSpeed	227
7.18.1.2	defaultBallYCoord	227
7.18.1.3	longObstacleWidth	227
7.18.1.4	numberOfObstaclesOnScreen	228
7.18.1.5	obstacleDistance	228
7.18.1.6	obstacleGap	228
7.18.1.7	obstacleGeneratorBuffer	228
7.18.1.8	obstacleSpeed	228
7.18.1.9	recoverySpeed	228
7.18.1.10	screenBottom	228
7.18.1.11	screenLeft	228
7.18.1.12	screenRight	228
7.18.1.13	screenTop	228
7.19	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔ BoardViewController.h File Reference	229
7.20	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔ BoardViewController.m File Reference	229
7.21	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ LoginViewController.h File Reference	229

7.22	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ LoginViewController.m File Reference	229
7.23	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/main.m File Reference	229
7.23.1	Function Documentation	230
7.23.1.1	main	230
7.24	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ ProfileViewController.h File Reference	230
7.25	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ ProfileViewController.m File Reference	230
7.26	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ ViewController.h File Reference	230
7.27	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ ViewController.m File Reference	230
7.28	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/↔ Tests/Bound_BreakerTests.m File Reference	230
7.29	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/Parse.h File Reference	231
7.30	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PF_Twitter.h File Reference	231
7.31	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFACL.h File Reference	231
7.32	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFAnalytics.h File Reference	232
7.33	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFAnonymousUtils.h File Reference	232
7.34	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFCloud.h File Reference	232
7.35	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFConfig.h File Reference	232
7.35.1	Typedef Documentation	233
7.35.1.1	PFConfigResultBlock	233
7.36	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFConstants.h File Reference	233
7.36.1	Macro Definition Documentation	234
7.36.1.1	PARSE_DEPRECATED	234
7.36.1.2	PARSE_IOS_ONLY	234
7.36.1.3	PARSE_OSX_ONLY	234
7.36.1.4	PARSE_VERSION	234
7.36.2	Typedef Documentation	235
7.36.2.1	PFArrayResultBlock	235
7.36.2.2	PFBooleanResultBlock	235
7.36.2.3	PFDataResultBlock	235
7.36.2.4	PFDataStreamResultBlock	235

7.36.2.5	PFIdResultBlock	235
7.36.2.6	PFIntegerResultBlock	235
7.36.2.7	PFOBJECTResultBlock	235
7.36.2.8	PFProgressBlock	235
7.36.2.9	PFSetResultBlock	235
7.36.2.10	PFStringResultBlock	235
7.36.2.11	PFUserResultBlock	235
7.36.3	Function Documentation	236
7.36.3.1	NS_ENUM	236
7.36.3.2	NS_ENUM	236
7.36.3.3	NS_ENUM	237
7.36.4	Variable Documentation	238
7.36.4.1	kPFDeviceType	238
7.36.4.2	kPFParseServer	238
7.36.4.3	NSColor	238
7.36.4.4	NSImage	239
7.36.4.5	NSView	239
7.36.4.6	PARSE_API_VERSION	239
7.36.4.7	PFParseErrorDomain	239
7.37	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFFile.h File Reference	239
7.38	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFGeoPoint.h File Reference	239
7.38.1	Typedef Documentation	239
7.38.1.1	PFGeoPointResultBlock	239
7.39	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFInstallation.h File Reference	240
7.40	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFNetworkActivityIndicatorManager.h File Reference	240
7.41	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFNullability.h File Reference	240
7.41.1	Macro Definition Documentation	240
7.41.1.1	PF_ASSUME_NONNULL_BEGIN	240
7.41.1.2	PF_ASSUME_NONNULL_END	240
7.41.1.3	PF_NONNULL	241
7.41.1.4	PF_NONNULL_S	241
7.41.1.5	PF_NULLABLE	241
7.41.1.6	PF_NULLABLE_PROPERTY	241
7.41.1.7	PF_NULLABLE_S	241
7.42	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↵ Headers/PFObject+Subclass.h File Reference	241

7.43	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFObject.h File Reference	241
7.43.1	Variable Documentation	242
7.43.1.1	PFObjectDefaultPin	242
7.44	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFProduct.h File Reference	242
7.45	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFPurchase.h File Reference	242
7.46	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFPush.h File Reference	242
7.47	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFQuery.h File Reference	243
7.48	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFRelation.h File Reference	243
7.49	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFRole.h File Reference	243
7.50	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFSession.h File Reference	243
7.50.1	Typedef Documentation	244
7.50.1.1	PFSessionResultBlock	244
7.51	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFSubclassing.h File Reference	244
7.52	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFTwitterUtils.h File Reference	244
7.53	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFUser.h File Reference	244
7.53.1	Typedef Documentation	245
7.53.1.1	PFUserLogoutResultBlock	245
7.53.2	Variable Documentation	245
7.53.2.1	PFUserSessionUpgradeResultBlock	245
7.54	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseCrash↔ Reporting.framework/Headers/ParseCrashReporting.h File Reference	245
7.55	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseFacebook↔ Utils.framework/Headers/PFFacebookUtils.h File Reference	245
7.56	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseFacebook↔ UtilsV4.framework/Headers/PFFacebookUtils.h File Reference	246
7.57	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/ParseUI.h File Reference	246
7.58	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/ParseUIConstants.h File Reference	246
7.58.1	Macro Definition Documentation	247
7.58.1.1	PARSE_UI_DEPRECATED	247
7.58.1.2	PFUI_ASSUME_NONNULL_BEGIN	247
7.58.1.3	PFUI_ASSUME_NONNULL_END	247
7.58.1.4	PFUI_NULL_UNSPECIFIED	247

7.58.1.5	PFUI_NULLABLE	247
7.58.1.6	PFUI_NULLABLE_PROPERTY	247
7.58.1.7	PFUI_NULLABLE_S	247
7.59	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFCollectionViewCell.h File Reference	247
7.60	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFImageView.h File Reference	248
7.60.1	Variable Documentation	248
7.60.1.1	PFImageViewImageResultBlock	248
7.61	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFLoginView.h File Reference	248
7.61.1	Function Documentation	248
7.61.1.1	NS_OPTIONS	248
7.62	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFLoginViewController.h File Reference	249
7.62.1	Variable Documentation	249
7.62.1.1	PFLoginCancelNotification	249
7.62.1.2	PFLoginFailureNotification	249
7.62.1.3	PFLoginSuccessNotification	249
7.63	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFProductTableViewController.h File Reference	250
7.64	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFPurchaseTableViewCell.h File Reference	250
7.64.1	Function Documentation	250
7.64.1.1	NS_ENUM	250
7.65	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFQueryCollectionViewController.h File Reference	250
7.66	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFQueryTableViewController.h File Reference	251
7.67	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFSignUpView.h File Reference	251
7.67.1	Function Documentation	251
7.67.1.1	NS_OPTIONS	251
7.68	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFSignUpViewController.h File Reference	252
7.68.1	Variable Documentation	252
7.68.1.1	PFSignUpCancelNotification	252
7.68.1.2	PFSignUpFailureNotification	252
7.68.1.3	PFSignUpSuccessNotification	252
7.69	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFTableViewCell.h File Reference	252
7.70	/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↵ Headers/PFTextField.h File Reference	253
7.70.1	Function Documentation	253

7.70.1.1	NS_OPTIONS	253
7.71	/Users/jscottrichards/Documents/GitHub/BoundBreaker/README.md File Reference	253
Index		255

Chapter 1

Bound Breaker

An addiciting iPhone Application

Contributors

- Victor Palacios
- Matt Schroeder
- Brent Pivnik
- John Richards

Feature Tracking

<https://trello.com/b/X540gviw/boundbreaker>

Requirements

Xcode

Chapter 2

Deprecated List

Member [\[PFFacebookUtils handleOpenURL:\]](#)

Please use '[FBAppCall handleOpenURL:url sourceApplication:sourceApplication withSession:[[PFFacebookUtils](#) session]]'; instead.

Member [\[PFFacebookUtils initializeWithApplicationId:\]](#)

Please use [[PFFacebookUtils](#) initializeFacebook] instead.

Member [\[PFFacebookUtils initializeWithApplicationId:urlSchemeSuffix:\]](#)

Please use [[PFFacebookUtils](#) initializeFacebookWithURLShemeSuffix:] instead.

Member [\[PFFacebookUtils linkUser:permissions:\]](#)

Please use [[PFFacebookUtils](#) linkUserInBackground:permissions:] instead.

Member [\[PFLoginView __attribute__\]](#)

This property is deprecated and will always be nil.

This property is deprecated and will always be nil.

This property is deprecated and will always be nil.

Member [\[PFObject relationForKey:\]](#)

Please use [[PFObject](#) relationForKey:] instead.

Member [\[PFPush setPushToAndroid:\]](#)

Please use a [[PFInstallation](#) query] with a constraint on deviceType instead.

Member [\[PFPush setPushToIOS:\]](#)

Please use a [[PFInstallation](#) query] with a constraint on deviceType instead.

Member [\[PFQuery PARSE_DEPRECATED\]](#)

Please use [[PFUser](#) query] instead.

Member [\[PFTwitterUtils linkUser:\]](#)

Please use [[PFTwitterUtils](#) linkUserInBackground:] instead.

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AppDelegate()	15
Game()	43
leaderBoardViewController()	46
LoginViewController()	48
<NSCoding>	
PFACL	53
PFGeoPoint	89
<NSCopying>	
PFACL	53
PFGeoPoint	89
PFPush	146
PFQuery	157
NSObject	
BFAppLink	16
BFAppLinkNavigation	17
BFAppLinkReturnToRefererController	21
BFAppLinkTarget	25
BFExecutor	26
BFMeasurementEvent	28
BFTask	28
BFTaskCompletionSource	33
BFURL	35
BFWebViewAppLinkResolver	37
Bolts	38
Parse	48
ParseCrashReporting	50
PF_Twitter	51
PFACL	53
PFAnalytics	59
PFAnonymousUtils	62
PFCloud	64
PFConfig	68
PFFacebookUtils	70
PFFacebookUtils	70
PFFile	82
PFGeoPoint	89
PFNetworkActivityIndicatorManager	103
PFObject	104

PFInstallation	94
PFProduct	140
PFRole	188
PFSession	191
PFUser	207
PFPurchase	143
PFPush	146
PFQuery	157
PFRelation	187
PFTwitterUtils	201
<NSObject>	
<BFAppLinkResolving>	19
BFWebViewAppLinkResolver	37
<BFAppLinkReturnToRefererControllerDelegate>	22
<BFAppLinkReturnToRefererViewDelegate>	24
BFAppLinkReturnToRefererController	21
<PFLoginViewControllerDelegate>	101
<PFSignUpViewControllerDelegate>	196
PFObjct(Subclass)	138
<PFSubclassing>	197
PFInstallation	94
PFProduct	140
PFRole	188
PFSession	191
PFUser	207
ProfileViewController()	217
<UIApplicationDelegate>	
AppDelegate	15
UICollectionViewCell	
PFCollectionViewCell	66
UICollectionViewController	
PFQueryCollectionViewController	178
<UICollectionViewDelegateFlowLayout>	
PFQueryCollectionViewController	178
UIImageView	
PFImageView	93
UIResponder	
AppDelegate	15
UIScrollView	
PFLoginView	96
PFSignUpView	192
<UIScrollViewDelegate>	
PFSignUpViewController	194
UITableViewController	
PFTableViewCell	199
PFPurchaseTableViewCell	144
UITableViewController	
PFQueryTableViewController	182
PFProductTableViewController	142
<UITableViewDataSource>	
PFQueryTableViewController	182
<UITableViewDelegate>	
PFQueryTableViewController	182
UITextField	
PFTextField	200
<UITextFieldDelegate>	
PFLoginViewController	99
PFSignUpViewController	194

UIView	
BFAppLinkReturnToRefererView	23
UIViewController	
Game	39
leaderBoardViewController	44
LoginViewController	46
PFLoginViewController	99
PFSignUpViewController	194
ProfileViewController	215
ViewController	217
ViewController()	218
XCTestCase	
Bound_BreakerTests	38

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AppDelegate	15
AppDelegate()	15
BFAppLink	16
BFAppLinkNavigation	17
<BFAppLinkResolving>	19
BFAppLinkReturnToRefererController	21
<BFAppLinkReturnToRefererControllerDelegate>	22
BFAppLinkReturnToRefererView	23
<BFAppLinkReturnToRefererViewDelegate>	24
BFAppLinkTarget	25
BFExecutor	26
BFMeasurementEvent	28
BFTask	28
BFTaskCompletionSource	33
BFURL	35
BFWebViewAppLinkResolver	37
Bolts	38
Bound_BreakerTests	38
Game	39
Game()	
Method file for the Game view. Implements game and player mechanisms	43
leaderBoardViewController	44
leaderBoardViewController()	46
LoginViewController	46
LoginViewController()	
This is the method file for LoginViewController.m . implement code for logging in and creating an account	48
Parse	48
ParseCrashReporting	50
PF_Twitter	51
PFACL	53
PFAnalytics	59
PFAnonymousUtils	62
PFCloud	64
PFCollectionViewCell	66
PFConfig	68
PFFacebookUtils	70
PFFile	82

PFGeoPoint	89
PFImageView	93
PFInstallation	94
PFLoginView	96
PFLoginViewController	99
<PFLoginViewControllerDelegate>	101
PFNetworkActivityIndicatorManager	103
PFObject	104
PFObject(Subclass)	138
PFProduct	140
PFProductTableViewController	142
PFPurchase	143
PFPurchaseTableViewCell	144
PFPush	146
PFQuery	157
PFQueryCollectionViewController	178
PFQueryTableViewController	182
PFRelation	187
PFRole	188
PFSession	191
PFSignUpView	192
PFSignUpViewController	194
<PFSignUpViewControllerDelegate>	196
<PFSubclassing>	197
PFTableViewCell	199
PFTextField	200
PFTwitterUtils	201
PFUser	207
ProfileViewController	215
ProfileViewController()	
This is the method file for ProfileViewController.m . Implement codes for viewing account info	217
ViewController	217
ViewController()	218

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLink.h	219
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLinkNavigation.h	219
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLinkResolving.h	220
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLinkReturnToRefererController.h	220
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLinkReturnToRefererView.h	220
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFAppLinkTarget.h	221
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFExecutor.h	221
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFMeasurementEvent.h	222
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFTask.h	223
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFTaskCompletionSource.h	224
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFURL.h	224
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BFWebViewAppLinkResolver.h	224
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/Bolts.h	224
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/↔	
Headers/BoltsVersion.h	225
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/App↔	
Delegate.h	225
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/App↔	
Delegate.m	225
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Game.h .	225
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Game.m	227
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔	
BoardViewController.h	229
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔	
BoardViewController.m	229

/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Login↔ ViewController.h	229
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Login↔ ViewController.m	229
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/main.m	229
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Profile↔ ViewController.h	230
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Profile↔ ViewController.m	230
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/View↔ Controller.h	230
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/View↔ Controller.m	230
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker↔ Tests/Bound_BreakerTests.m	230
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/Parse.h	231
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PF_Twitter.h	231
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFACL.h	231
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFAnalytics.h	232
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFAnonymousUtils.h	232
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFCloud.h	232
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFConfig.h	232
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFConstants.h	233
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFFile.h	239
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFGeoPoint.h	239
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFInstallation.h	240
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFNetworkActivityIndicatorManager.h	240
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFNullability.h	240
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFObject+Subclass.h	241
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFObject.h	241
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFProduct.h	242
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFPurchase.h	242
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFPush.h	242
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFQuery.h	243
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFRelation.h	243
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/↔ Headers/PFRole.h	243

/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/Parse.framework/↔	
Headers/PFSession.h		243
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/Parse.framework/↔	
Headers/PFSubclassing.h		244
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/Parse.framework/↔	
Headers/PFTwitterUtils.h		244
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/Parse.framework/↔	
Headers/PFUser.h		244
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseCrashReporting.↔	
framework/Headers/ParseCrashReporting.h		245
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseFacebookUtils.↔	
framework/Headers/PFFacebookUtils.h		245
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseFacebookUtilsV4.↔	
framework/Headers/PFFacebookUtils.h		246
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/ParseUI.h		246
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/ParseUIConstants.h		246
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFCollectionViewCell.h		247
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFImageView.h		248
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFLoginView.h		248
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFLoginViewController.h		249
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFProductTableViewController.h		250
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFPurchaseTableViewCell.h		250
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFQueryCollectionViewController.h		250
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFQueryTableViewController.h		251
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFSignUpView.h		251
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFSignUpViewController.h		252
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFTableViewCell.h		252
/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound	Breaker/ParseUI.framework/↔	
Headers/PFTextField.h		253

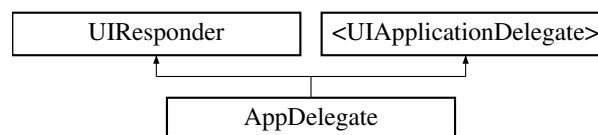
Chapter 6

Class Documentation

6.1 AppDelegate Class Reference

```
#import <AppDelegate.h>
```

Inheritance diagram for AppDelegate:



Properties

- UIWindow * [window](#)

6.1.1 Detailed Description

Definition at line 11 of file AppDelegate.h.

6.1.2 Property Documentation

6.1.2.1 - (UIWindow*) [window](#) [read], [write], [nonatomic], [strong]

Definition at line 13 of file AppDelegate.h.

The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/[AppDelegate.h](#)

6.2 AppDelegate() Category Reference

6.2.1 Detailed Description

Definition at line 12 of file AppDelegate.m.

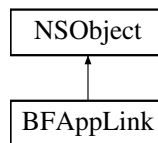
The documentation for this category was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/AppDelegate.m](#)

6.3 BFApplink Class Reference

```
#import <BFApplink.h>
```

Inheritance diagram for BFApplink:



Class Methods

- (instancetype) + [applinkWithSourceURL:targets:webURL:](#)

Properties

- NSURL * [sourceURL](#)
- NSArray * [targets](#)
- NSURL * [webURL](#)

6.3.1 Detailed Description

Contains App Link metadata relevant for navigation on this device derived from the HTML at a given URL.

Definition at line 20 of file BFApplink.h.

6.3.2 Method Documentation

6.3.2.1 + (instancetype) [applinkWithSourceURL:](#) (NSURL *) *sourceURL* targets:(NSArray *) *targets* webURL:(NSURL *) *webURL*

Creates a [BFApplink](#) with the given list of BFApplinkTargets and target URL.

Generally, this will only be used by implementers of the [BFApplinkResolving](#) protocol, as these implementers will produce App Link metadata for a given URL.

Parameters

<i>sourceURL</i>	the URL from which this App Link is derived
<i>targets</i>	an ordered list of BFApplinkTargets for this platform derived from App Link metadata.
<i>webURL</i>	the fallback web URL, if any, for the app link.

6.3.3 Property Documentation

6.3.3.1 - (NSURL*) [sourceURL](#) [read], [nonatomic], [strong]

The URL from which this [BFApplink](#) was derived

Definition at line 38 of file BFApplink.h.

6.3.3.2 - (NSArray*) targets [read], [nonatomic], [copy]

The ordered list of targets applicable to this platform that will be used for navigation.

Definition at line 44 of file BFAppLink.h.

6.3.3.3 - (NSURL*) webURL [read], [nonatomic], [strong]

The fallback web URL to use if no targets are installed on this device.

Definition at line 47 of file BFAppLink.h.

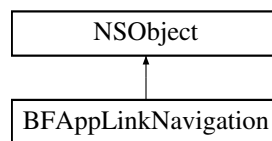
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/B↵
FAppLink.h

6.4 BFAppLinkNavigation Class Reference

```
#import <BFAppLinkNavigation.h>
```

Inheritance diagram for BFAppLinkNavigation:



Instance Methods

- (BFAppLinkNavigationType) - [navigate:](#)

Class Methods

- (instancetype) + [navigationWithAppLink:extras:appLinkData:](#)
- (BFTask *) + [resolveAppLinkInBackground:](#)
- (BFTask *) + [resolveAppLinkInBackground:resolver:](#)
- (BFAppLinkNavigationType) + [navigateToAppLink:error:](#)
- (BFTask *) + [navigateToURLInBackground:](#)
- (BFTask *) + [navigateToURLInBackground:resolver:](#)
- (id< BFAppLinkResolving >) + [defaultResolver](#)
- (void) + [setDefaultResolver:](#)

Properties

- NSDictionary * [extras](#)
- NSDictionary * [appLinkData](#)
- BFAppLink * [appLink](#)

6.4.1 Detailed Description

Represents a pending request to navigate to an App Link. Most developers will simply use `navigateToURLInBackground:` to open a URL, but developers can build custom requests with additional navigation and app data attached to them by creating `BFAppLinkNavigations` themselves.

Definition at line 36 of file `BFAppLinkNavigation.h`.

6.4.2 Method Documentation

6.4.2.1 `+(id<BFAppLinkResolving>) defaultResolver`

Gets the default resolver to be used for App Link resolution. If the developer has not set one explicitly, a basic, built-in resolver will be used.

6.4.2.2 `-(BFAppLinkNavigationType) navigate: (NSError **) error`

Performs the navigation

6.4.2.3 `+(BFAppLinkNavigationType) navigateToAppLink: (BFAppLink *) link error:(NSError **) error`

Navigates to a [BFAppLink](#) and returns whether it opened in-app or in-browser

6.4.2.4 `+(BFTask *) navigateToURLInBackground: (NSURL *) destination`

Navigates to a URL (an asynchronous action) and returns a `BFNavigationType`

6.4.2.5 `+(BFTask *) navigateToURLInBackground: (NSURL *) destination resolver:(id< BFAppLinkResolving >) resolver`

Navigates to a URL (an asynchronous action) using the given App Link resolution strategy and returns a `BFNavigationType`

6.4.2.6 `+(instancetype) navigationWithAppLink: (BFAppLink *) appLink extras:(NSDictionary *) extras appLinkData:(NSDictionary *) appLinkData`

Creates an `AppLinkNavigation` with the given link, extras, and App Link data

6.4.2.7 `+(BFTask *) resolveAppLinkInBackground: (NSURL *) destination`

Returns a [BFAppLink](#) for the given URL

6.4.2.8 `+(BFTask *) resolveAppLinkInBackground: (NSURL *) destination resolver:(id< BFAppLinkResolving >) resolver`

Returns a [BFAppLink](#) for the given URL using the given App Link resolution strategy

6.4.2.9 `+(void) setDefaultResolver: (id< BFAppLinkResolving >) resolver`

Sets the default resolver to be used for App Link resolution. Setting this to nil will revert the default resolver to the basic, built-in resolver provided by [Bolts](#).

6.4.3 Property Documentation

6.4.3.1 -(BFAppLink*) appLink [read],[nonatomic],[strong]

The AppLink to navigate to

Definition at line 53 of file BFAppLinkNavigation.h.

6.4.3.2 -(NSDictionary*) appLinkData [read],[nonatomic],[copy]

The al_appLink_data for the AppLinkNavigation. This will generally contain data common to navigation attempts such as back-links, user agents, and other information that may be used in routing and handling an App Link request.

Definition at line 50 of file BFAppLinkNavigation.h.

6.4.3.3 -(NSDictionary*) extras [read],[nonatomic],[copy]

The extras for the AppLinkNavigation. This will generally contain application-specific data that should be passed along with the request, such as advertiser or affiliate IDs or other such metadata relevant on this device.

Definition at line 43 of file BFAppLinkNavigation.h.

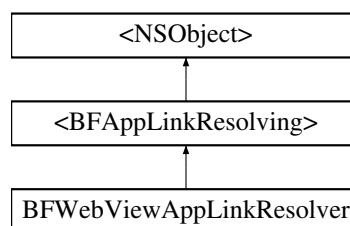
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/B↵
FAppLinkNavigation.h

6.5 <BFAppLinkResolving> Protocol Reference

```
#import <BFAppLinkResolving.h>
```

Inheritance diagram for <BFAppLinkResolving>:



Instance Methods

- (BFTask *) - [appLinkFromURLInBackground:](#)

6.5.1 Detailed Description

Implement this protocol to provide an alternate strategy for resolving App Links that may include pre-fetching, caching, or querying for App Link data from an index provided by a service provider.

Definition at line 20 of file BFAppLinkResolving.h.

6.5.2 Method Documentation

6.5.2.1 -(BFTask *) appLinkFromURLInBackground: (NSURL *) *url*

Asynchronously resolves App Link data for a given URL.

Parameters

<i>url</i>	The URL to resolve into an App Link.
------------	--------------------------------------

Returns

A [BFTask](#) that will return a [BFAppLink](#) for the given URL.

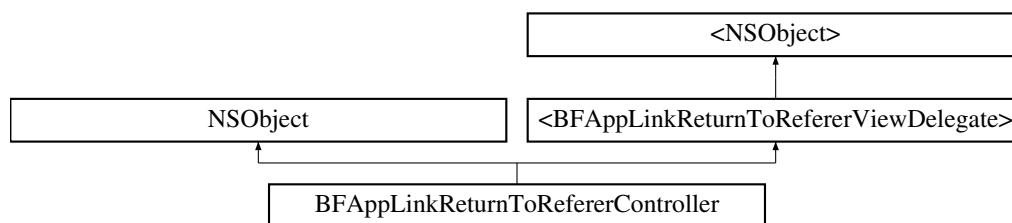
The documentation for this protocol was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFAppLinkResolving.h](#)

6.6 BFAppLinkReturnToRefererController Class Reference

```
#import <BFAppLinkReturnToRefererController.h>
```

Inheritance diagram for BFAppLinkReturnToRefererController:



Instance Methods

- (instancetype) - [init](#)
- (instancetype) - [initWithDisplayAboveNavController:](#)
- (void) - [removeFromNavController](#)
- (void) - [showViewForRefererAppLink:](#)
- (void) - [showViewForRefererURL:](#)
- (void) - [closeViewAnimated:](#)

Properties

- id< [BFAppLinkReturnToRefererControllerDelegate](#) > [delegate](#)
- [BFAppLinkReturnToRefererView](#) * [view](#)

6.6.1 Detailed Description

A controller class that implements default behavior for a [BFAppLinkReturnToRefererView](#), including the ability to display the view above the navigation bar for navigation-based apps.

Definition at line 43 of file [BFAppLinkReturnToRefererController.h](#).

6.6.2 Method Documentation

6.6.2.1 -(void) closeViewAnimated: (BOOL) animated

Closes the view, possibly animating it.

6.6.2.2 - (instancetype) init

Initializes a controller suitable for controlling a [BFAppLinkReturnToRefererView](#) that is to be displayed contained within another UIView (i.e., not displayed above the navigation bar).

6.6.2.3 - (instancetype) initWithDisplayAboveNavController: (UINavigationController *) navCtrl

Initializes a controller suitable for controlling a [BFAppLinkReturnToRefererView](#) that is to be displayed above the navigation bar.

6.6.2.4 - (void) removeFromNavController

Removes the view entirely from the navigation controller it is currently displayed in.

6.6.2.5 - (void) showViewForRefererAppLink: (BFAppLink *) refererAppLink

Shows the [BFAppLinkReturnToRefererView](#) with the specified referer information. If nil or missing data, the view will not be displayed.

6.6.2.6 - (void) showViewForRefererURL: (NSURL *) url

Shows the [BFAppLinkReturnToRefererView](#) with referer information extracted from the specified URL. If nil or missing referer App Link data, the view will not be displayed.

6.6.3 Property Documentation

6.6.3.1 - (id<BFAppLinkReturnToRefererControllerDelegate>) delegate [read], [write], [nonatomic], [weak]

The delegate that will be notified when the user navigates back to the referer.

Definition at line 48 of file [BFAppLinkReturnToRefererController.h](#).

6.6.3.2 - (BFAppLinkReturnToRefererView*) view [read], [write], [nonatomic], [strong]

The [BFAppLinkReturnToRefererView](#) this controller is controlling.

Definition at line 53 of file [BFAppLinkReturnToRefererController.h](#).

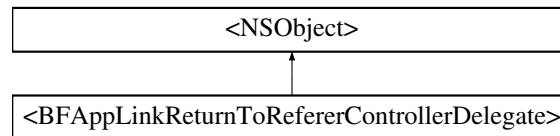
The documentation for this class was generated from the following file:

- [/Users/jscootrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFAppLinkReturnToRefererController.h](#)

6.7 <BFAppLinkReturnToRefererControllerDelegate> Protocol Reference

```
#import <BFAppLinkReturnToRefererController.h>
```

Inheritance diagram for <BFAppLinkReturnToRefererControllerDelegate>:



Instance Methods

- (void) - [returnToRefererController:willNavigateToAppLink:](#)
- (void) - [returnToRefererController:didNavigateToAppLink:type:](#)

6.7.1 Detailed Description

Protocol that a class can implement in order to be notified when the user has navigated back to the referer of an App Link.

Definition at line 23 of file BFApplinkReturnToRefererController.h.

6.7.2 Method Documentation

6.7.2.1 - (void) [returnToRefererController:](#) ([BFApplinkReturnToRefererController *](#)) *controller*
[didNavigateToAppLink:](#)([BFApplink *](#)) *url* [type:](#)([BFApplinkNavigationType](#)) *type* [optional]

Called after the navigation has been attempted, with an indication of whether the referer app link was successfully opened.

6.7.2.2 - (void) [returnToRefererController:](#) ([BFApplinkReturnToRefererController *](#)) *controller*
[willNavigateToAppLink:](#)([BFApplink *](#)) *appLink* [optional]

Called when the user has tapped to navigate, but before the navigation has been performed.

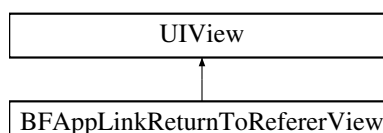
The documentation for this protocol was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFApplinkReturnToRefererController.h](#)

6.8 BFApplinkReturnToRefererView Class Reference

```
#import <BFApplinkReturnToRefererView.h>
```

Inheritance diagram for BFApplinkReturnToRefererView:



Properties

- id < [BFApplinkReturnToRefererViewDelegate](#) > *delegate*
- UIColor * *textColor*

- [BFAppLink](#) * [refererAppLink](#)
- [BFIncludeStatusBarInSize](#) [includeStatusBarInSize](#)
- `BOOL` [closed](#)

6.8.1 Detailed Description

Provides a `UIView` that displays a button allowing users to navigate back to the application that launched the App Link currently being handled, if the App Link contained referer data. The user can also close the view by clicking a close button rather than navigating away. If the view is provided an App Link that does not contain referer data, it will have zero size and no UI will be displayed.

Definition at line 51 of file `BFAppLinkReturnToRefererView.h`.

6.8.2 Property Documentation

6.8.2.1 `-(BOOL) closed` `[read], [write], [nonatomic], [assign]`

Indicates whether the user has closed the view by clicking the close button.

Definition at line 75 of file `BFAppLinkReturnToRefererView.h`.

6.8.2.2 `-(id<BFAppLinkReturnToRefererViewDelegate>) delegate` `[read], [write], [nonatomic], [weak]`

The delegate that will be notified when the user navigates back to the referer.

Definition at line 56 of file `BFAppLinkReturnToRefererView.h`.

6.8.2.3 `-(BFIncludeStatusBarInSize) includeStatusBarInSize` `[read], [write], [nonatomic], [assign]`

Indicates whether to extend the size of the view to include the current status bar size, for use in scenarios where the view might extend under the status bar on iOS 7 and above; this property has no effect on earlier versions of iOS.

Definition at line 70 of file `BFAppLinkReturnToRefererView.h`.

6.8.2.4 `-(BFAppLink*) refererAppLink` `[read], [write], [nonatomic], [strong]`

Definition at line 63 of file `BFAppLinkReturnToRefererView.h`.

6.8.2.5 `-(UIColor*) textColor` `[read], [write], [nonatomic], [strong]`

The color of the text label and close button.

Definition at line 61 of file `BFAppLinkReturnToRefererView.h`.

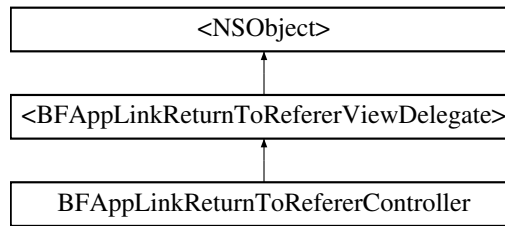
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFAppLinkReturnToRefererView.h`

6.9 <BFAppLinkReturnToRefererViewDelegate> Protocol Reference

```
#import <BFAppLinkReturnToRefererView.h>
```

Inheritance diagram for <BFAppLinkReturnToRefererViewDelegate>:



Instance Methods

- (void) - [returnToRefererViewDidTapInsideCloseButton:](#)
- (void) - [returnToRefererViewDidTapInsideLink:link:](#)

6.9.1 Detailed Description

Protocol that a class can implement in order to be notified when the user has navigated back to the referer of an App Link.

Definition at line 29 of file BFApplinkReturnToRefererView.h.

6.9.2 Method Documentation

6.9.2.1 - (void) [returnToRefererViewDidTapInsideCloseButton:](#) ([BFApplinkReturnToRefererView *](#)) *view*

Called when the user has tapped inside the close button.

6.9.2.2 - (void) [returnToRefererViewDidTapInsideLink:](#) ([BFApplinkReturnToRefererView *](#)) *view* [link:\(BFApplink *\) link](#)

Called when the user has tapped inside the App Link portion of the view.

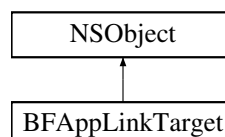
The documentation for this protocol was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFApplinkReturnToRefererView.h](#)

6.10 BFApplinkTarget Class Reference

```
#import <BFApplinkTarget.h>
```

Inheritance diagram for BFApplinkTarget:



Class Methods

- (instancetype) + [applinkTargetWithURL:appStoreId:appName:](#)

Properties

- NSURL * [URL](#)
- NSString * [appStoreId](#)
- NSString * [appName](#)

6.10.1 Detailed Description

Represents a target defined in App Link metadata, consisting of at least a URL, and optionally an App Store ID and name.

Definition at line 17 of file BFApplinkTarget.h.

6.10.2 Method Documentation

6.10.2.1 **+** (instancetype) [applinkTargetWithURL:](#) (NSURL *) *url* appStoreId:(NSString *) *appStoreId* appName:(NSString *) *appName*

Creates a [BFApplinkTarget](#) with the given app site and target URL.

6.10.3 Property Documentation

6.10.3.1 **-** (NSString*) [appName](#) [read], [nonatomic], [copy]

The name of the app

Definition at line 31 of file BFApplinkTarget.h.

6.10.3.2 **-** (NSString*) [appStoreId](#) [read], [nonatomic], [copy]

The app ID for the app store

Definition at line 28 of file BFApplinkTarget.h.

6.10.3.3 **-** (NSURL*) [URL](#) [read], [nonatomic], [strong]

The URL prefix for this app link target

Definition at line 25 of file BFApplinkTarget.h.

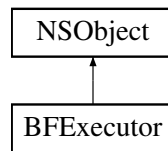
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/[BFApplinkTarget.h](#)

6.11 BFApplinkTarget Class Reference

```
#import <BFApplinkTarget.h>
```

Inheritance diagram for BFApplinkTarget:



Instance Methods

- (void) - [execute](#):

Class Methods

- (instancetype) + [defaultExecutor](#)
- (instancetype) + [immediateExecutor](#)
- (instancetype) + [mainThreadExecutor](#)
- (instancetype) + [executorWithBlock](#):
- (instancetype) + [executorWithDispatchQueue](#):
- (instancetype) + [executorWithOperationQueue](#):

6.11.1 Detailed Description

An object that can run a given block.

Definition at line 16 of file BFExecutor.h.

6.11.2 Method Documentation

6.11.2.1 + (instancetype) defaultExecutor

Returns a default executor, which runs continuations immediately until the call stack gets too deep, then dispatches to a new GCD queue.

6.11.2.2 - (void) execute: (void(^)) block

Runs the given block using this executor's particular strategy.

Parameters

<i>block</i>	The block to execute.
--------------	-----------------------

6.11.2.3 + (instancetype) executorWithBlock: (void(^)(void(^ block))) block

Returns a new executor that uses the given block to execute continuations.

Parameters

<i>block</i>	The block to use.
--------------	-------------------

6.11.2.4 + (instancetype) executorWithDispatchQueue: (dispatch_queue_t) queue

Returns a new executor that runs continuations on the given queue.

Parameters

<i>queue</i>	The instance of <code>dispatch_queue_t</code> to dispatch all continuations onto.
--------------	---

6.11.2.5 + (instancetype) `executorWithOperationQueue:` (NSOperationQueue *) *queue*

Returns a new executor that runs continuations on the given queue.

Parameters

<i>queue</i>	The instance of <code>NSOperationQueue</code> to run all continuations on.
--------------	--

6.11.2.6 + (instancetype) `immediateExecutor`

Returns an executor that runs continuations on the thread where the previous task was completed.

6.11.2.7 + (instancetype) `mainThreadExecutor`

Returns an executor that runs continuations on the main thread.

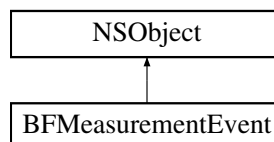
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFExecutor.h](#)

6.12 BFMeasurementEvent Class Reference

```
#import <BFMeasurementEvent.h>
```

Inheritance diagram for `BFMeasurementEvent`:



6.12.1 Detailed Description

Definition at line 44 of file `BFMeasurementEvent.h`.

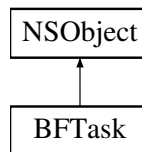
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFMeasurementEvent.h](#)

6.13 BFTask Class Reference

```
#import <BFTask.h>
```

Inheritance diagram for `BFTask`:



Instance Methods

- (instancetype) - [continueWithBlock:](#)
- (instancetype) - [continueWithExecutor:withBlock:](#)
- (instancetype) - [continueWithSuccessBlock:](#)
- (instancetype) - [continueWithExecutor:withSuccessBlock:](#)
- (void) - [waitUntilFinished](#)

Class Methods

- (instancetype) + [taskWithResult:](#)
- (instancetype) + [taskWithError:](#)
- (instancetype) + [taskWithException:](#)
- (instancetype) + [cancelledTask](#)
- (instancetype) + [taskForCompletionOfAllTasks:](#)
- (instancetype) + [taskForCompletionOfAllTasksWithResults:](#)
- (instancetype) + [taskWithDelay:](#)
- (instancetype) + [taskFromExecutor:withBlock:](#)

Properties

- id [result](#)
- NSError * [error](#)
- NSException * [exception](#)
- BOOL [cancelled](#)
- BOOL [faulted](#)
- BOOL [completed](#)

6.13.1 Detailed Description

The consumer view of a Task. A [BFTask](#) has methods to inspect the state of the task, and to add continuations to be run once the task is complete.

Definition at line 36 of file BFTask.h.

6.13.2 Method Documentation

6.13.2.1 + (instancetype) cancelledTask

Creates a task that is already cancelled.

6.13.2.2 - (instancetype) continueWithBlock: (BFContinuationBlock) block

Enqueues the given block to be run once this task is complete. This method uses a default execution strategy. The block will be run on the thread where the previous task completes, unless the the stack depth is too deep, in which case it will be run on a dispatch queue with default priority.

Parameters

<i>block</i>	The block to be run once this task is complete.
--------------	---

Returns

A task that will be completed after block has run. If block returns a [BFTask](#), then the task returned from this method will not be completed until that task is completed.

6.13.2.3 - (instancetype) continueWithExecutor: (BFXecutor *) executor withBlock:(BFContinuationBlock) block

Enqueues the given block to be run once this task is complete.

Parameters

<i>executor</i>	A BFXecutor responsible for determining how the continuation block will be run.
<i>block</i>	The block to be run once this task is complete.

Returns

A task that will be completed after block has run. If block returns a [BFTask](#), then the task returned from this method will not be completed until that task is completed.

6.13.2.4 - (instancetype) continueWithExecutor: (BFXecutor *) executor withSuccessBlock:(BFContinuationBlock) block

Identical to `continueWithExecutor:withBlock:`, except that the block is only run if this task did not produce a cancellation, error, or exception. If it did, then the failure will be propagated to the returned task.

Parameters

<i>executor</i>	A BFXecutor responsible for determining how the continuation block will be run.
<i>block</i>	The block to be run once this task is complete.

Returns

A task that will be completed after block has run. If block returns a [BFTask](#), then the task returned from this method will not be completed until that task is completed.

6.13.2.5 - (instancetype) continueWithSuccessBlock: (BFContinuationBlock) block

Identical to `continueWithBlock:`, except that the block is only run if this task did not produce a cancellation, error, or exception. If it did, then the failure will be propagated to the returned task.

Parameters

<i>block</i>	The block to be run once this task is complete.
--------------	---

Returns

A task that will be completed after block has run. If block returns a [BFTask](#), then the task returned from this method will not be completed until that task is completed.

6.13.2.6 + (instancetype) taskForCompletionOfAllTasks: (NSArray *) tasks

Returns a task that will be completed (with result == nil) once all of the input tasks have completed.

Parameters

<i>tasks</i>	An <code>NSArray</code> of the tasks to use as an input.
--------------	--

6.13.2.7 + (instancetype) taskForCompletionOfAllTasksWithResults: (NSArray *) tasks

Returns a task that will be completed once all of the input tasks have completed. If all tasks complete successfully without being faulted or cancelled the result will be an `NSArray` of all task results in the order they were provided.

Parameters

<i>tasks</i>	An <code>NSArray</code> of the tasks to use as an input.
--------------	--

6.13.2.8 + (instancetype) taskFromExecutor: (BFTask *) executor withBlock:(id (^)(void)) block

Returns a task that will be completed after the given block completes with the specified executor.

Parameters

<i>executor</i>	A <code>BFTask</code> responsible for determining how the continuation block will be run.
<i>block</i>	The block to immediately schedule to run with the given executor.

Returns

A task that will be completed after block has run. If block returns a `BFTask`, then the task returned from this method will not be completed until that task is completed.

6.13.2.9 + (instancetype) taskWithDelay: (int) millis

Returns a task that will be completed a certain amount of time in the future.

Parameters

<i>millis</i>	The approximate number of milliseconds to wait before the task will be finished (with result == nil).
---------------	---

6.13.2.10 + (instancetype) taskWithError: (NSError *) error

Creates a task that is already completed with the given error.

Parameters

<i>error</i>	The error for the task.
--------------	-------------------------

6.13.2.11 + (instancetype) taskWithException: (NSException *) exception

Creates a task that is already completed with the given exception.

Parameters

<i>exception</i>	The exception for the task.
------------------	-----------------------------

6.13.2.12 + (instancetype) taskWithResult: (id) result

Creates a task that is already completed with the given result.

Parameters

<i>result</i>	The result for the task.
---------------	--------------------------

6.13.2.13 - (void) waitUntilFinished

Waits until this operation is completed. This method is inefficient and consumes a thread resource while it's running. It should be avoided. This method logs a warning message if it is used on the main thread.

6.13.3 Property Documentation

6.13.3.1 - (BOOL) cancelled [read], [nonatomic], [assign]

Whether this task has been cancelled.

Definition at line 117 of file BFTask.h.

6.13.3.2 - (BOOL) completed [read], [nonatomic], [assign]

Whether this task has completed.

Definition at line 127 of file BFTask.h.

6.13.3.3 - (NSError*) error [read], [nonatomic], [strong]

The error of a failed task.

Definition at line 107 of file BFTask.h.

6.13.3.4 - (NSException*) exception [read], [nonatomic], [strong]

The exception of a failed task.

Definition at line 112 of file BFTask.h.

6.13.3.5 - (BOOL) faulted [read], [nonatomic], [assign]

Whether this task has completed due to an error or exception.

Definition at line 122 of file BFTask.h.

6.13.3.6 - (id) result [read], [nonatomic], [strong]

The result of a successful task.

Definition at line 101 of file BFTask.h.

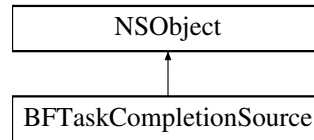
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFTask.h

6.14 BFTaskCompletionSource Class Reference

```
#import <BFTaskCompletionSource.h>
```

Inheritance diagram for BFTaskCompletionSource:



Instance Methods

- (void) - [setResult:](#)
- (void) - [setError:](#)
- (void) - [setException:](#)
- (void) - [cancel](#)
- (BOOL) - [trySetResult:](#)
- (BOOL) - [trySetError:](#)
- (BOOL) - [trySetException:](#)
- (BOOL) - [trySetCancelled](#)

Class Methods

- (instancetype) + [taskCompletionSource](#)

Properties

- [BFTask](#) * [task](#)

6.14.1 Detailed Description

A [BFTaskCompletionSource](#) represents the producer side of tasks. It is a task that also has methods for changing the state of the task by settings its completion values.

Definition at line 20 of file BFTaskCompletionSource.h.

6.14.2 Method Documentation

6.14.2.1 - (void) [cancel](#)

Completes the task by marking it as cancelled. Attempting to set this for a completed task will raise an exception.

6.14.2.2 - (void) [setError:](#) (NSError *) *error*

Completes the task by setting the error. Attempting to set this for a completed task will raise an exception.

Parameters

<i>error</i>	The error for the task.
--------------	-------------------------

6.14.2.3 - (void) setException: (NSEException *) exception

Completes the task by setting an exception. Attempting to set this for a completed task will raise an exception.

Parameters

<i>exception</i>	The exception for the task.
------------------	-----------------------------

6.14.2.4 - (void) setResult: (id) result

Completes the task by setting the result. Attempting to set this for a completed task will raise an exception.

Parameters

<i>result</i>	The result of the task.
---------------	-------------------------

6.14.2.5 + (instancetype) taskCompletionSource

Creates a new unfinished task.

6.14.2.6 - (BOOL) trySetCancelled

Sets the cancellation state of the task if it wasn't already completed.

Returns

whether the new value was set.

6.14.2.7 - (BOOL) trySetError: (NSError *) error

Sets the error of the task if it wasn't already completed.

Parameters

<i>error</i>	The error for the task.
--------------	-------------------------

Returns

whether the new value was set.

6.14.2.8 - (BOOL) trySetException: (NSEException *) exception

Sets the exception of the task if it wasn't already completed.

Parameters

<i>exception</i>	The exception for the task.
------------------	-----------------------------

Returns

whether the new value was set.

6.14.2.9 - (BOOL) trySetResult: (id) result

Sets the result of the task if it wasn't already completed.

Returns

whether the new value was set.

6.14.3 Property Documentation**6.14.3.1 - (BFTask*) task** [read], [nonatomic], [retain]

The task associated with this TaskCompletionSource.

Definition at line 30 of file BFTaskCompletionSource.h.

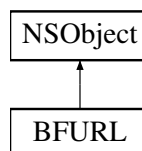
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFTaskCompletionSource.h

6.15 BFURL Class Reference

```
#import <BFURL.h>
```

Inheritance diagram for BFURL:

**Class Methods**

- (BFURL *) + [URLWithURL:](#)
- (BFURL *) + [URLWithInboundURL:sourceApplication:](#)

Properties

- NSURL * [targetURL](#)
- NSDictionary * [targetQueryParameters](#)
- NSDictionary * [appLinkData](#)
- NSDictionary * [appLinkExtras](#)
- BFAppLink * [appLinkReferer](#)
- NSURL * [inputURL](#)
- NSDictionary * [inputQueryParameters](#)

6.15.1 Detailed Description

Provides a set of utilities for working with NSURLs, such as parsing of query parameters and handling for App Link requests.

Definition at line 21 of file BFURL.h.

6.15.2 Method Documentation

6.15.2.1 `+(BFURL *) URLWithInboundURL: (NSURL *) url sourceApplication:(NSString *) sourceApplication`

Creates a link target from a raw URL received from an external application. This is typically called from the app delegate's `application:openURL:sourceApplication:annotation:` and will post the `BFApplinkNavigateInEventName` measurement event.

Parameters

<i>url</i>	The instance of <code>NSURL</code> to create <code>BFURL</code> from.
<i>source↔ Application</i>	the bundle ID of the app that is requesting your app to open the URL. The same <i>source↔ Application</i> in <code>application:openURL:sourceApplication:annotation:</code>

6.15.2.2 `+(BFURL *) URLWithURL: (NSURL *) url`

Creates a link target from a raw URL. On success, this posts the `BFApplinkParseEventName` measurement event. If you are constructing the `BFURL` within your application delegate's `application:openURL:sourceApplication↔:annotation:`, you should instead use `URLWithInboundURL:sourceApplication:` to support better `BFMeasurement↔Event` notifications

Parameters

<i>url</i>	The instance of <code>NSURL</code> to create <code>BFURL</code> from.
------------	---

6.15.3 Property Documentation

6.15.3.1 `-(NSDictionary*) appLinkData [read],[nonatomic],[strong]`

If this link target is an App Link, this is the data found in `al_applink_data`. Otherwise, it is nil.

Definition at line 55 of file BFURL.h.

6.15.3.2 `-(NSDictionary*) appLinkExtras [read],[nonatomic],[strong]`

If this link target is an App Link, this is the data found in `extras`.

Definition at line 60 of file BFURL.h.

6.15.3.3 `-(BFAppLink*) appLinkReferer [read],[nonatomic],[strong]`

The App Link indicating how to navigate back to the referer app, if any.

Definition at line 65 of file BFURL.h.

6.15.3.4 `-(NSDictionary*) inputQueryParameters [read],[nonatomic],[strong]`

The query parameters of the `inputURL`, parsed into an `NSDictionary`.

Definition at line 75 of file BFURL.h.

6.15.3.5 - (NSURL*) `inputURL` [read], [nonatomic], [strong]

The URL that was used to create this [BFURL](#).

Definition at line 70 of file BFURL.h.

6.15.3.6 - (NSDictionary*) `targetQueryParameters` [read], [nonatomic], [strong]

Gets the query parameters for the target, parsed into an NSDictionary.

Definition at line 49 of file BFURL.h.

6.15.3.7 - (NSURL*) `targetURL` [read], [nonatomic], [strong]

Gets the target URL. If the link is an App Link, this is the target of the App Link. Otherwise, it is the url that created the target.

Definition at line 44 of file BFURL.h.

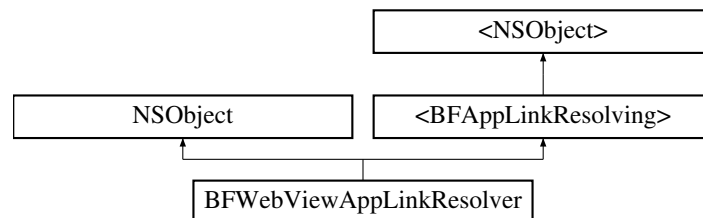
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/[BFURL.h](#)

6.16 BFWebViewAppLinkResolver Class Reference

```
#import <BFWebViewAppLinkResolver.h>
```

Inheritance diagram for BFWebViewAppLinkResolver:



Class Methods

- (instancetype) + [sharedInstance](#)

Additional Inherited Members

6.16.1 Detailed Description

A reference implementation for an App Link resolver that uses a hidden UIWebView to parse the HTML containing App Link metadata.

Definition at line 19 of file BFWebViewAppLinkResolver.h.

6.16.2 Method Documentation

6.16.2.1 + (instancetype) sharedInstance

Gets the instance of a [BFWebViewAppLinkResolver](#).

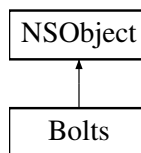
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFWebViewAppLinkResolver.h](#)

6.17 Bolts Class Reference

```
#import <Bolts.h>
```

Inheritance diagram for Bolts:



Class Methods

- (NSString *) + [version](#)

6.17.1 Detailed Description

Definition at line 29 of file Bolts.h.

6.17.2 Method Documentation

6.17.2.1 + (NSString *) version

Returns the version of the [Bolts](#) Framework as an NSString.

Returns

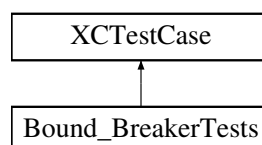
The NSString representation of the current version.

The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/Bolts.h](#)

6.18 Bound_BreakerTests Class Reference

Inheritance diagram for Bound_BreakerTests:



6.18.1 Detailed Description

Definition at line 12 of file Bound_BreakerTests.m.

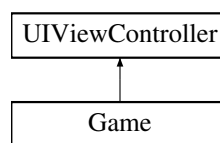
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound BreakerTests/Bound_BreakerTests.m](#)

6.19 Game Class Reference

```
#import <Game.h>
```

Inheritance diagram for Game:



Instance Methods

- (IBAction) - [StartGame](#):
Action that begins the game.
- (IBAction) - [EasyDifficulty](#):
Action that sets difficulty to easy.
- (IBAction) - [MediumDifficulty](#):
Action that sets difficulty to medium.
- (IBAction) - [HardDifficulty](#):
Action that sets difficulty to hard.
- (IBAction) - [RightPress](#):
Action that initializes ball right movement Triggered by rpressing the right movement button.
- (IBAction) - [RightLiftInside](#):
Action that deactivates ball right movement Triggered by releasing inside of the right movement button.
- (IBAction) - [RightLiftOutside](#):
Action that deactivates ball right movement Triggered by releasing outside of the right movement button.
- (IBAction) - [LeftPress](#):
Action that deactivates left ball movement Triggered by pressing the left movement button.
- (IBAction) - [LeftLiftInside](#):
Action that deactivates ball left movement Triggered by releasing inside of the left movement button.
- (IBAction) - [LeftLiftOutside](#):
Action that deactivates ball left movement Triggered by releasing outside of the left movement button.
- (void) - [ObstaclesMoving](#)
Method triggered by the ObstacleMovement timer Executes methods involved with motion.
- (void) - [ballTouchingObstacle](#)
- (void) - [ballNotTouchingObstacle](#)
- (void) - [PlaceObstacles1](#)
- (void) - [PlaceObstacles2](#)
- (void) - [Score](#)
Method to increase the score by 1, also logs the score number.
- (void) - [GameOver](#)

Protected Attributes

- IBOutlet UIButton * [TapRight](#)
- IBOutlet UIButton * [TapLeft](#)
- IBOutlet UIButton * [GameOver](#)
- IBOutlet UIButton * [EasyDifficultyButton](#)
- IBOutlet UIButton * [MediumDifficultyButton](#)
- IBOutlet UIButton * [HardDifficultyButton](#)
- IBOutlet UIImageView * [Ball](#)
- IBOutlet UIImageView * [longObstacle1](#)
- IBOutlet UIImageView * [longObstacle2](#)
- IBOutlet UIImageView * [longObstacle3](#)
- IBOutlet UIImageView * [longObstacle4](#)
- IBOutlet UILabel * [WelcomeLabel](#)
- IBOutlet UILabel * [ScoreLabel](#)
- NSTimer * [ObstacleMovement](#)

6.19.1 Detailed Description

User interface elements

Definition at line 22 of file Game.h.

6.19.2 Method Documentation

6.19.2.1 - (void) ballNotTouchingObstacle

Definition at line 178 of file Game.m.

6.19.2.2 - (void) ballTouchingObstacle

Move ball down if touching obstacles, otherwise move up towards default

Definition at line 175 of file Game.m.

6.19.2.3 - (IBAction) EasyDifficulty: (id) sender

Action that sets difficulty to easy.

Definition at line 39 of file Game.m.

6.19.2.4 - (void) GameOver

Executed once ball touches bottom of screen

Definition at line 210 of file Game.m.

6.19.2.5 - (IBAction) HardDifficulty: (id) sender

Action that sets difficulty to hard.

Definition at line 53 of file Game.m.

6.19.2.6 - (IBAction) LeftLiftInside: (id) sender

Action that deactivates ball left movement Triggered by releasing inside of the left movement button.

Definition at line 111 of file Game.m.

6.19.2.7 - (IBAction) LeftLiftOutside: (id) sender

Action that deactivates ball left movement Triggered by releasing outside of the left movement button.

Definition at line 116 of file Game.m.

6.19.2.8 - (IBAction) LeftPress: (id) sender

Action that deactivates left ball movement Triggered by pressing the left movement button.

Definition at line 106 of file Game.m.

6.19.2.9 - (IBAction) MediumDifficulty: (id) sender

Action that sets difficulty to medium.

Definition at line 46 of file Game.m.

6.19.2.10 - (void) ObstaclesMoving

Method triggered by the ObstacleMovement timer Executes methods involved with motion.

Move ball if movement buttons are pressed

Move obstacles towards bottom of screen

Place obstacles based on position of other obstacles

Highly variable depending on number of obstacles and obstacle types

Move ball towards bottom of screen if touching non-hidden obstacles, otherwise move ball up

Increase score when obstacles pass the ball's center

[Game](#) over if ball touches bottom of screen

Definition at line 123 of file Game.m.

6.19.2.11 - (void) PlaceObstacles1

Method for the random placement of obstacles

Definition at line 186 of file Game.m.

6.19.2.12 - (void) PlaceObstacles2

Definition at line 197 of file Game.m.

6.19.2.13 - (IBAction) RightLiftInside: (id) sender

Action that deactivates ball right movement Triggered by releasing inside of the right movement button.

Definition at line 95 of file Game.m.

6.19.2.14 - (IBAction) RightLiftOutside: (id) sender

Action that deactivates ball right movement Triggered by releasing outside of the right movement button.

Definition at line 100 of file Game.m.

6.19.2.15 - (IBAction) RightPress: (id) sender

Action that initializes ball right movement Triggered by rpressing the right movement button.

Definition at line 90 of file Game.m.

6.19.2.16 - (void) Score

Method to increase the score by 1, also logs the score number.

Definition at line 82 of file Game.m.

6.19.2.17 - (IBAction) StartGame: (id) sender

Action that begins the game.

Actions and methods

Definition at line 61 of file Game.m.

6.19.3 Member Data Documentation**6.19.3.1 - (IBOutlet UIImageView*) Ball [protected]**

Definition at line 31 of file Game.h.

6.19.3.2 - (IBOutlet UIButton*) EasyDifficultyButton [protected]

Definition at line 27 of file Game.h.

6.19.3.3 - (IBOutlet UIButton*) GameOver [protected]

Definition at line 26 of file Game.h.

6.19.3.4 - (IBOutlet UIButton*) HardDifficultyButton [protected]

Definition at line 29 of file Game.h.

6.19.3.5 - (IBOutlet UIImageView*) longObstacle1 [protected]

Definition at line 32 of file Game.h.

6.19.3.6 - (IBOutlet UIImageView*) longObstacle2 [protected]

Definition at line 33 of file Game.h.

6.19.3.7 - (IBOutlet UIImageView*) longObstacle3 [protected]

Definition at line 34 of file Game.h.

6.19.3.8 - (IBOutlet UIImageView*) longObstacle4 [protected]

Definition at line 35 of file Game.h.

6.19.3.9 - (IBOutlet UIButton*) MediumDifficultyButton [protected]

Definition at line 28 of file Game.h.

6.19.3.10 - (NSTimer*) ObstacleMovement [protected]

Definition at line 40 of file Game.h.

6.19.3.11 - (IBOutlet UILabel*) ScoreLabel [protected]

Definition at line 38 of file Game.h.

6.19.3.12 - (IBOutlet UIButton*) TapLeft [protected]

Definition at line 25 of file Game.h.

6.19.3.13 - (IBOutlet UIButton*) TapRight [protected]

Definition at line 24 of file Game.h.

6.19.3.14 - (IBOutlet UILabel*) WelcomeLabel [protected]

Definition at line 37 of file Game.h.

The documentation for this class was generated from the following files:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/[Game.h](#)
- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/[Game.m](#)

6.20 Game() Category Reference

Method file for the [Game](#) view. Implements game and player mechanisms.

6.20.1 Detailed Description

Method file for the [Game](#) view. Implements game and player mechanisms.

[Game.m](#)

Author

Victor Palacios

Copyright

2015 Victor Palacios

Definition at line 13 of file Game.m.

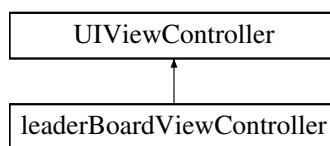
The documentation for this category was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/[Game.m](#)

6.21 leaderBoardViewController Class Reference

```
#import <leaderBoardViewController.h>
```

Inheritance diagram for leaderBoardViewController:

**Properties**

- IBOutlet UILabel * [username1](#)
- IBOutlet UILabel * [username2](#)
- IBOutlet UILabel * [username3](#)
- IBOutlet UILabel * [username4](#)
- IBOutlet UILabel * [username5](#)
- IBOutlet UILabel * [username6](#)
- IBOutlet UILabel * [username7](#)
- IBOutlet UILabel * [username8](#)
- IBOutlet UILabel * [username9](#)
- IBOutlet UILabel * [username10](#)
- IBOutlet UILabel * [score1](#)
- IBOutlet UILabel * [score2](#)
- IBOutlet UILabel * [score3](#)
- IBOutlet UILabel * [score4](#)
- IBOutlet UILabel * [score5](#)
- IBOutlet UILabel * [score6](#)
- IBOutlet UILabel * [score7](#)
- IBOutlet UILabel * [score8](#)
- IBOutlet UILabel * [score9](#)
- IBOutlet UILabel * [score10](#)

6.21.1 Detailed Description

Definition at line 11 of file leaderBoardViewController.h.

6.21.2 Property Documentation

6.21.2.1 - (IBOutlet UILabel*) [score1](#) [read], [write], [nonatomic], [weak]

Definition at line 22 of file leaderBoardViewController.h.

6.21.2.2 - (IBOutlet UILabel*) **score10** [read], [write], [nonatomic], [weak]

Definition at line 31 of file leaderBoardViewController.h.

6.21.2.3 - (IBOutlet UILabel*) **score2** [read], [write], [nonatomic], [weak]

Definition at line 23 of file leaderBoardViewController.h.

6.21.2.4 - (IBOutlet UILabel*) **score3** [read], [write], [nonatomic], [weak]

Definition at line 24 of file leaderBoardViewController.h.

6.21.2.5 - (IBOutlet UILabel*) **score4** [read], [write], [nonatomic], [weak]

Definition at line 25 of file leaderBoardViewController.h.

6.21.2.6 - (IBOutlet UILabel*) **score5** [read], [write], [nonatomic], [weak]

Definition at line 26 of file leaderBoardViewController.h.

6.21.2.7 - (IBOutlet UILabel*) **score6** [read], [write], [nonatomic], [weak]

Definition at line 27 of file leaderBoardViewController.h.

6.21.2.8 - (IBOutlet UILabel*) **score7** [read], [write], [nonatomic], [weak]

Definition at line 28 of file leaderBoardViewController.h.

6.21.2.9 - (IBOutlet UILabel*) **score8** [read], [write], [nonatomic], [weak]

Definition at line 29 of file leaderBoardViewController.h.

6.21.2.10 - (IBOutlet UILabel*) **score9** [read], [write], [nonatomic], [weak]

Definition at line 30 of file leaderBoardViewController.h.

6.21.2.11 - (IBOutlet UILabel*) **username1** [read], [write], [nonatomic], [weak]

Definition at line 12 of file leaderBoardViewController.h.

6.21.2.12 - (IBOutlet UILabel*) **username10** [read], [write], [nonatomic], [weak]

Definition at line 21 of file leaderBoardViewController.h.

6.21.2.13 - (IBOutlet UILabel*) **username2** [read], [write], [nonatomic], [weak]

Definition at line 13 of file leaderBoardViewController.h.

6.21.2.14 - (IBOutlet UILabel*) username3 [read], [write], [nonatomic], [weak]

Definition at line 14 of file leaderBoardViewController.h.

6.21.2.15 - (IBOutlet UILabel*) username4 [read], [write], [nonatomic], [weak]

Definition at line 15 of file leaderBoardViewController.h.

6.21.2.16 - (IBOutlet UILabel*) username5 [read], [write], [nonatomic], [weak]

Definition at line 16 of file leaderBoardViewController.h.

6.21.2.17 - (IBOutlet UILabel*) username6 [read], [write], [nonatomic], [weak]

Definition at line 17 of file leaderBoardViewController.h.

6.21.2.18 - (IBOutlet UILabel*) username7 [read], [write], [nonatomic], [weak]

Definition at line 18 of file leaderBoardViewController.h.

6.21.2.19 - (IBOutlet UILabel*) username8 [read], [write], [nonatomic], [weak]

Definition at line 19 of file leaderBoardViewController.h.

6.21.2.20 - (IBOutlet UILabel*) username9 [read], [write], [nonatomic], [weak]

Definition at line 20 of file leaderBoardViewController.h.

The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔
BoardViewController.h

6.22 leaderBoardViewController() Category Reference

6.22.1 Detailed Description

Definition at line 12 of file leaderBoardViewController.m.

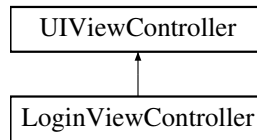
The documentation for this category was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/leader↔
BoardViewController.m

6.23 LoginViewController Class Reference

```
#import <LoginViewController.h>
```

Inheritance diagram for LoginViewController:



Instance Methods

- (IBAction) - [registerAction:](#)
- (IBAction) - [loginButton:](#)

login and log in memory

Properties

- IBOutlet UITextField * [userNameField](#)
- IBOutlet UITextField * [emailField](#)
- IBOutlet UITextField * [passwordField](#)
- IBOutlet UITextField * [reEnterPasswordField](#)
- IBOutlet UITextField * [loginUsernameField](#)
- IBOutlet UITextField * [loginPasswordField](#)

6.23.1 Detailed Description

Definition at line 12 of file `LoginViewController.h`.

6.23.2 Method Documentation

6.23.2.1 - (IBAction) `loginButton:` (id) *sender*

login and log in memory

Definition at line 93 of file `LoginViewController.m`.

6.23.2.2 - (IBAction) `registerAction:` (id) *sender*

Definition at line 46 of file `LoginViewController.m`.

6.23.3 Property Documentation

6.23.3.1 - (IBOutlet UITextField*) `emailField` [read], [write], [nonatomic], [weak]

Definition at line 14 of file `LoginViewController.h`.

6.23.3.2 - (IBOutlet UITextField*) `loginPasswordField` [read], [write], [nonatomic], [weak]

Definition at line 21 of file `LoginViewController.h`.

6.23.3.3 - (IBOutlet UITextField*) `loginUsernameField` [read], [write], [nonatomic], [weak]

Definition at line 20 of file `LoginViewController.h`.

6.23.3.4 - (IBOutlet UITextField*) passwordField [read], [write], [nonatomic], [weak]

Definition at line 15 of file LoginViewController.h.

6.23.3.5 - (IBOutlet UITextField*) reEnterPasswordField [read], [write], [nonatomic], [weak]

Definition at line 16 of file LoginViewController.h.

6.23.3.6 - (IBOutlet UITextField*) userNameField [read], [write], [nonatomic], [weak]

Definition at line 13 of file LoginViewController.h.

The documentation for this class was generated from the following files:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/LoginView↵
Controller.h
- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/LoginView↵
Controller.m

6.24 LoginViewController() Category Reference

This is the method file for [LoginViewController.m](#). implement code for logging in and creating an account.

6.24.1 Detailed Description

This is the method file for [LoginViewController.m](#). implement code for logging in and creating an account.

[LoginViewController.m](#)

Author

JScott Richards

Copyright

2015 JScott Richards

Definition at line 13 of file LoginViewController.m.

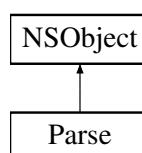
The documentation for this category was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/LoginView↵
Controller.m

6.25 Parse Class Reference

```
#import <Parse.h>
```

Inheritance diagram for Parse:



Class Methods

Connecting to Parse

- (void) + [setApplicationId:clientKey:](#)
- (NSString *) + [getApplicationId](#)
- (NSString *) + [getClientKey](#)

Enabling Local Datastore

- (void) + [enableLocalDatastore](#)
- (BOOL) + [isLocalDatastoreEnabled](#)

Logging

- (void) + [setLogLevel:](#)
- (PFLogLevel) + [logLevel](#)

6.25.1 Detailed Description

The [Parse](#) class contains static functions that handle global configuration for the [Parse](#) framework.

Definition at line 62 of file [Parse.h](#).

6.25.2 Method Documentation

6.25.2.1 + (void) enableLocalDatastore

Enable pinning in your application. This must be called before your application can use pinning. The recommended way is to call this method before [setApplicationId:clientKey:](#).

6.25.2.2 + (NSString *) getApplicationId

The current application id that was used to configure [Parse](#) framework.

6.25.2.3 + (NSString *) getClientKey

The current client key that was used to configure [Parse](#) framework.

6.25.2.4 + (BOOL) isLocalDatastoreEnabled

Flag that indicates whether Local Datastore is enabled.

Returns

YES if Local Datastore is enabled, otherwise NO.

6.25.2.5 + (PFLogLevel) logLevel

Log level that will be displayed.

By default:

- If running inside an app that was downloaded from iOS App Store - it is set to `<PFLogLevelNone>`
- All other cases - it is set to `<PFLogLevelWarning>`

Returns

A <PFLogLevel> value.

See also

PFLogLevel

6.25.2.6 + (void) setApplicationId: (NSString *) *applicationId* clientKey:(NSString *) *clientKey*

Sets the applicationId and clientKey of your application.

Parameters

<i>applicationId</i>	The application id of your Parse application.
<i>clientKey</i>	The client key of your Parse application.

6.25.2.7 + (void) setLogLevel: (PFLogLevel) *logLevel*

Sets the level of logging to display.

By default:

- If running inside an app that was downloaded from iOS App Store - it is set to <PFLogLevelNone>
- All other cases - it is set to <PFLogLevelWarning>

Parameters

<i>logLevel</i>	Log level to set.
-----------------	-------------------

See also

PFLogLevel

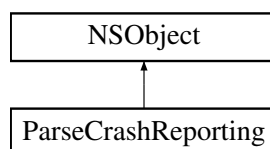
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[Parse.h](#)↩

6.26 ParseCrashReporting Class Reference

```
#import <ParseCrashReporting.h>
```

Inheritance diagram for ParseCrashReporting:

**Class Methods****Crash Reporting**

- (void) + [enable](#)
- (BOOL) + [isCrashReportingEnabled](#)

6.26.1 Detailed Description

The `ParseCrashReporting` class is responsible for enabling crash reporting in your application.

Definition at line 14 of file `ParseCrashReporting.h`.

6.26.2 Method Documentation

6.26.2.1 + (void) enable

Enables crash reporting for your app.

Warning

This must be called before you set Application ID and Client Key on `Parse`.

6.26.2.2 + (BOOL) isCrashReportingEnabled

Indicates whether crash reporting is currently enabled.

Returns

YES if crash reporting is enabled, NO - otherwise.

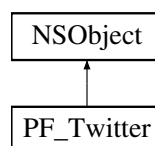
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseCrashReporting.h`

6.27 PF_Twitter Class Reference

```
#import <PF_Twitter.h>
```

Inheritance diagram for `PF_Twitter`:



Instance Methods

- (BFTask *) - `authorizeInBackground`
- (void) - `authorizeWithSuccess:failure:cancel:`
- (void) - `signRequest:`

Properties

- NSString * `consumerKey`
- NSString * `consumerSecret`
- NSString * `authToken`
- NSString * `authTokenSecret`
- NSString * `userId`
- NSString * `screenName`

6.27.1 Detailed Description

The `PF_Twitter` class is a simple interface for interacting with the Twitter REST API, automating sign-in and OAuth signing of requests against the API.

Definition at line 19 of file `PF_Twitter.h`.

6.27.2 Method Documentation

6.27.2.1 `-(BFTask *) authorizeInBackground`

Displays an auth dialog and populates the `authToken`, `authTokenSecret`, `userId`, and `screenName` properties if the Twitter user grants permission to the application.

Returns

The task, that encapsulates the work being done.

6.27.2.2 `-(void) authorizeWithSuccess: (PF_NULLABLE void (^)(void)) success failure:(PF_NULLABLE void (^)(NSError *PF_NULLABLE_S error)) failure cancel:(PF_NULLABLE void (^)(void)) cancel`

Displays an auth dialog and populates the `authToken`, `authTokenSecret`, `userId`, and `screenName` properties if the Twitter user grants permission to the application.

Parameters

<i>success</i>	Invoked upon successful authorization.
<i>failure</i>	Invoked upon an error occurring in the authorization process.
<i>cancel</i>	Invoked when the user cancels authorization.

6.27.2.3 `-(void) signRequest: (PF_NULLABLE NSMutableURLRequest *) request`

Adds a 3-legged OAuth signature to an `NSMutableURLRequest` based upon the properties set for the Twitter object.

Use this function to sign requests being made to the Twitter API.

Parameters

<i>request</i>	Request to sign.
----------------	------------------

6.27.3 Property Documentation

6.27.3.1 `-(NSString*) authToken` `[read], [write], [nonatomic], [copy]`

Auth token for the current user.

Definition at line 34 of file `PF_Twitter.h`.

6.27.3.2 `-(NSString*) authTokenSecret` `[read], [write], [nonatomic], [copy]`

Auth token secret for the current user.

Definition at line 39 of file `PF_Twitter.h`.

6.27.3.3 - (NSString*) **consumerKey** [read], [write], [nonatomic], [copy]

Consumer key of the application that is used to authorize with Twitter.

Definition at line 24 of file PF_Twitter.h.

6.27.3.4 - (NSString*) **consumerSecret** [read], [write], [nonatomic], [copy]

Consumer secret of the application that is used to authorize with Twitter.

Definition at line 29 of file PF_Twitter.h.

6.27.3.5 - (NSString*) **screenName** [read], [write], [nonatomic], [copy]

Twitter screen name of the currently signed in user.

Definition at line 49 of file PF_Twitter.h.

6.27.3.6 - (NSString*) **userId** [read], [write], [nonatomic], [copy]

Twitter user id of the currently signed in user.

Definition at line 44 of file PF_Twitter.h.

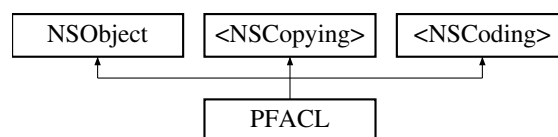
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PF_Twitter.h

6.28 PFACL Class Reference

```
#import <PFACL.h>
```

Inheritance diagram for PFACL:



Instance Methods

Controlling Public Access

- (void) - [setPublicReadAccess:](#)
- (BOOL) - [getPublicReadAccess](#)
- (void) - [setPublicWriteAccess:](#)
- (BOOL) - [getPublicWriteAccess](#)

Controlling Access Per-User

- (void) - [setReadAccess:forUserId:](#)
- (BOOL) - [getReadAccessForUserId:](#)
- (void) - [setWriteAccess:forUserId:](#)
- (BOOL) - [getWriteAccessForUserId:](#)

- (void) - [setReadAccess:forUser:](#)
- (BOOL) - [getReadAccessForUser:](#)
- (void) - [setWriteAccess:forUser:](#)
- (BOOL) - [getWriteAccessForUser:](#)

Controlling Access Per-Role

- (BOOL) - [getReadAccessForRoleWithName:](#)
- (void) - [setReadAccess:forRoleWithName:](#)
- (BOOL) - [getWriteAccessForRoleWithName:](#)
- (void) - [setWriteAccess:forRoleWithName:](#)
- (BOOL) - [getReadAccessForRole:](#)
- (void) - [setReadAccess:forRole:](#)
- (BOOL) - [getWriteAccessForRole:](#)
- (void) - [setWriteAccess:forRole:](#)

Class Methods

Creating an ACL

- (PFACL *) + [ACL](#)
- (PFACL *) + [ACLWithUser:](#)

Setting Access Defaults

- (void) + [setDefaultACL:withAccessForCurrentUser:](#)

6.28.1 Detailed Description

The [PFACL](#) class is used to control which users can access or modify a particular object. Each <PFObj> can have its own [PFACL](#). You can grant read and write permissions separately to specific users, to groups of users that belong to roles, or you can grant permissions to "the public" so that, for example, any user could read a particular object but only a particular set of users could write to that object.

Definition at line 26 of file PFACL.h.

6.28.2 Method Documentation

6.28.2.1 + (PFACL *) ACL

Creates an ACL with no permissions granted.

Returns

Returns a new [PFACL](#).

6.28.2.2 + (PFACL *) ACLWithUser: (PFUser *) user

Creates an ACL where only the provided user has access.

Parameters

<i>user</i>	The user to assign access.
-------------	----------------------------

6.28.2.3 - (BOOL) getPublicReadAccess

Gets whether the public is allowed to read this object.

Returns

YES if the public read access is enabled, otherwise NO.

6.28.2.4 - (BOOL) getPublicWriteAccess

Gets whether the public is allowed to write this object.

Returns

YES if the public write access is enabled, otherwise NO.

6.28.2.5 - (BOOL) getReadAccessForRole: (PFRole *) role

Get whether users belonging to the given role are allowed to read this object. Even if this returns NO, the role may still be able to read it if a parent role has read access.

The role must already be saved on the server and it's data must have been fetched in order to use this method.

Parameters

<i>role</i>	The name of the role.
-------------	-----------------------

Returns

YES if the role has read access, otherwise NO.

6.28.2.6 - (BOOL) getReadAccessForRoleWithName: (NSString *) name

Get whether users belonging to the role with the given name are allowed to read this object. Even if this returns NO, the role may still be able to read it if a parent role has read access.

Parameters

<i>name</i>	The name of the role.
-------------	-----------------------

Returns

YES if the role has read access, otherwise NO.

6.28.2.7 - (BOOL) getReadAccessForUser: (PFUser *) user

Gets whether the given user is *explicitly* allowed to read this object. Even if this returns NO, the user may still be able to access it if <getPublicReadAccess> returns YES or if the user belongs to a role that has access.

Parameters

<i>user</i>	The user for which to retrieve access.
-------------	--

Returns

YES if the user has *explicit* read access, otherwise NO.

6.28.2.8 - (BOOL) getReadAccessForUserId: (NSString *) userId

Gets whether the given user id is *explicitly* allowed to read this object. Even if this returns NO, the user may still be able to access it if <getPublicReadAccess> returns YES or if the user belongs to a role that has access.

Parameters

<i>userId</i>	The <[PFObjecT objectId]> of the user for which to retrieve access.
---------------	---

Returns

YES if the user with this objectId has *explicit* read access, otherwise NO.

6.28.2.9 - (BOOL) getWriteAccessForRole: (PFRole *) role

Get whether users belonging to the given role are allowed to write this object. Even if this returns NO, the role may still be able to write it if a parent role has write access.

The role must already be saved on the server and its data must have been fetched in order to use this method.

Parameters

<i>role</i>	The name of the role.
-------------	-----------------------

Returns

YES if the role has write access, otherwise NO.

6.28.2.10 - (BOOL) getWriteAccessForRoleWithName: (NSString *) name

Get whether users belonging to the role with the given name are allowed to write this object. Even if this returns NO, the role may still be able to write it if a parent role has write access.

Parameters

<i>name</i>	The name of the role.
-------------	-----------------------

Returns

YES if the role has read access, otherwise NO.

6.28.2.11 - (BOOL) getWriteAccessForUser: (PFUser *) user

Gets whether the given user is *explicitly* allowed to write this object. Even if this returns NO, the user may still be able to write it if <getPublicWriteAccess> returns YES or if the user belongs to a role that has access.

Parameters

<i>user</i>	The user for which to retrieve access.
-------------	--

Returns

YES if the user has *explicit* write access, otherwise NO.

6.28.2.12 - (BOOL) getWriteAccessForUserId: (NSString *) *userId*

Gets whether the given user id is *explicitly* allowed to write this object. Even if this returns NO, the user may still be able to write it if <getPublicWriteAccess> returns YES or if the user belongs to a role that has access.

Parameters

<i>userId</i>	The <[PFObject objectId]> of the user for which to retrieve access.
---------------	---

Returns

YES if the user with this *objectId* has *explicit* write access, otherwise NO.

6.28.2.13 + (void) setDefaultACL: (PF_NULLABLE PFAcl *) *acl* withAccessForCurrentUser:(BOOL) *currentUserAccess*

Sets a default ACL that will be applied to all instances of <PFObject> when they are created.

Parameters

<i>acl</i>	The ACL to use as a template for all instance of <PFObject> created after this method has been called. This value will be copied and used as a template for the creation of new ACLs, so changes to the instance after this method has been called will not be reflected in new instance of <PFObject>.
<i>currentUserAccess</i>	<p>- If YES, the PFAcl that is applied to newly-created instance of <PFObject> will provide read and write access to the <[PFUser currentUser]> at the time of creation.</p> <ul style="list-style-type: none"> • If NO, the provided <i>acl</i> will be used without modification. • If <i>acl</i> is nil, this value is ignored.

6.28.2.14 - (void) setPublicReadAccess: (BOOL) *allowed*

Set whether the public is allowed to read this object.

Parameters

<i>allowed</i>	Whether the public can read this object.
----------------	--

6.28.2.15 - (void) setPublicWriteAccess: (BOOL) *allowed*

Set whether the public is allowed to write this object.

Parameters

<i>allowed</i>	Whether the public can write this object.
----------------	---

6.28.2.16 - (void) setReadAccess: (BOOL) *allowed* forRole:(PFRole *) *role*

Set whether users belonging to the given role are allowed to read this object.

The role must already be saved on the server and it's data must have been fetched in order to use this method.

Parameters

<i>allowed</i>	Whether the given role can read this object.
<i>role</i>	The role to assign access.

6.28.2.17 - (void) setReadAccess: (BOOL) *allowed* forRoleWithName:(NSString *) *name*

Set whether users belonging to the role with the given name are allowed to read this object.

Parameters

<i>allowed</i>	Whether the given role can read this object.
<i>name</i>	The name of the role.

6.28.2.18 - (void) setReadAccess: (BOOL) *allowed* forUser:(PFUser *) *user*

Set whether the given user is allowed to read this object.

Parameters

<i>allowed</i>	Whether the given user can read this object.
<i>user</i>	The user to assign access.

6.28.2.19 - (void) setReadAccess: (BOOL) *allowed* forUserId:(NSString *) *userId*

Set whether the given user id is allowed to read this object.

Parameters

<i>allowed</i>	Whether the given user can write this object.
<i>userId</i>	The <[PFObject objectId]> of the user to assign access.

6.28.2.20 - (void) setWriteAccess: (BOOL) *allowed* forRole:(PFRole *) *role*

Set whether users belonging to the given role are allowed to write this object.

The role must already be saved on the server and it's data must have been fetched in order to use this method.

Parameters

<i>allowed</i>	Whether the given role can write this object.
<i>role</i>	The role to assign access.

6.28.2.21 - (void) setWriteAccess: (BOOL) *allowed* forRoleWithName:(NSString *) *name*

Set whether users belonging to the role with the given name are allowed to write this object.

Parameters

<i>allowed</i>	Whether the given role can write this object.
<i>name</i>	The name of the role.

6.28.2.22 - (void) setWriteAccess: (BOOL) *allowed* forUser:(PFUser *) *user*

Set whether the given user is allowed to write this object.

Parameters

<i>allowed</i>	Whether the given user can write this object.
<i>user</i>	The user to assign access.

6.28.2.23 - (void) setWriteAccess: (BOOL) *allowed* forUserId:(NSString *) *userId*

Set whether the given user id is allowed to write this object.

Parameters

<i>allowed</i>	Whether the given user can read this object.
<i>userId</i>	The <code>objectId</code> of the user to assign access.

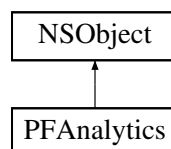
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFAalytics.h](#)

6.29 PFAalytics Class Reference

```
#import <PFAalytics.h>
```

Inheritance diagram for PFAalytics:



Class Methods

App-Open / Push Analytics

- (BFTask *) + [trackAppOpenedWithLaunchOptions:](#)
- (void) + [trackAppOpenedWithLaunchOptionsInBackground:block:](#)
- (BFTask *) + [trackAppOpenedWithRemoteNotificationPayload:](#)
- (void) + [trackAppOpenedWithRemoteNotificationPayloadInBackground:block:](#)

Custom Analytics

- (BFTask *) + [trackEvent:](#)
- (void) + [trackEventInBackground:block:](#)
- (BFTask *) + [trackEvent:dimensions:](#)
- (void) + [trackEventInBackground:dimensions:block:](#)

6.29.1 Detailed Description

`PFAnalytics` provides an interface to `Parse`'s logging and analytics backend.

Methods will return immediately and cache the request (+ timestamp) to be handled "eventually." That is, the request will be sent immediately if possible or the next time a network connection is available.

Definition at line 26 of file `PFAnalytics.h`.

6.29.2 Method Documentation

6.29.2.1 `+(BFTask *) trackAppOpenedWithLaunchOptions: (PF_NULLABLE NSDictionary *) launchOptions`

Tracks this application being launched. If this happened as the result of the user opening a push notification, this method sends along information to correlate this open with that push.

Pass in `nil` to track a standard "application opened" event.

Parameters

<i>launchOptions</i>	The <code>NSDictionary</code> indicating the reason the application was launched, if any. This value can be found as a parameter to various <code>UIApplicationDelegate</code> methods, and can be empty or <code>nil</code> .
----------------------	--

Returns

Returns the task encapsulating the work being done.

6.29.2.2 `+(void) trackAppOpenedWithLaunchOptionsInBackground: (PF_NULLABLE NSDictionary *) launchOptions block:(PF_NULLABLE PFBooleanResultBlock) block`

Tracks this application being launched. If this happened as the result of the user opening a push notification, this method sends along information to correlate this open with that push.

Pass in `nil` to track a standard "application opened" event.

Parameters

<i>launchOptions</i>	The dictionary indicating the reason the application was launched, if any. This value can be found as a parameter to various <code>UIApplicationDelegate</code> methods, and can be empty or <code>nil</code> .
<i>block</i>	The block to execute on server response. It should have the following argument signature: <code>^(BOOL succeeded, NSError *error)</code>

6.29.2.3 `+(BFTask *) trackAppOpenedWithRemoteNotificationPayload: (PF_NULLABLE NSDictionary *) userInfo`

Tracks this application being launched. If this happened as the result of the user opening a push notification, this method sends along information to correlate this open with that push.

Parameters

<i>userInfo</i>	The Remote Notification payload, if any. This value can be found either under <code>UIApplicationLaunchOptionsRemoteNotificationKey</code> on <code>launchOptions</code> , or as a parameter to <code>application:didReceiveRemoteNotification:.</code> This can be empty or <code>nil</code> .
-----------------	---

Returns

Returns the task encapsulating the work being done.

6.29.2.4 + (void) trackAppOpenedWithRemoteNotificationPayloadInBackground: (PF_NULLABLE NSDictionary *) *userInfo* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Tracks this application being launched. If this happened as the result of the user opening a push notification, this method sends along information to correlate this open with that push.

Parameters

<i>userInfo</i>	The Remote Notification payload, if any. This value can be found either under <code>UIApplicationLaunchOptionsRemoteNotificationKey</code> on <code>launchOptions</code> , or as a parameter to <code>application:didReceiveRemoteNotification:</code> . This can be empty or <code>nil</code> .
<i>block</i>	The block to execute on server response. It should have the following argument signature: <code>^(BOOL succeeded, NSError *error)</code>

6.29.2.5 + (BFTask *) trackEvent: (NSString *) *name*

Tracks the occurrence of a custom event.

[Parse](#) will store a data point at the time of invocation with the given event name.

Parameters

<i>name</i>	The name of the custom event to report to Parse as having happened.
-------------	---

Returns

Returns the task encapsulating the work being done.

6.29.2.6 + (BFTask *) trackEvent: (NSString *) *name* dimensions:(PF_NULLABLE NSDictionary *) *dimensions*

Tracks the occurrence of a custom event with additional dimensions. [Parse](#) will store a data point at the time of invocation with the given event name.

Dimensions will allow segmentation of the occurrences of this custom event. Keys and values should be `NSString`s, and will throw otherwise.

To track a user signup along with additional metadata, consider the following:

```
NSDictionary *dimensions = @"gender": @"m", @"source": @"web", @"dayType": @"weekend" }; [PFAalytics trackEvent:@"signup" dimensions:dimensions];
```

Warning

There is a default limit of 8 dimensions per event tracked.

Parameters

<i>name</i>	The name of the custom event to report to Parse as having happened.
<i>dimensions</i>	The <code>NSDictionary</code> of information by which to segment this event.

Returns

Returns the task encapsulating the work being done.

6.29.2.7 + (void) trackEventInBackground: (NSString *) *name* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Tracks the occurrence of a custom event. [Parse](#) will store a data point at the time of invocation with the given event name. The event will be sent at some unspecified time in the future, even if [Parse](#) is currently inaccessible.

Parameters

<i>name</i>	The name of the custom event to report to Parse as having happened.
<i>block</i>	The block to execute on server response. It should have the following argument signature: <code>^(BOOL succeeded, NSError *error)</code>

6.29.2.8 + (void) trackEventInBackground: (NSString *) *name* dimensions:(PF_NULLABLE NSDictionary *) *dimensions* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Tracks the occurrence of a custom event with additional dimensions. [Parse](#) will store a data point at the time of invocation with the given event name. The event will be sent at some unspecified time in the future, even if [Parse](#) is currently inaccessible.

will allow segmentation of the occurrences of this custom event. Keys and values should be NSStrings, and will throw otherwise.

To track a user signup along with additional metadata, consider the following: NSDictionary *dimensions = @"gender": @"m", @"source": @"web", @"dayType": @"weekend" }; [PFAalytics trackEvent:@"signup" dimensions:dimensions];

There is a default limit of 8 dimensions per event tracked.

Parameters

<i>name</i>	The name of the custom event to report to Parse as having happened.
<i>dimensions</i>	The NSDictionary of information by which to segment this event.
<i>block</i>	The block to execute on server response. It should have the following argument signature: <code>^(BOOL succeeded, NSError *error)</code>

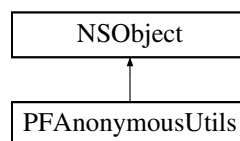
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFAalytics.h](#)

6.30 PFAnonymousUtils Class Reference

```
#import <PFAnonymousUtils.h>
```

Inheritance diagram for PFAnonymousUtils:



Class Methods

Creating an Anonymous User

- (BFTask *) + [logInBackground](#)
- (void) + [logInWithBlock:](#)
- (void) + [logInWithTarget:selector:](#)

Determining Whether a User is Anonymous

- (BOOL) + [isLinkedWithUser:](#)

6.30.1 Detailed Description

Provides utility functions for working with Anonymously logged-in users. Anonymous users have some unique characteristics:

- Anonymous users don't need a user name or password.
- Once logged out, an anonymous user cannot be recovered.
- When the current user is anonymous, the following methods can be used to switch to a different user or convert the anonymous user into a regular one:
- `signUp` converts an anonymous user to a standard user with the given username and password. Data associated with the anonymous user is retained.
- `login` switches users without converting the anonymous user. Data associated with the anonymous user will be lost.
- Service login (e.g. Facebook, Twitter) will attempt to convert the anonymous user into a standard user by linking it to the service. If a user already exists that is linked to the service, it will instead switch to the existing user.
- Service linking (e.g. Facebook, Twitter) will convert the anonymous user into a standard user by linking it to the service.

Definition at line 37 of file `PFAnonymousUtils.h`.

6.30.2 Method Documentation

6.30.2.1 + (BOOL) isLinkedWithUser: (PF_NULLABLE PFUser *) user

Whether the <PFUser> object is logged in anonymously.

Parameters

<i>user</i>	<PFUser> object to check for anonymity. The user must be logged in on this device.
-------------	--

Returns

YES if the user is anonymous. NO if the user is not the current user or is not anonymous.

6.30.2.2 + (BFTask *) loginInBackground

Creates an anonymous user asynchronously and sets as a result to [BFTask](#).

Returns

The task, that encapsulates the work being done.

6.30.2.3 + (void) loginWithBlock: (PF_NULLABLE PFUserResultBlock) block

Creates an anonymous user.

Parameters

<i>block</i>	The block to execute when anonymous user creation is complete. It should have the following argument signature: \wedge (PFUser *user, NSError *error).
--------------	---

6.30.2.4 + (void) loginWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

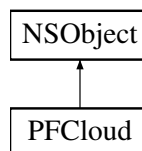
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFAuthenticationUtils.h](#)

6.31 PFCloud Class Reference

```
#import <PFCloud.h>
```

Inheritance diagram for PFCloud:



Class Methods

- (PF_NULLABLE_S id) + callFunction:withParameters:
- (PF_NULLABLE_S id) + callFunction:withParameters:error:
- (BFTask *) + callFunctionInBackground:withParameters:
- (void) + callFunctionInBackground:withParameters:block:
- (void) + callFunctionInBackground:withParameters:target:selector:

6.31.1 Detailed Description

The [PFCloud](#) class provides methods for interacting with [Parse](#) Cloud Functions.

Definition at line 22 of file [PFCloud.h](#).

6.31.2 Method Documentation

6.31.2.1 + (PF_NULLABLE_S id) callFunction: (NSString *) *function* withParameters:(PF_NULLABLE NSDictionary *) *parameters*

Calls the given cloud function *synchronously* with the parameters provided.

Parameters

<i>function</i>	The function name to call.
<i>parameters</i>	The parameters to send to the function.

Returns

The response from the cloud function.

6.31.2.2 + (PF_NULLABLE_S id) callFunction: (NSString *) *function* withParameters:(PF_NULLABLE NSDictionary *)
parameters error:(NSError **) *error*

Calls the given cloud function *synchronously* with the parameters provided and sets the error if there is one.

Parameters

<i>function</i>	The function name to call.
<i>parameters</i>	The parameters to send to the function.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

The response from the cloud function. This result could be a `NSDictionary`, an `NSArray`, `NSNumber` or `NSString`.

6.31.2.3 + (BFTask *) callFunctionInBackground: (NSString *) *function* withParameters:(PF_NULLABLE NSDictionary *) *parameters*

Calls the given cloud function *asynchronously* with the parameters provided.

Parameters

<i>function</i>	The function name to call.
<i>parameters</i>	The parameters to send to the function.

Returns

The task, that encapsulates the work being done.

6.31.2.4 + (void) callFunctionInBackground: (NSString *) *function* withParameters:(PF_NULLABLE NSDictionary *) *parameters* block:(PF_NULLABLE PFIdResultBlock) *block*

Calls the given cloud function *asynchronously* with the parameters provided and executes the given block when it is done.

Parameters

<i>function</i>	The function name to call.
<i>parameters</i>	The parameters to send to the function.
<i>block</i>	The block to execute when the function call finished. It should have the following argument signature: <code>^(id result, NSError *error).</code>

6.31.2.5 + (void) callFunctionInBackground: (NSString *) *function* withParameters:(PF_NULLABLE NSDictionary *) *parameters* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

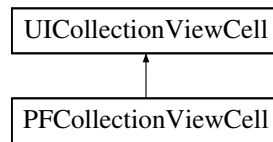
The documentation for this class was generated from the following file:

- /Users/jscootrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFCLOUD.h](#)

6.32 PFCollectionTableViewCell Class Reference

```
#import <PFCollectionTableViewCell.h>
```

Inheritance diagram for `PFCollectionTableViewCell`:



Instance Methods

- (void) - [updateFromObject:](#)

Properties

- UILabel * [textLabel](#)
- [PFImageView](#) * [imageView](#)

6.32.1 Detailed Description

The [PFCollectionViewCell](#) class represents a collection view cell which can download and display remote images stored on [Parse](#) as well as has a default simple text label.

Definition at line 35 of file [PFCollectionViewCell.h](#).

6.32.2 Method Documentation

6.32.2.1 - (void) [updateFromObject:](#) ([PFUI_NULLABLE PFObject *](#)) *object*

This method should update all the relevant information inside a subclass of [PFCollectionViewCell](#).

This method is automatically called by `<PFQueryCollectionViewController>` whenever the cell should display new information. By default this method does nothing.

Parameters

<i>object</i>	An instance of PFObject to update from.
---------------	---

6.32.3 Property Documentation

6.32.3.1 - ([PFImageView*](#)) [imageView](#) [[read](#)], [[nonatomic](#)], [[strong](#)]

The lazy-loaded [imageView](#) of the collection view cell.

See also

[PFImageView](#)

Definition at line 47 of file [PFCollectionViewCell.h](#).

6.32.3.2 - ([UILabel*](#)) [textLabel](#) [[read](#)], [[nonatomic](#)], [[strong](#)]

A simple lazy-loaded label for the collection view cell.

Definition at line 40 of file [PFCollectionViewCell.h](#).

The documentation for this class was generated from the following file:

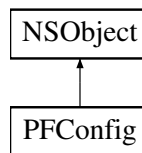
- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/BoundHeaders/PFCollectionViewCell.h](#)

[Breaker/ParseUI.framework/↔](#)

6.33 PFConfig Class Reference

```
#import <PFConfig.h>
```

Inheritance diagram for PFConfig:



Instance Methods

Parameters

- (PF_NULLABLE_S id) - [objectForKey:](#)
- (PF_NULLABLE_S id) - [objectForKeyedSubscript:](#)

Class Methods

Current Config

- (PFConfig *) + [currentConfig](#)

Retrieving Config

- (PF_NULLABLE PFConfig *) + [getConfig](#)
- (PF_NULLABLE PFConfig *) + [getConfig:](#)
- (BFTask *) + [getConfigInBackground](#)
- (void) + [getConfigInBackgroundWithBlock:](#)

6.33.1 Detailed Description

[PFConfig](#) is a representation of the remote configuration object. It enables you to add things like feature gating, a/b testing or simple "Message of the day".

Definition at line 26 of file [PFConfig.h](#).

6.33.2 Method Documentation

6.33.2.1 + (PFConfig *) [currentConfig](#)

Returns the most recently fetched config.

If there was no config fetched - this method will return an empty instance of [PFConfig](#).

Returns

Current, last fetched instance of [PFConfig](#).

6.33.2.2 + (PF_NULLABLE PFConfig *) getConfig

Gets the [PFConfig](#) object *synchronously* from the server.

Returns

Instance of [PFConfig](#) if the operation succeeded, otherwise `nil`.

6.33.2.3 + (PF_NULLABLE PFConfig *) getConfig: (NSError **) error

Gets the [PFConfig](#) object *synchronously* from the server and sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Instance of [PFConfig](#) if the operation succeeded, otherwise `nil`.

6.33.2.4 + (BFTask *) getConfigInBackground

Gets the [PFConfig](#) *asynchronously* and sets it as a result of a task.

Returns

The task, that encapsulates the work being done.

6.33.2.5 + (void) getConfigInBackgroundWithBlock: (PF_NULLABLE PFConfigResultBlock) block

Gets the [PFConfig](#) *asynchronously* and executes the given callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(PFConfig *config, NSError *error)</code> .
--------------	---

6.33.2.6 - (PF_NULLABLE_S id) objectForKey: (NSString *) key

Returns the object associated with a given key.

Parameters

<i>key</i>	The key for which to return the corresponding configuration value.
------------	--

Returns

The value associated with `key`, or `nil` if there is no such value.

6.33.2.7 - (PF_NULLABLE_S id) objectForKeyedSubscript: (NSString *) keyedSubscript

Returns the object associated with a given key.

This method enables usage of literal syntax on [PFConfig](#). E.g. `NSString *value = config[@"key"];`

See also

- [objectForKey:](#)

Parameters

<i>keyedSubscript</i>	The keyed subscript for which to return the corresponding configuration value.
-----------------------	--

Returns

The value associated with `key`, or `nil` if there is no such value.

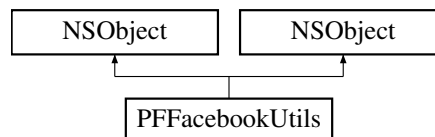
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/P↵FConfig.h](#)

6.34 PFFacebookUtils Class Reference

```
#import <PFFacebookUtils.h>
```

Inheritance diagram for PFFacebookUtils:



Class Methods

Interacting With Facebook

- (PF_NULLABLE FBSession *) + [session](#)
- (void) + [initWithApplicationId:](#)
- (void) + [initWithApplicationId:urlSchemeSuffix:](#)
- (void) + [initializeFacebook](#)
- (void) + [initializeFacebookWithURLShemeSuffix:](#)
- (BOOL) + [isLinkedWithUser:](#)
- (void) + [initializeFacebookWithApplicationLaunchOptions:](#)
- (FBSDKLoginManager *) + [facebookLoginManager](#)

Customizing Login Behavior

- (void) + [setFacebookLoginBehavior:](#)
- (FBSessionLoginBehavior) + [facebookLoginBehavior](#)

Logging In & Creating Facebook-Linked Users

- (BFTask *) + [loginWithPermissionsInBackground:](#)
- (void) + [loginWithPermissions:block:](#)
- (void) + [loginWithPermissions:target:selector:](#)
- (BFTask *) + [loginWithFacebookIdInBackground:accessToken:expirationDate:](#)
- (void) + [loginWithFacebookId:accessToken:expirationDate:block:](#)
- (void) + [loginWithFacebookId:accessToken:expirationDate:target:selector:](#)

Linking Users with Facebook

- (void) + [linkUser:permissions:](#)
- (BFTask *) + [linkUserInBackground:permissions:](#)
- (void) + [linkUser:permissions:block:](#)

- (void) + [linkUser:permissions:target:selector:](#)
- (BFTask *) + [linkUserInBackground:facebookId:accessToken:expirationDate:](#)
- (void) + [linkUser:facebookId:accessToken:expirationDate:block:](#)
- (void) + [linkUser:facebookId:accessToken:expirationDate:target:selector:](#)

Unlinking Users from Facebook

- (BOOL) + [unlinkUser:](#)
- (BOOL) + [unlinkUser:error:](#)
- (BFTask *) + [unlinkUserInBackground:](#)
- (void) + [unlinkUserInBackground:block:](#)
- (void) + [unlinkUserInBackground:target:selector:](#)

Obtaining New Permissions

- (BFTask *) + [reauthorizeUserInBackground:withPublishPermissions:audience:](#)
- (void) + [reauthorizeUser:withPublishPermissions:audience:block:](#)
- (void) + [reauthorizeUser:withPublishPermissions:audience:target:selector:](#)

Delegating URL Actions

- (BOOL) + [handleOpenURL:](#)

Logging In

- (BFTask *) + [loginInBackgroundWithReadPermissions:](#)
- (void) + [loginInBackgroundWithReadPermissions:block:](#)
- (BFTask *) + [loginInBackgroundWithPublishPermissions:](#)
- (void) + [loginInBackgroundWithPublishPermissions:block:](#)
- (BFTask *) + [loginInBackgroundWithAccessToken:](#)
- (void) + [loginInBackgroundWithAccessToken:block:](#)

Linking Users

- (BFTask *) + [linkUserInBackground:withReadPermissions:](#)
- (void) + [linkUserInBackground:withReadPermissions:block:](#)
- (BFTask *) + [linkUserInBackground:withPublishPermissions:](#)
- (void) + [linkUserInBackground:withPublishPermissions:block:](#)
- (BFTask *) + [linkUserInBackground:withAccessToken:](#)
- (void) + [linkUserInBackground:withAccessToken:block:](#)

Unlinking Users

- (BFTask *) + [unlinkUserInBackground:](#)
- (void) + [unlinkUserInBackground:block:](#)

Getting Linked State

- (BOOL) + [isLinkedWithUser:](#)

6.34.1 Detailed Description

The `PFFacebookUtils` class provides utility functions for working with Facebook in a [Parse](#) application.

This class is currently for iOS only.

The `PFFacebookUtils` class provides utility functions for using Facebook authentication with `<PFUser>`s.

Warning

This class supports official Facebook iOS SDK v4.0+ and is available only on iOS.

Definition at line 24 of file `PFFacebookUtils.h`.

6.34.2 Method Documentation

6.34.2.1 + (FBSessionLoginBehavior) facebookLoginBehavior

Login behavior controls whether to allow, force or prohibit Facebook Login or Inline Facebook Login.

Default is to allow Facebook Login, with fallback to Inline Facebook Login.

Returns

The login behavior currently set.

6.34.2.2 + (FBSDKLoginManager *) facebookLoginManager

`FBSDKLoginManager` provides methods for configuring login behavior, default audience and managing Facebook Access Token.

Returns

An instance of `FBSDKLoginManager` that is used by `PFFacebookUtils`.

6.34.2.3 + (BOOL) handleOpenURL: ("Use instead.") *PARSE_DEPRECATED*[FBAppCall handleOpenURL:sourceApplication:withSession:]

Handles URLs being opened by your `AppDelegate`. Invoke and return this from `application:handleOpenURL:` or `application:openURL:sourceApplication:annotation:` in your `AppDelegate`.

Parameters

<i>url</i>	URL being opened by your application.
------------	---------------------------------------

Returns

YES if Facebook will handle this URL, otherwise NO.

Deprecated Please use '[FBAppCall handleOpenURL:url sourceApplication:sourceApplication withSession:[`PFFacebookUtils` session]]'; instead.

6.34.2.4 + (void) initializeFacebook

Initializes the Facebook singleton.

You must provide your Facebook application ID as the value for FacebookAppID in your bundle's plist file as described here: <https://developers.facebook.com/docs/getting-started/facebook-sdk-for-ios/>

Warning

You must invoke this in order to use the Facebook functionality in `Parse`.

6.34.2.5 + (void) initializeFacebookWithApplicationLaunchOptions: (NSDictionary *) launchOptions

Initializes `Parse` Facebook Utils.

You must provide your Facebook application ID as the value for FacebookAppID in your bundle's plist file as described here: <https://developers.facebook.com/docs/getting-started/facebook-sdk-for-ios/>

Warning

You must invoke this in order to use the Facebook functionality in `Parse`.

Parameters

<i>launchOptions</i>	The launchOptions as passed to [UIApplicationDelegate application:didFinishLaunchingWithOptions:].
----------------------	--

6.34.2.6 + (void) initializeFacebookWithURLSchemeSuffix: (PF_NULLABLE NSString *) *urlSchemeSuffix*

Initializes the Facebook singleton.

You must provide your Facebook application ID as the value for FacebookAppID in your bundle's plist file as described here: <https://developers.facebook.com/docs/getting-started/facebook-sdk-for-ios/>

Warning

You must invoke this in order to use the Facebook functionality in [Parse](#).

Parameters

<i>urlSchemeSuffix</i>	The URL suffix for this application - used when multiple applications with the same Facebook application ID may be on the same device.
------------------------	--

6.34.2.7 + (void) initializeWithApplicationId: ("Use instead.") *PARSE_DEPRECATED*[PFFacebookUtils initializeFacebook]

Initializes the Facebook singleton.

Warning

You must invoke this in order to use the Facebook functionality in [Parse](#).

Parameters

<i>appId</i>	The Facebook application id that you are using with your Parse application.
--------------	---

Deprecated Please use [PFFacebookUtils initializeFacebook] instead.

6.34.2.8 + (void) initializeWithApplicationId: (PF_NULLABLE NSString *) *appId* urlSchemeSuffix:("Use instead.") *PARSE_DEPRECATED*[PFFacebookUtils initializeFacebookWithURLSchemeSuffix:]

Initializes the Facebook singleton.

Warning

You must invoke this in order to use the Facebook functionality in [Parse](#).

Parameters

<i>appId</i>	The Facebook application id that you are using with your Parse application.
<i>urlSchemeSuffix</i>	The URL suffix for this application - used when multiple applications with the same

Deprecated Please use [PFFacebookUtils initializeFacebookWithURLSchemeSuffix:] instead.

6.34.2.9 + (BOOL) isLinkedWithUser: (PFUser *) *user*

Whether the user has their account linked to Facebook.

Parameters

<i>user</i>	User to check for a facebook link. The user must be logged in on this device.
-------------	---

Returns

YES if the user has their account linked to Facebook, otherwise NO.

6.34.2.10 + (BOOL) isLinkedWithUser: (PFUser *) user

Whether the user has their account linked to Facebook.

Parameters

<i>user</i>	User to check for a facebook link. The user must be logged in on this device.
-------------	---

Returns

YES if the user has their account linked to Facebook, otherwise NO.

6.34.2.11 + (void) linkUser: (PFUser *) user facebookId:(NSString *) facebookId accessToken:(NSString *) accessToken expirationDate:(NSDate *) expirationDate block:(PF_NULLABLE PFBooleanResultBlock) block

Links Facebook to an existing <PFUser> *asynchronously*.

Allows you to handle user login to Facebook, then provide authentication data to link the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Facebook.
<i>facebookId</i>	The id of the Facebook user being linked
<i>accessToken</i>	The access token for the user's session
<i>expirationDate</i>	The expiration date for the access token
<i>block</i>	The block to execute. It should have the following argument signature: ^ (BOOL success, NSError *error).

6.34.2.12 + (void) linkUser: (PFUser *) user facebookId:(NSString *) facebookId accessToken:(NSString *) accessToken expirationDate:(NSDate *) expirationDate target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.34.2.13 + (void) linkUser: (PFUser *) user permissions:(NSString *) permissions PARSE_DEPRECATED[PFFacebookUtils linkUserInBackground:permissions:]

Links Facebook to an existing <PFUser> *asynchronously*.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Facebook.
<i>permissions</i>	The permissions required for Facebook log in. This passed to the authorize method on the Facebook instance.

Deprecated Please use `[PFFacebookUtils linkUserInBackground:permissions:]` instead.

6.34.2.14 + (void) linkUser: (PFUser *) user permissions:(PF_NULLABLE NSArray *) permissions
block:(PF_NULLABLE PFBooleanResultBlock) block

Links Facebook to an existing <PFUser> *asynchronously*.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Facebook.
<i>permissions</i>	The permissions required for Facebook log in. This passed to the authorize method on the Facebook instance.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL ↵ L *success, NSError *error).

6.34.2.15 + (void) linkUser: (PFUser *) user permissions:(PF_NULLABLE NSArray *) permissions
target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.34.2.16 + (BFTask *) linkUserInBackground: (PFUser *) user facebookId:(NSString *) facebookId accessToken:(NSString *) accessToken expirationDate:(NSDate *) expirationDate

Links Facebook to an existing <PFUser> *asynchronously*.

Allows you to handle user login to Facebook, then provide authentication data to link the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Facebook.
<i>facebookId</i>	The id of the Facebook user being linked
<i>accessToken</i>	The access token for the user's session
<i>expirationDate</i>	The expiration date for the access token

Returns

The task, that encapsulates the work being done.

6.34.2.17 + (BFTask *) linkUserInBackground: (PFUser *) user permissions:(PF_NULLABLE NSArray *) permissions

Links Facebook to an existing <PFUser> *asynchronously*.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Facebook.
<i>permissions</i>	The permissions required for Facebook log in. This passed to the authorize method on the Facebook instance.

Returns

The task, that encapsulates the work being done.

6.34.2.18 + (BFTask *) linkUserInBackground: (PFUser *) user withAccessToken:(FBSDKAccessToken *) accessToken

Asynchronously links Facebook Access Token to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the *user*.

Parameters

<i>user</i>	User to link to Facebook.
<i>accessToken</i>	An instance of <code>FBSDKAccessToken</code> to use.

Returns

The task that will have a `result` set to `@YES` if operation succeeds.

6.34.2.19 + (void) linkUserInBackground: (PFUser *) *user* withAccessToken:(FBSDKAccessToken *) *accessToken* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously links Facebook Access Token to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the `user`.

Parameters

<i>user</i>	User to link to Facebook.
<i>accessToken</i>	An instance of <code>FBSDKAccessToken</code> to use.
<i>block</i>	The block to execute when the linking completes. It should have the following signature: <code>^(BOOL succeeded, NSError *error)</code> .

6.34.2.20 + (BFTask *) linkUserInBackground: (PFUser *) *user* withPublishPermissions:(NSArray *) *permissions*

Asynchronously links Facebook with publish permissions to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the `user`.

Parameters

<i>user</i>	User to link to Facebook.
<i>permissions</i>	Array of publish permissions to use.

Returns

The task that will have a `result` set to `@YES` if operation succeeds.

6.34.2.21 + (void) linkUserInBackground: (PFUser *) *user* withPublishPermissions:(NSArray *) *permissions* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously links Facebook with publish permissions to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the `user`.

Parameters

<i>user</i>	User to link to Facebook.
<i>permissions</i>	Array of publish permissions to use.
<i>block</i>	The block to execute when the linking completes. It should have the following signature: <code>^(BOOL succeeded, NSError *error)</code> .

6.34.2.22 + (BFTask *) linkUserInBackground: (PFUser *) *user* withReadPermissions:(NSArray *) *permissions*

Asynchronously links Facebook with read permissions to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the `user`.

Parameters

<code>user</code>	User to link to Facebook.
<code>permissions</code>	Array of read permissions to use when logging in with Facebook.

Returns

The task that will have a `result` set to @YES if operation succeeds.

6.34.2.23 + (void) linkUserInBackground: (PFUser *) *user* withReadPermissions:(PF_NULLABLE NSArray *) *permissions* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously links Facebook with read permissions to an existing <PFUser>.

This method delegates to the Facebook SDK to authenticate the user, and then automatically links the account to the <PFUser>. It will also save any unsaved changes that were made to the `user`.

Parameters

<code>user</code>	User to link to Facebook.
<code>permissions</code>	Array of read permissions to use.
<code>block</code>	The block to execute when the linking completes. It should have the following signature: ^ (BOOL succeeded, NSError *error).

6.34.2.24 + (BFTask *) loginInBackgroundWithAccessToken: (FBSDKAccessToken *) *accessToken*

Asynchronously logs in a user using given Facebook Access Token.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<code>accessToken</code>	An instance of FBSDKAccessToken to use when logging in.
--------------------------	---

Returns

The task that has will a have `result` set to <PFUser> if operation succeeds.

6.34.2.25 + (void) loginInBackgroundWithAccessToken: (FBSDKAccessToken *) *accessToken* block:(PF_NULLABLE PFUserResultBlock) *block*

Asynchronously logs in a user using given Facebook Access Token.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<code>accessToken</code>	An instance of FBSDKAccessToken to use when logging in.
<code>block</code>	The block to execute when the log in completes. It should have the following signature: ^ (PFUser *user, NSError *error).

6.34.2.26 + (BFTask *) logInInBackgroundWithPublishPermissions: (PF_NULLABLE NSArray *) permissions

Asynchronously logs in a user using Facebook with publish permissions.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<i>permissions</i>	Array of publish permissions to use.
--------------------	--------------------------------------

Returns

The task that has will have `result` set to <PFUser> if operation succeeds.

6.34.2.27 + (void) logInInBackgroundWithPublishPermissions: (PF_NULLABLE NSArray *) permissions block:(PF_NULLABLE PFUserResultBlock) block

Asynchronously logs in a user using Facebook with publish permissions.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<i>permissions</i>	Array of publish permissions to use.
<i>block</i>	The block to execute when the log in completes. It should have the following signature: ^(PFUser *user, NSError *error).

6.34.2.28 + (BFTask *) logInInBackgroundWithReadPermissions: (PF_NULLABLE NSArray *) permissions

Asynchronously logs in a user using Facebook with read permissions.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<i>permissions</i>	Array of read permissions to use.
--------------------	-----------------------------------

Returns

The task that has will have `result` set to <PFUser> if operation succeeds.

6.34.2.29 + (void) logInInBackgroundWithReadPermissions: (PF_NULLABLE NSArray *) permissions block:(PF_NULLABLE PFUserResultBlock) block

Asynchronously logs in a user using Facebook with read permissions.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<i>permissions</i>	Array of read permissions to use.
--------------------	-----------------------------------

<i>block</i>	The block to execute when the log in completes. It should have the following signature: \wedge (PFUser *user, NSError *error).
--------------	--

6.34.2.30 + (void) loginWithFacebookId: (NSString *) *facebookId* accessToken:(NSString *) *accessToken* expirationDate:(NSDate *) *expirationDate* block:(PF_NULLABLE PFUserResultBlock) *block*

Logs in a user using Facebook *asynchronously*.

Allows you to handle user login to Facebook, then provide authentication data to log in (or create, in the case where it is a new user) the <PFUser>.

Parameters

<i>facebookId</i>	The id of the Facebook user being linked
<i>accessToken</i>	The access token for the user's session
<i>expirationDate</i>	The expiration date for the access token
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (PFUser *user, NSError *error).

6.34.2.31 + (void) loginWithFacebookId: (NSString *) *facebookId* accessToken:(NSString *) *accessToken* expirationDate:(NSDate *) *expirationDate* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.34.2.32 + (BFTask *) loginWithFacebookIdInBackground: (NSString *) *facebookId* accessToken:(NSString *) *accessToken* expirationDate:(NSDate *) *expirationDate*

Logs in a user using Facebook *asynchronously*.

Allows you to handle user login to Facebook, then provide authentication data to log in (or create, in the case where it is a new user) the <PFUser>.

Parameters

<i>facebookId</i>	The id of the Facebook user being linked
<i>accessToken</i>	The access token for the user's session
<i>expirationDate</i>	The expiration date for the access token

Returns

The task, that encapsulates the work being done.

6.34.2.33 + (void) loginWithPermissions: (PF_NULLABLE NSArray *) *permissions* block:(PF_NULLABLE PFUserResultBlock) *block*

Logs in a user using Facebook.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>. *user* is going to be non-nil if the authentication was succesful. *error* is set if there was an error. *user* and *error* are both nil - if the user cancelled authentication by switching back to the application.

Parameters

<i>permissions</i>	The permissions required for Facebook log in. This passed to the authorize method on the Facebook instance.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (PFUser *user, NSError *error).

6.34.2.34 + (void) **loginWithPermissions:** (PF_NULLABLE NSArray *) *permissions* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.34.2.35 + (BFTask *) **loginWithPermissionsInBackground:** (PF_NULLABLE NSArray *) *permissions*

Logs in a user using Facebook *asynchronously*.

This method delegates to the Facebook SDK to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Parameters

<i>permissions</i>	The permissions required for Facebook log in. This passed to the authorize method on the Facebook instance.
--------------------	---

Returns

The task, that encapsulates the work being done.

6.34.2.36 + (void) **reauthorizeUser:** (PFUser *) *user* withPublishPermissions:(PF_NULLABLE NSArray *) *permissions* audience:(FBSessionDefaultAudience) *audience* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Requests new Facebook publish permissions for the given user *asynchronously*.

The user will be saved as part of this operation.

Warning

This may prompt the user to reauthorize the application.

Parameters

<i>user</i>	User to request new permissions for. The user must be linked to Facebook.
<i>permissions</i>	The new publishing permissions to request.
<i>audience</i>	The default audience for publishing permissions to request.
<i>block</i>	The block to execute. It should have the following argument signature: ^ (BOOL↔ L succeeded, NSError *error).

6.34.2.37 + (void) **reauthorizeUser:** (PFUser *) *user* withPublishPermissions:(PF_NULLABLE NSArray *) *permissions* audience:(FBSessionDefaultAudience) *audience* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.34.2.38 + (BFTask *) **reauthorizeUserInBackground:** (PFUser *) *user* withPublishPermissions:(PF_NULLABLE NSArray *) *permissions* audience:(FBSessionDefaultAudience) *audience*

Requests new Facebook publish permissions for the given user *asynchronously*.

The user will be saved as part of this operation.

Warning

This may prompt the user to reauthorize the application.

Parameters

<i>user</i>	User to request new permissions for. The user must be linked to Facebook.
<i>permissions</i>	The new publishing permissions to request.
<i>audience</i>	The default audience for publishing permissions to request.

Returns

The task, that encapsulates the work being done.

6.34.2.39 + (PF_NULLABLE FBSession *) session

Gets the Facebook session for the current user.

6.34.2.40 + (void) setFacebookLoginBehavior: (FBSessionLoginBehavior) behavior

Sets the default login behavior to be used when logging in or linking users with Facebook.

Default is to allow Facebook Login, with fallback to Inline Facebook Login.

Parameters

<i>behavior</i>	The behavior to set.
-----------------	----------------------

6.34.2.41 + (BOOL) unlinkUser: (PFUser *) user

Unlinks the <PFUser> from a Facebook account *synchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
-------------	-------------------------------

Returns

Returns YES if the unlink was successful, otherwise NO.

6.34.2.42 + (BOOL) unlinkUser: (PFUser *) user error:(NSError **) error

Unlinks the <PFUser> from a Facebook account *synchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
<i>error</i>	Error object to set on error.

Returns

Returns YES if the unlink was successful, otherwise NO.

6.34.2.43 + (BFTask *) unlinkUserInBackground: (PFUser *) user

Unlinks the <PFUser> from a Facebook account *asynchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
-------------	-------------------------------

Returns

The task, that encapsulates the work being done.

6.34.2.44 + (BFTask *) unlinkUserInBackground: (PFUser *) user

Unlinks the <PFUser> from a Facebook account *asynchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
-------------	-------------------------------

Returns

The task, that encapsulates the work being done.

6.34.2.45 + (void) unlinkUserInBackground: (PFUser *) user block:(PF_NULLABLE PFBooleanResultBlock) block

Unlinks the <PFUser> from a Facebook account *asynchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
<i>block</i>	The block to execute. It should have the following argument signature: ^ (BOOL↵ L succeeded, NSError *error).

6.34.2.46 + (void) unlinkUserInBackground: (PFUser *) user block:(PF_NULLABLE PFBooleanResultBlock) block

Unlinks the <PFUser> from a Facebook account *asynchronously*.

Parameters

<i>user</i>	User to unlink from Facebook.
<i>block</i>	The block to execute. It should have the following argument signature: ^ (BOOL↵ L succeeded, NSError *error).

6.34.2.47 + (void) unlinkUserInBackground: (PFUser *) user target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

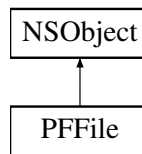
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseFacebookUtils.↵
framework/Headers/PFFacebookUtils.h

6.35 PFFile Class Reference

```
#import <PFFile.h>
```

Inheritance diagram for PFFile:



Instance Methods

Interrupting a Transfer

- (void) - [cancel](#)

Creating a PFFile

- NSString * [name](#)
- NSString * [url](#)
- (instancetype) + [fileWithData:](#)
- (instancetype) + [fileWithName:data:](#)
- (instancetype) + [fileWithName:contentsAtPath:](#)
- (instancetype) + [fileWithName:data:contentType:](#)
- (instancetype) + [fileWithData:contentType:](#)

Storing Data with Parse

- BOOL [isDirty](#)
- (BOOL) - [save](#)
- (BOOL) - [save:](#)
- (BFTask *) - [saveInBackground](#)
- (void) - [saveInBackgroundWithBlock:](#)
- (void) - [saveInBackgroundWithBlock:progressBlock:](#)
- (void) - [saveInBackgroundWithTarget:selector:](#)

Getting Data from Parse

- BOOL [isDataAvailable](#)
- (PF_NULLABLE NSData *) - [getData](#)
- (PF_NULLABLE NSInputStream *) - [getDataStream](#)
- (PF_NULLABLE NSData *) - [getData:](#)
- (PF_NULLABLE NSInputStream *) - [getDataStream:](#)
- (BFTask *) - [getDataInBackground](#)
- (BFTask *) - [getDataStreamInBackground](#)
- (void) - [getDataInBackgroundWithBlock:](#)
- (void) - [getDataStreamInBackgroundWithBlock:](#)
- (void) - [getDataInBackgroundWithBlock:progressBlock:](#)
- (void) - [getDataStreamInBackgroundWithBlock:progressBlock:](#)
- (void) - [getDataInBackgroundWithTarget:selector:](#)

6.35.1 Detailed Description

[PFFile](#) represents a file of binary data stored on the [Parse](#) servers. This can be a image, video, or anything else that an application needs to reference in a non-relational way.

Definition at line 23 of file PFFile.h.

6.35.2 Method Documentation

6.35.2.1 - (void) cancel

Cancels the current request (upload or download of file).

6.35.2.2 + (instancetype) fileWithData: (NSData *) data

Creates a file with given data. A name will be assigned to it by the server.

Parameters

<i>data</i>	The contents of the new PFFile .
-------------	--

Returns

A new [PFFile](#).

6.35.2.3 + (instancetype) fileWithData: (NSData *) data contentType:(PF_NULLABLE NSString *) contentType

Creates a file with given data and content type.

Parameters

<i>data</i>	The contents of the new PFFile .
<i>contentType</i>	Represents MIME type of the data.

Returns

A new [PFFile](#) object.

6.35.2.4 + (instancetype) fileName: (PF_NULLABLE NSString *) name contentsAtPath:(NSString *) path

Creates a file with the contents of another file.

Parameters

<i>name</i>	The name of the new PFFile . The file name must begin with an alphanumeric character, and consist of alphanumeric characters, periods, spaces, underscores, or dashes.
<i>path</i>	The path to the file that will be uploaded to Parse .

6.35.2.5 + (instancetype) fileName: (PF_NULLABLE NSString *) name data:(NSData *) data

Creates a file with given data and name.

Parameters

<i>name</i>	The name of the new PFFile . The file name must begin with an alphanumeric character, and consist of alphanumeric characters, periods, spaces, underscores, or dashes.
<i>data</i>	The contents of the new PFFile .

Returns

A new [PFFile](#) object.

6.35.2.6 + (instancetype) [initWithName:](#) (PF_NULLABLE NSString *) *name* data:(NSData *) *data* contentType:(PF_NULLABLE NSString *) *contentType*

Creates a file with given data, name and content type.

Parameters

<i>name</i>	The name of the new PFFile . The file name must begin with an alphanumeric character, and consist of alphanumeric characters, periods, spaces, underscores, or dashes.
<i>data</i>	The contents of the new PFFile .
<i>contentType</i>	Represents MIME type of the data.

Returns

A new [PFFile](#) object.

6.35.2.7 - (PF_NULLABLE NSData *) [getData](#)

Synchronously gets the data from cache if available or fetches its contents from the network.

Returns

The `NSData` object containing file data. Returns `nil` if there was an error in fetching.

6.35.2.8 - (PF_NULLABLE NSData *) [getData:](#) (NSError **) *error*

Synchronously gets the data from cache if available or fetches its contents from the network. Sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

The `NSData` object containing file data. Returns `nil` if there was an error in fetching.

6.35.2.9 - (BFTask *) [getDataInBackground](#)

This method is like `<getData>` but avoids ever holding the entire [PFFile](#) contents in memory at once.

This can help applications with many large files avoid memory warnings.

See also

- [getData](#)

Returns

A stream containing the data. Returns `nil` if there was an error in fetching.

6.35.2.10 - (void) getDataInBackgroundWithBlock: (PF_NULLABLE PFDataResultBlock) block

Asynchronously gets the data from cache if available or fetches its contents from the network.

Parameters

<i>block</i>	The block should have the following argument signature: <code>^(NSData *result, NSError *error)</code>
--------------	--

**6.35.2.11 - (void) getDataInBackgroundWithBlock: (PF_NULLABLE PFDataResultBlock) resultBlock
progressBlock:(PF_NULLABLE PFProgressBlock) progressBlock**

Asynchronously gets the data from cache if available or fetches its contents from the network.

This method will execute the `progressBlock` periodically with the percent progress. `progressBlock` will get called with 100 before `resultBlock` is called.

Parameters

<i>resultBlock</i>	The block should have the following argument signature: <code>(NSData *result, NSError *error)</code>
<i>progressBlock</i>	The block should have the following argument signature: <code>(int percentDone)</code>

6.35.2.12 - (void) getDataInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector
6.35.2.13 - (PF_NULLABLE NSInputStream *) getDataStream

This method is like `<getData>` but avoids ever holding the entire `PFFile` contents in memory at once.

This can help applications with many large files avoid memory warnings.

Returns

A stream containing the data. Returns `nil` if there was an error in fetching.

6.35.2.14 - (PF_NULLABLE NSInputStream *) getDataStream: (NSError **) error

This method is like `<getData>` but avoids ever holding the entire `PFFile` contents in memory at once.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

A stream containing the data. Returns `nil` if there was an error in fetching.

6.35.2.15 - (BFTask *) getDataStreamInBackground

This method is like `<getDataInBackground>` but avoids ever holding the entire `PFFile` contents in memory at once.

This can help applications with many large files avoid memory warnings.

Returns

The task, that encapsulates the work being done.

6.35.2.16 - (void) getDataStreamInBackgroundWithBlock: (PF_NULLABLE PFDataStreamResultBlock) block

This method is like <getDataInBackgroundWithBlock:> but avoids ever holding the entire [PFFile](#) contents in memory at once.

This can help applications with many large files avoid memory warnings.

Parameters

<i>block</i>	The block should have the following argument signature: (NSInputStream *result, NSError *error)
--------------	---

6.35.2.17 - (void) getDataStreamInBackgroundWithBlock: (PF_NULLABLE PFDataStreamResultBlock) resultBlock progressBlock:(PF_NULLABLE PFProgressBlock) progressBlock

This method is like <getDataInBackgroundWithBlock:progressBlock:> but avoids ever holding the entire [PFFile](#) contents in memory at once.

This can help applications with many large files avoid memory warnings.

Parameters

<i>resultBlock</i>	The block should have the following argument signature: ^ (NSInputStream *result, NSError *error).
<i>progressBlock</i>	The block should have the following argument signature: ^ (int percentDone).

6.35.2.18 - (BOOL) save

Saves the file *synchronously*.

Returns

Returns whether the save succeeded.

6.35.2.19 - (BOOL) save: (NSError **) error

Saves the file *synchronously* and sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns whether the save succeeded.

6.35.2.20 - (BFTask *) saveInBackground

Saves the file *asynchronously*.

Returns

The task, that encapsulates the work being done.

6.35.2.21 - (void) saveInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) *block*

Saves the file *asynchronously* and executes the given block.

Parameters

<i>block</i>	The block should have the following argument signature: \wedge (BOOL succeeded, NSError *error).
--------------	--

6.35.2.22 - (void) saveInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) *block*
 progressBlock:(PF_NULLABLE PFProgressBlock) *progressBlock*

Saves the file *asynchronously* and executes the given block.

This method will execute the *progressBlock* periodically with the percent progress. *progressBlock* will get called with 100 before *resultBlock* is called.

Parameters

<i>block</i>	The block should have the following argument signature: \wedge (BOOL succeeded, NSError *error)
<i>progressBlock</i>	The block should have the following argument signature: \wedge (int percentDone)

6.35.2.23 - (void) saveInBackgroundWithTarget: (PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.35.3 Property Documentation

6.35.3.1 - (BOOL) *isDataAvailable* [read], [atomic], [assign]

Whether the data is available in memory or needs to be downloaded.

Definition at line 169 of file PFFile.h.

6.35.3.2 - (BOOL) *isDirty* [read], [nonatomic], [assign]

Whether the file has been uploaded for the first time.

Definition at line 107 of file PFFile.h.

6.35.3.3 - (NSString*) *name* [read], [nonatomic], [copy]

The name of the file.

Before the file is saved, this is the filename given by the user. After the file is saved, that name gets prefixed with a unique identifier.

Definition at line 93 of file PFFile.h.

6.35.3.4 - (NSString*) *url* [read], [nonatomic], [copy]

The url of the file.

Definition at line 98 of file PFFile.h.

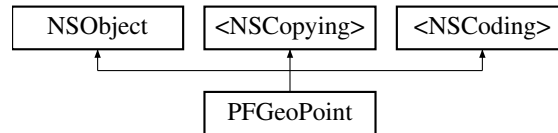
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFFile.h

6.36 PFGeoPoint Class Reference

```
#import <PFGeoPoint.h>
```

Inheritance diagram for PFGeoPoint:



Instance Methods

Calculating Distance

- (double) - [distanceInRadiansTo:](#)
- (double) - [distanceInMilesTo:](#)
- (double) - [distanceInKilometersTo:](#)

Class Methods

Creating a Geo Point

- (PFGeoPoint *) + [geoPoint](#)
- (PFGeoPoint *) + [geoPointWithLocation:](#)
- (PFGeoPoint *) + [geoPointWithLatitude:longitude:](#)
- (void) + [geoPointForCurrentLocationInBackground:](#)

Properties

Controlling Position

- double [latitude](#)
- double [longitude](#)

6.36.1 Detailed Description

[PFGeoPoint](#) may be used to embed a latitude / longitude point as the value for a key in a [<PFObject>](#). It could be used to perform queries in a geospatial manner using [<\[PFQuery whereKey:nearGeoPoint:\]>](#).

Currently, instances of [<PFObject>](#) may only have one key associated with a [PFGeoPoint](#) type.

Definition at line 28 of file [PFGeoPoint.h](#).

6.36.2 Method Documentation

6.36.2.1 - (double) distanceInKilometersTo: (PF_NULLABLE PFGeoPoint *) point

Get distance in kilometers from this point to specified point.

Parameters

<i>point</i>	PFGeoPoint that represents the location of other point.
--------------	---

Returns

Distance in kilometers between the receiver and `point`.

6.36.2.2 - (double) distanceInMilesTo: (PF_NULLABLE PFGeoPoint *) *point*

Get distance in miles from this point to specified point.

Parameters

<i>point</i>	PFGeoPoint that represents the location of other point.
--------------	---

Returns

Distance in miles between the receiver and `point`.

6.36.2.3 - (double) distanceInRadiansTo: (PF_NULLABLE PFGeoPoint *) *point*

Get distance in radians from this point to specified point.

Parameters

<i>point</i>	PFGeoPoint that represents the location of other point.
--------------	---

Returns

Distance in radians between the receiver and `point`.

6.36.2.4 + (PFGeoPoint *) geoPoint

Create a [PFGeoPoint](#) object. Latitude and longitude are set to 0.0.

Returns

Returns a new [PFGeoPoint](#).

6.36.2.5 + (void) geoPointForCurrentLocationInBackground: (PF_NULLABLE PFGeoPointResultBlock) *geoPointHandler*

Fetches the current device location and executes a block with a new [PFGeoPoint](#) object.

Parameters

<i>geoPointHandler</i>	A block which takes the newly created PFGeoPoint as an argument. It should have the following argument signature: <code>^(PFGeoPoint *geoPoint, NSError *error)</code>
------------------------	--

6.36.2.6 + (PFGeoPoint *) geoPointWithLatitude: (double) *latitude* longitude:(double) *longitude*

Create a new [PFGeoPoint](#) object with the specified latitude and longitude.

Parameters

<i>latitude</i>	Latitude of point in degrees.
<i>longitude</i>	Longitude of point in degrees.

Returns

New point object with specified latitude and longitude.

6.36.2.7 + (PFGeoPoint *) geoPointWithLocation: (PF_NULLABLE CLLocation *) location

Creates a new [PFGeoPoint](#) object for the given `CLLocation`, set to the location's coordinates.

Parameters

<i>location</i>	Instance of <code>CLLocation</code> , with set latitude and longitude.
-----------------	--

Returns

Returns a new [PFGeoPoint](#) at specified location.

6.36.3 Property Documentation

6.36.3.1 - (double) latitude [read], [write], [nonatomic], [assign]

Latitude of point in degrees. Valid range is from -90.0 to 90.0 .

Definition at line 75 of file `PFGeoPoint.h`.

6.36.3.2 - (double) longitude [read], [write], [nonatomic], [assign]

Longitude of point in degrees. Valid range is from -180.0 to 180.0 .

Definition at line 80 of file `PFGeoPoint.h`.

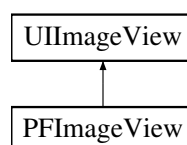
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/P↵
FGeoPoint.h`

6.37 PFIImageView Class Reference

```
#import <PFIImageView.h>
```

Inheritance diagram for `PFIImageView`:

**Instance Methods**

- [\(BFTask *\) - loadInBackground](#)
- [\(void\) - loadInBackground:](#)
- [\(void\) - loadInBackground:progressBlock:](#)

Properties

- [PFFile](#) * `file`

6.37.1 Detailed Description

An image view that downloads and displays remote image stored on [Parse](#)'s server.

Definition at line 36 of file PFIImageView.h.

6.37.2 Method Documentation

6.37.2.1 - (BFTask *) loadInBackground

Initiate downloading of the remote image.

Once the download completes, the remote image will be displayed.

Returns

The task, that encapsulates the work being done.

6.37.2.2 - (void) loadInBackground: (PFUI_NULLABLE PFIImageViewImageResultBlock) completion

Initiate downloading of the remote image.

Once the download completes, the remote image will be displayed.

Parameters

<i>completion</i>	the completion block.
-------------------	-----------------------

6.37.2.3 - (void) loadInBackground: (PFUI_NULLABLE PFIImageViewImageResultBlock) completion progressBlock:(PFUI_NULLABLE void(^)(int percentDone)) progressBlock

Initiate downloading of the remote image.

Once the download completes, the remote image will be displayed.

Parameters

<i>completion</i>	the completion block.
<i>progressBlock</i>	called with the download progress as the image is being downloaded. Will be called with a value of 100 before the completion block is called.

6.37.3 Property Documentation

6.37.3.1 - (PFFile*) file [read], [write], [nonatomic], [strong]

The remote file on [Parse](#)'s server that stores the image.

Warning

Note that the download does not start until `<loadInBackground:>` is called.

Definition at line 43 of file PFIImageView.h.

The documentation for this class was generated from the following file:

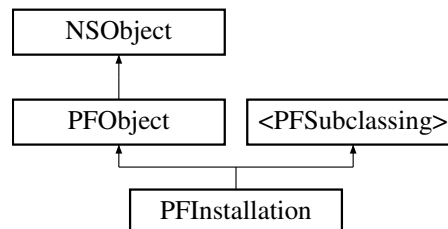
- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/BoundHeaders/PFImageView.h](#)

[Breaker/ParseUI.framework/↔](#)

6.38 PFInstallation Class Reference

```
#import <PFInstallation.h>
```

Inheritance diagram for PFInstallation:



Class Methods

Accessing the Current Installation

- (instancetype) + [currentInstallation](#)

Querying for Installations

- (PF_NULLABLE PFQuery *) + [query](#)

Installation Properties

- NSString * [deviceType](#)
- NSString * [installationId](#)
- NSString * [deviceToken](#)
- NSInteger [badge](#)
- NSString * [timeZone](#)
- NSArray * [channels](#)
- (void) - [setDeviceTokenFromData:](#)

Additional Inherited Members

6.38.1 Detailed Description

A [Parse](#) Framework Installation Object that is a local representation of an installation persisted to the [Parse](#) cloud. This class is a subclass of a [PFObject](#), and retains the same functionality of a [PFObject](#), but also extends it with installation-specific fields and related immutability and validity checks.

A valid [PFInstallation](#) can only be instantiated via [<\[PFInstallation currentInstallation\]>](#) because the required identifier fields are readonly. The [<timeZone>](#) and [<badge>](#) fields are also readonly properties which are automatically updated to match the device's time zone and application badge when the [PFInstallation](#) is saved, thus these fields might not reflect the latest device state if the installation has not recently been saved.

[PFInstallation](#) objects which have a valid [<deviceToken>](#) and are saved to the [Parse](#) cloud can be used to target push notifications.

This class is currently for iOS only. There is no [PFInstallation](#) for [Parse](#) applications running on OS X, because they cannot receive push notifications.

Definition at line 36 of file PFInstallation.h.

6.38.2 Method Documentation

6.38.2.1 + (instancetype) currentInstallation

Gets the currently-running installation from disk and returns an instance of it.

If this installation is not stored on disk, returns a [PFInstallation](#) with <deviceType> and <installationId> fields set to those of the current installation.

Returns

Returns a [PFInstallation](#) that represents the currently-running installation.

6.38.2.2 + (PF_NULLABLE PFQuery *) query

Creates a <PFQuery> for [PFInstallation](#) objects.

Only the following types of queries are allowed for installations:

- [query getObjectWithId:<value>]
- [query whereKey:@"installationId" equalTo:<value>]
- [query whereKey:@"installationId" matchesKey:<key in query> inQuery:<query>]

You can add additional query conditions, but one of the above must appear as a top-level AND clause in the query.

Reimplemented from [<PFSubclassing>](#).

6.38.2.3 - (void) setDeviceTokenFromData: (PF_NULLABLE NSData *) deviceTokenData

Sets the device token string property from an NSData-encoded token.

Parameters

<i>deviceTokenData</i>	A token that identifies the device.
------------------------	-------------------------------------

6.38.3 Property Documentation

6.38.3.1 - (NSInteger) badge [read], [write], [nonatomic], [assign]

The badge for the [PFInstallation](#).

Definition at line 75 of file PFInstallation.h.

6.38.3.2 - (NSArray*) channels [read], [write], [nonatomic], [strong]

The channels for the [PFInstallation](#).

Definition at line 85 of file PFInstallation.h.

6.38.3.3 - (NSString*) **deviceToken** [read], [write], [nonatomic], [strong]

The device token for the [PFInstallation](#).

Definition at line 70 of file PFInstallation.h.

6.38.3.4 - (NSString*) **deviceType** [read], [nonatomic], [strong]

The device type for the [PFInstallation](#).

Definition at line 60 of file PFInstallation.h.

6.38.3.5 - (NSString*) **installationId** [read], [nonatomic], [strong]

The installationId for the [PFInstallation](#).

Definition at line 65 of file PFInstallation.h.

6.38.3.6 - (NSString*) **timeZone** [read], [nonatomic], [strong]

The name of the time zone for the [PFInstallation](#).

Definition at line 80 of file PFInstallation.h.

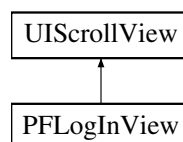
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFInstallation.h](#)

6.39 PFLoginView Class Reference

```
#import <PFLoginView.h>
```

Inheritance diagram for PFLoginView:



Properties

Customizing the Logo

- UIView * [logo](#)
The logo. By default, it is the [Parse](#) logo.

Configure Username Behaviour

- BOOL [emailAsUsername](#)

Creating Log In View

- UIViewController * [presentingViewController](#)
- (instancetype) - [initWithFields:](#)

Log In Elements

- PFLogInFields [fields](#)
- [PFTextField](#) * [usernameField](#)
- [PFTextField](#) * [passwordField](#)
- UIButton * [passwordForgottenButton](#)
- UIButton * [loginButton](#)
- UIButton * [facebookButton](#)
- UIButton * [twitterButton](#)
- UIButton * [signUpButton](#)
- UIButton * [dismissButton](#)
- (UILabel *externalLoginLabel) - [__attribute__](#)
- (UILabel *signUpLabel) - [__attribute__](#)

6.39.1 Detailed Description

The [PFLogInView](#) class provides a standard log in interface for authenticating a <PFUser>.

Definition at line 65 of file PFLogInView.h.

6.39.2 Method Documentation

6.39.2.1 - (UILabel* externalLoginLabel) [__attribute__](#) ([PARSE_UI_DEPRECATED](#)("This property is deprecated and will always be nil."))

The facebook/twitter login label.

Deprecated This property is deprecated and will always be nil.

6.39.2.2 - (UILabel* signUpLabel) [__attribute__](#) ([PARSE_UI_DEPRECATED](#)("This property is deprecated and will always be nil."))

The sign up label.

Deprecated This property is deprecated and will always be nil.

6.39.2.3 - (instancetype) initWithFields: (PFLogInFields) *fields*

Initializes the view with the specified log in elements.

Parameters

<i>fields</i>	A bitmask specifying the log in elements which are enabled in the view
---------------	--

Returns

An initialized [PFLogInView](#) object or `nil` if the object couldn't be created.

See also

[PFLogInFields](#)

6.39.3 Property Documentation

6.39.3.1 - (UIButton*) **dismissButton** [read],[nonatomic],[strong]

It is `nil` if the element is not enabled.

Definition at line 154 of file PFLogInView.h.

6.39.3.2 - (BOOL) **emailAsUsername** [read],[write],[nonatomic],[assign]

If email should be used to log in, instead of username

By default, this is set to `NO`.

Definition at line 105 of file PFLogInView.h.

6.39.3.3 - (UIButton*) **facebookButton** [read],[nonatomic],[strong]

The Facebook button. It is `nil` if the element is not enabled.

Definition at line 139 of file PFLogInView.h.

6.39.3.4 - (PFLogInFields) **fields** [read],[nonatomic],[assign]

The bitmask which specifies the enabled log in elements in the view.

Definition at line 114 of file PFLogInView.h.

6.39.3.5 - (UIButton*) **loginButton** [read],[nonatomic],[strong]

The log in button. It is `nil` if the element is not enabled.

Definition at line 134 of file PFLogInView.h.

6.39.3.6 - (UIView*) **logo** [read],[write],[nonatomic],[strong]

The logo. By default, it is the [Parse](#) logo.

Definition at line 94 of file PFLogInView.h.

6.39.3.7 - (PFTextField*) **passwordField** [read],[nonatomic],[strong]

The password text field. It is `nil` if the element is not enabled.

Definition at line 124 of file PFLogInView.h.

6.39.3.8 - (UIButton*) **passwordForgottenButton** [read],[nonatomic],[strong]

The password forgotten button. It is `nil` if the element is not enabled.

Definition at line 129 of file PFLogInView.h.

6.39.3.9 - (UIViewController*) **presentingViewController** [read],[write],[nonatomic],[weak]

The view controller that will present this view.

Used to lay out elements correctly when the presenting view controller has translucent elements.

Definition at line 87 of file PFLoginView.h.

6.39.3.10 `-(UIButton*) signUpButton` `[read], [nonatomic], [strong]`

The sign up button. It is `nil` if the element is not enabled.

Definition at line 149 of file PFLoginView.h.

6.39.3.11 `-(UIButton*) twitterButton` `[read], [nonatomic], [strong]`

The Twitter button. It is `nil` if the element is not enabled.

Definition at line 144 of file PFLoginView.h.

6.39.3.12 `-(UITextField*) usernameField` `[read], [nonatomic], [strong]`

The username text field. It is `nil` if the element is not enabled.

Definition at line 119 of file PFLoginView.h.

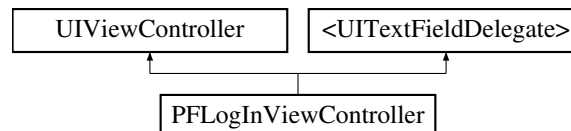
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFLoginView.h`

6.40 PFLoginViewController Class Reference

```
#import <PFLoginViewController.h>
```

Inheritance diagram for PFLoginViewController:



Properties

Configuring Log In Elements

- `PFLoginFields` `fields`
- `PFLoginView` * `loginView`

Configuring Log In Behaviors

- `id` < `PFLoginViewControllerDelegate` > `delegate`
- `NSArray` * `facebookPermissions`
- `PFSignUpViewController` * `signUpController`
- `BOOL` `emailAsUsername`

6.40.1 Detailed Description

The `PFLoginViewController` class presents and manages a standard authentication interface for logging in a `<PFUser>`.

Definition at line 36 of file PFLoginViewController.h.

6.40.2 Property Documentation

6.40.2.1 - (id<PFLogInViewControllerDelegate>) **delegate** [read], [write], [nonatomic], [weak]

The delegate that responds to the control events of [PFLogInViewController](#).

See also

[PFLogInViewControllerDelegate](#)

Definition at line 66 of file PFLogInViewController.h.

6.40.2.2 - (BOOL) **emailAsUsername** [read], [write], [nonatomic], [assign]

Whether to prompt for the email as username on the login view.

If set to YES, we'll prompt for the email in the username field. This property value propagates to the attached <signUpController>. By default, this is set to NO.

Definition at line 90 of file PFLogInViewController.h.

6.40.2.3 - (NSArray*) **facebookPermissions** [read], [write], [nonatomic], [copy]

The facebook permissions that Facebook log in requests for.

If unspecified, the default is basic facebook permissions.

Definition at line 73 of file PFLogInViewController.h.

6.40.2.4 - (PFLogInFields) **fields** [read], [write], [nonatomic], [assign]

A bitmask specifying the log in elements which are enabled in the view.

See also

[PFLogInFields](#)

Definition at line 47 of file PFLogInViewController.h.

6.40.2.5 - (PFLogInView*) **loginView** [read], [nonatomic], [strong]

The log in view. It contains all the enabled log in elements.

See also

[PFLogInView](#)

Definition at line 55 of file PFLogInViewController.h.

6.40.2.6 - (PFSignUpViewController*) **signUpController** [read], [write], [nonatomic], [strong]

The sign up controller if sign up is enabled.

Use this to configure the sign up view, and the transition animation to the sign up view. The default is a sign up view with a username, a password, a dismiss button and a sign up button.

Definition at line 81 of file PFLogInViewController.h.

The documentation for this class was generated from the following file:

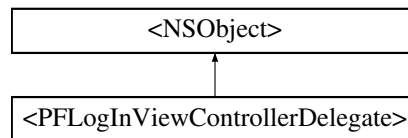
- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound
Headers/PFLogInViewController.h

Breaker/ParseUI.framework/↔

6.41 <PFLogInViewControllerDelegate> Protocol Reference

```
#import <PFLogInViewController.h>
```

Inheritance diagram for <PFLogInViewControllerDelegate>:



Instance Methods

Customizing Behavior

- (BOOL) - [loginViewController:shouldBeginLoginWithUsername:password:](#)

Responding to Actions

- (void) - [loginViewController:didLoginUser:](#)
- (void) - [loginViewController:didFailToLoginWithError:](#)
- (void) - [loginViewControllerDidCancelLogin:](#)

6.41.1 Detailed Description

The [PFLogInViewControllerDelegate](#) protocol defines methods a delegate of a <PFLogInViewController> should implement. All methods of this protocol are optional.

Definition at line 122 of file PFLogInViewController.h.

6.41.2 Method Documentation

6.41.2.1 - (void) loginViewController: (PFLogInViewController *) loginController didFailToLoginWithError:(PFUI_NUL↔
lable NSError *) error

Sent to the delegate when the log in attempt fails.

Parameters

<i>loginController</i>	The login view controller where login failed.
<i>error</i>	NSError object representing the error that occurred.

6.41.2.2 - (void) loginViewController: (PFLogInViewController *) loginController didLoginUser:(PFUser *) user

Sent to the delegate when a <PFUser> is logged in.

Parameters

<i>loginController</i>	The login view controller where login finished.
<i>user</i>	<PFUser> object that is a result of the login.

6.41.2.3 - (BOOL) loginViewController: (PFLogInViewController *) loginController shouldBeginLoginWithUsername:(NSString *) username password:(NSString *) password

Sent to the delegate to determine whether the log in request should be submitted to the server.

Parameters

<i>loginController</i>	The login view controller that is requesting the data.
<i>username</i>	the username the user tries to log in with.
<i>password</i>	the password the user tries to log in with.

Returns

A BOOL indicating whether the log in should proceed.

6.41.2.4 - (void) loginViewControllerDidCancelLogin: (PFLogInViewController *) loginController

Sent to the delegate when the log in screen is cancelled.

Parameters

<i>loginController</i>	The login view controller where login was cancelled.
------------------------	--

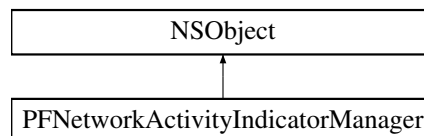
The documentation for this protocol was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFLogInViewController.h

6.42 PFNetworkActivityIndicatorManager Class Reference

```
#import <PFNetworkActivityIndicatorManager.h>
```

Inheritance diagram for PFNetworkActivityIndicatorManager:



Instance Methods

- (void) - [incrementActivityCount](#)
- (void) - [decrementActivityCount](#)

Class Methods

- (instancetype) + [sharedManager](#)

Properties

- BOOL [enabled](#)
- BOOL [networkActivityIndicatorVisible](#)
- NSInteger [networkActivityCount](#)

6.42.1 Detailed Description

[PFNetworkActivityIndicatorManager](#) manages the state of the network activity indicator in the status bar. When enabled, it will start managing the network activity indicator in the status bar, according to the network operations that are performed by [Parse](#) SDK.

The number of active requests is incremented or decremented like a stack or a semaphore, the activity indicator will animate, as long as the number is greater than zero.

Definition at line 21 of file [PFNetworkActivityIndicatorManager.h](#).

6.42.2 Method Documentation

6.42.2.1 - (void) decrementActivityCount

Decrements the number of active network requests.

If this number becomes zero after decrementing, this will stop animating network activity indicator in the status bar.

6.42.2.2 - (void) incrementActivityCount

Increments the number of active network requests.

If this number was zero before incrementing, this will start animating network activity indicator in the status bar.

6.42.2.3 + (instancetype) sharedManager

Returns the shared network activity indicator manager object for the system.

Returns

The systemwide network activity indicator manager.

6.42.3 Property Documentation

6.42.3.1 - (BOOL) enabled `[read], [write], [nonatomic], [assign]`

A Boolean value indicating whether the manager is enabled. If YES - the manager will start managing the status bar network activity indicator, according to the network operations that are performed by [Parse](#) SDK. The default value is YES.

Definition at line 29 of file [PFNetworkActivityIndicatorManager.h](#).

6.42.3.2 - (NSInteger) networkActivityCount `[read], [nonatomic], [assign]`

The value that indicates current network activities count.

Definition at line 39 of file [PFNetworkActivityIndicatorManager.h](#).

6.42.3.3 - (BOOL) networkActivityIndicatorVisible [read], [nonatomic], [assign]

A Boolean value indicating whether the network activity indicator is currently displayed in the status bar.

Definition at line 34 of file PFNetworkActivityIndicatorManager.h.

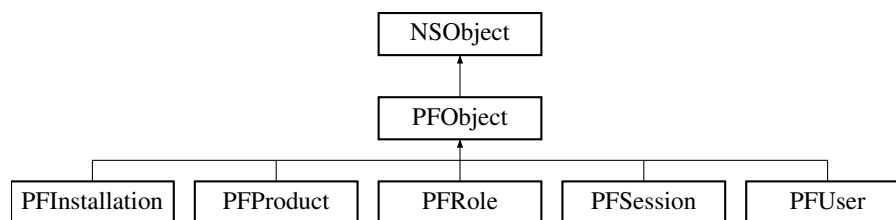
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFNetworkActivityIndicatorManager.h](#)

6.43 PFObjcet Class Reference

```
#import <PFObjcet.h>
```

Inheritance diagram for PFObjcet:



Instance Methods

Accessors

- (PF_NULLABLE_S id) - [objectForKey:](#)
- (void) - [setObject:forKey:](#)
- (void) - [removeObjectForKey:](#)
- (PF_NULLABLE_S id) - [objectForKeyedSubscript:](#)
- (void) - [setObject:forKeyedSubscript:](#)
- (PFRelation *) - [relationForKey:](#)
- (PFRelation *) - [relationforKey:](#)

Array Accessors

- (void) - [addObject:forKey:](#)
- (void) - [addObjectsFromArray:forKey:](#)
- (void) - [addUniqueObject:forKey:](#)
- (void) - [addUniqueObjectsFromArray:forKey:](#)
- (void) - [removeObject:forKey:](#)
- (void) - [removeObjectsInArray:forKey:](#)

Increment

- (void) - [incrementKey:](#)
- (void) - [incrementKey:byAmount:](#)

Saving Objects

- (BOOL) - [save](#)
- (BOOL) - [save:](#)
- (BFTask *) - [saveInBackground](#)
- (void) - [saveInBackgroundWithBlock:](#)
- (void) - [saveInBackgroundWithTarget:selector:](#)

- (BFTask *) - [saveEventually](#)
- (void) - [saveEventually:](#)

Getting an Object

- (BOOL) - [isDataAvailable](#)
- (void) - [fetch](#)
- (void) - [fetch:](#)
- (PF_NULLABLE PFOBJECT *) - [fetchIfNeeded](#)
- (PF_NULLABLE PFOBJECT *) - [fetchIfNeeded:](#)
- (BFTask *) - [fetchInBackground](#)
- (void) - [fetchInBackgroundWithBlock:](#)
- (void) - [fetchInBackgroundWithTarget:selector:](#)
- (BFTask *) - [fetchIfNeededInBackground](#)
- (void) - [fetchIfNeededInBackgroundWithBlock:](#)
- (void) - [fetchIfNeededInBackgroundWithTarget:selector:](#)

Fetching From Local Datastore

- (void) - [fetchFromLocalDatastore](#)
- (void) - [fetchFromLocalDatastore:](#)
- (BFTask *) - [fetchFromLocalDatastoreInBackground](#)
- (void) - [fetchFromLocalDatastoreInBackgroundWithBlock:](#)

Deleting an Object

- (BOOL) - [delete](#)
- (BOOL) - [delete:](#)
- (BFTask *) - [deleteInBackground](#)
- (void) - [deleteInBackgroundWithBlock:](#)
- (void) - [deleteInBackgroundWithTarget:selector:](#)
- (BFTask *) - [deleteEventually](#)

Dirtiness

- (BOOL) - [isDirty](#)
- (BOOL) - [isDirtyForKey:](#)

Pinning

- (BOOL) - [pin](#)
- (BOOL) - [pin:](#)
- (BOOL) - [pinWithName:](#)
- (BOOL) - [pinWithName:error:](#)
- (BFTask *) - [pinInBackground](#)
- (void) - [pinInBackgroundWithBlock:](#)
- (BFTask *) - [pinInBackgroundWithName:](#)
- (void) - [pinInBackgroundWithName:block:](#)

Unpinning

- (BOOL) - [unpin](#)
- (BOOL) - [unpin:](#)
- (BOOL) - [unpinWithName:](#)
- (BOOL) - [unpinWithName:error:](#)
- (BFTask *) - [unpinInBackground](#)
- (void) - [unpinInBackgroundWithBlock:](#)
- (BFTask *) - [unpinInBackgroundWithName:](#)
- (void) - [unpinInBackgroundWithName:block:](#)

Class Methods

Saving Many Objects

- (BOOL) + [saveAll:](#)
- (BOOL) + [saveAll:error:](#)
- (BFTask *) + [saveAllInBackground:](#)
- (void) + [saveAllInBackground:block:](#)
- (void) + [saveAllInBackground:target:selector:](#)

Deleting Many Objects

- (BOOL) + [deleteAll:](#)
- (BOOL) + [deleteAll:error:](#)
- (BFTask *) + [deleteAllInBackground:](#)
- (void) + [deleteAllInBackground:block:](#)
- (void) + [deleteAllInBackground:target:selector:](#)

Getting Many Objects

- (void) + [fetchAll:](#)
- (void) + [fetchAll:error:](#)
- (void) + [fetchAllIfNeeded:](#)
- (void) + [fetchAllIfNeeded:error:](#)
- (BFTask *) + [fetchAllInBackground:](#)
- (void) + [fetchAllInBackground:block:](#)
- (void) + [fetchAllInBackground:target:selector:](#)
- (BFTask *) + [fetchAllIfNeededInBackground:](#)
- (void) + [fetchAllIfNeededInBackground:block:](#)
- (void) + [fetchAllIfNeededInBackground:target:selector:](#)

Pinning Many Objects

- (BOOL) + [pinAll:](#)
- (BOOL) + [pinAll:error:](#)
- (BOOL) + [pinAll:withName:](#)
- (BOOL) + [pinAll:withName:error:](#)
- (BFTask *) + [pinAllInBackground:](#)
- (void) + [pinAllInBackground:block:](#)
- (BFTask *) + [pinAllInBackground:withName:](#)
- (void) + [pinAllInBackground:withName:block:](#)

Unpinning Many Objects

- (BOOL) + [unpinAllObjects](#)
- (BOOL) + [unpinAllObjects:](#)
- (BOOL) + [unpinAllObjectsWithName:](#)
- (BOOL) + [unpinAllObjectsWithName:error:](#)
- (BFTask *) + [unpinAllObjectsInBackground](#)
- (void) + [unpinAllObjectsInBackgroundWithBlock:](#)
- (BFTask *) + [unpinAllObjectsInBackgroundWithName:](#)
- (void) + [unpinAllObjectsInBackgroundWithName:block:](#)
- (BOOL) + [unpinAll:](#)
- (BOOL) + [unpinAll:error:](#)
- (BOOL) + [unpinAll:withName:](#)
- (BOOL) + [unpinAll:withName:error:](#)
- (BFTask *) + [unpinAllInBackground:](#)
- (void) + [unpinAllInBackground:block:](#)
- (BFTask *) + [unpinAllInBackground:withName:](#)
- (void) + [unpinAllInBackground:withName:block:](#)

Protected Attributes

- BOOL [dirty](#)
- NSMutableArray *[PF_NULLABLE_S operationSetQueue](#)
- NSMutableDictionary *[PF_NULLABLE_S estimatedData](#)

Methods for Subclasses

- (instancetype) + [object](#)
- (instancetype) + [objectWithoutDataWithObjectId:](#)
- (void) + [registerSubclass](#)
- ([PF_NULLABLE PFQuery *](#)) + [query](#)
- ([PF_NULLABLE PFQuery *](#)) + [queryWithPredicate:](#)
- (instancetype) - [init](#)

Creating a PFOBJECT

- (instancetype) + [objectWithClassName:](#)
- (instancetype) + [objectWithoutDataWithClassName:objectId:](#)
- ([PFOBJECT *](#)) + [objectWithClassName:dictionary:](#)
- (instancetype) - [initWithClassName:](#)

Managing Object Properties

- NSString * [parseClassName](#)
- NSString * [objectId](#)
- NSDate * [updatedAt](#)
- NSDate * [createdAt](#)
- [PFAcl](#) * [ACL](#)
- (NSArray *) - [allKeys](#)

6.43.1 Detailed Description

The [PFOBJECT](#) class is a local representation of data persisted to the [Parse](#) cloud. This is the main class that is used to interact with objects in your app.

Definition at line 33 of file [PFOBJECT.h](#).

6.43.2 Method Documentation

6.43.2.1 - (void) addObject: (id) *object* forKey:(NSString *) *key*

Adds an object to the end of the array associated with a given key.

Parameters

<i>object</i>	The object to add.
<i>key</i>	The key.

6.43.2.2 - (void) addObjectsFromArray: (NSArray *) *objects* forKey:(NSString *) *key*

Adds the objects contained in another array to the end of the array associated with a given key.

Parameters

<i>objects</i>	The array of objects to add.
<i>key</i>	The key.

6.43.2.3 - (void) addUniqueObject: (id) *object* forKey:(NSString *) *key*

Adds an object to the array associated with a given key, only if it is not already present in the array.

The position of the insert is not guaranteed.

Parameters

<i>object</i>	The object to add.
<i>key</i>	The key.

6.43.2.4 - (void) addUniqueObjectsFromArray: (NSArray *) *objects* forKey:(NSString *) *key*

Adds the objects contained in another array to the array associated with a given key, only adding elements which are not already present in the array.

The position of the insert is not guaranteed.

Parameters

<i>objects</i>	The array of objects to add.
<i>key</i>	The key.

6.43.2.5 - (NSArray *) allKeys

Returns an array of the keys contained in this object.

This does not include `createdAt`, `updatedAt`, `authData`, or `objectId`. It does include things like `username` and `ACL`.

6.43.2.6 - (BOOL) delete

Synchronously deletes the [PFObject](#).

Returns

Returns whether the delete succeeded.

6.43.2.7 - (BOOL) delete: (NSError **) *error*

Synchronously deletes the [PFObject](#) and sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns whether the delete succeeded.

6.43.2.8 + (BOOL) deleteAll: (PF_NULLABLE NSArray *) *objects*

Synchronously deletes a collection of objects all at once.

Parameters

<i>objects</i>	The array of objects to delete.
----------------	---------------------------------

Returns

Returns whether the delete succeeded.

6.43.2.9 + (BOOL) deleteAll: (PF_NULLABLE NSArray *) *objects* error:(NSError **) *error*

Synchronously deletes a collection of objects all at once and sets an error if necessary.

Parameters

<i>objects</i>	The array of objects to delete.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the delete succeeded.

6.43.2.10 + (BFTask *) deleteAllInBackground: (PF_NULLABLE NSArray *) *objects*

Deletes a collection of objects all at once asynchronously.

Parameters

<i>objects</i>	The array of objects to delete.
----------------	---------------------------------

Returns

The task that encapsulates the work being done.

6.43.2.11 + (void) deleteAllInBackground: (PF_NULLABLE NSArray *) *objects* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Deletes a collection of objects all at once *asynchronously* and executes the block when done.

Parameters

<i>objects</i>	The array of objects to delete.
<i>block</i>	The block to execute. It should have the following argument signature: $\wedge (BOOL \leftrightarrow L \text{ succeeded}, NSError *error)$.

6.43.2.12 + (void) deleteAllInBackground: (PF_NULLABLE NSArray *) *objects* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.43.2.13 - (BFTask *) deleteEventually

Deletes this object from the server at some unspecified time in the future, even if [Parse](#) is currently inaccessible.

Use this when you may not have a solid network connection, and don't need to know when the delete completes. If there is some problem with the object such that it can't be deleted, the request will be silently discarded.

Delete instructions made with this method will be stored locally in an on-disk cache until they can be transmitted to [Parse](#). They will be sent immediately if possible. Otherwise, they will be sent the next time a network connection is

available. Delete requests will persist even after the app is closed, in which case they will be sent the next time the app is opened. If more than 10MB of <saveEventually> or <deleteEventually> commands are waiting to be sent, subsequent calls to <saveEventually> or <deleteEventually> will cause old requests to be silently discarded until the connection can be re-established, and the queued requests can go through.

Returns

The task that encapsulates the work being done.

6.43.2.14 - (BFTask *) deleteInBackground

Deletes the [PFOBJECT](#) *asynchronously*.

Returns

The task that encapsulates the work being done.

6.43.2.15 - (void) deleteInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Deletes the [PFOBJECT](#) *asynchronously* and executes the given callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: $\wedge (BOOL \leftarrow L \text{ succeeded}, NSError *error)$.
--------------	---

6.43.2.16 - (void) deleteInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.43.2.17 - (void) fetch

Synchronously fetches the [PFOBJECT](#) with the current data from the server.

6.43.2.18 - (void) fetch: (NSError **) error

Synchronously fetches the [PFOBJECT](#) with the current data from the server and sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

6.43.2.19 + (void) fetchAll: (PF_NULLABLE NSArray *) objects

Synchronously fetches all of the [PFOBJECT](#) objects with the current data from the server.

Parameters

<i>objects</i>	The list of objects to fetch.
----------------	-------------------------------

6.43.2.20 + (void) fetchAll: (PF_NULLABLE NSArray *) objects error:(NSError **) error

Synchronously fetches all of the [PFOBJECT](#) objects with the current data from the server and sets an error if it occurs.

Parameters

<i>objects</i>	The list of objects to fetch.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

6.43.2.21 + (void) fetchAllIfNeeded: (PF_NULLABLE NSArray *) *objects*

Synchronously fetches all of the `PFObject` objects with the current data from the server.

Parameters

<i>objects</i>	The list of objects to fetch.
----------------	-------------------------------

6.43.2.22 + (void) fetchAllIfNeeded: (PF_NULLABLE NSArray *) *objects* error:(NSError **) *error*

Synchronously fetches all of the `PFObject` objects with the current data from the server and sets an error if it occurs.

Parameters

<i>objects</i>	The list of objects to fetch.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

6.43.2.23 + (BFTask *) fetchAllIfNeededInBackground: (PF_NULLABLE NSArray *) *objects*

Fetches all of the `PFObject` objects with the current data from the server *asynchronously*.

Parameters

<i>objects</i>	The list of objects to fetch.
----------------	-------------------------------

Returns

The task that encapsulates the work being done.

6.43.2.24 + (void) fetchAllIfNeededInBackground: (PF_NULLABLE NSArray *) *objects* block:(PF_NULLABLE PFArrayResultBlock) *block*

Fetches all of the `PFObjects` with the current data from the server *asynchronously* and calls the given block.

Parameters

<i>objects</i>	The list of objects to fetch.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (NSArray *objects, NSError *error).

6.43.2.25 + (void) fetchAllIfNeededInBackground: (PF_NULLABLE NSArray *) *objects* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*6.43.2.26 + (BFTask *) fetchAllInBackground: (PF_NULLABLE NSArray *) *objects*

Fetches all of the `PFObject` objects with the current data from the server *asynchronously*.

Parameters

<i>objects</i>	The list of objects to fetch.
----------------	-------------------------------

Returns

The task that encapsulates the work being done.

6.43.2.27 + (void) fetchAllInBackground: (PF_NULLABLE NSArray *) objects block:(PF_NULLABLE PFArrayResultBlock) block

Fetches all of the [PFOBJECT](#) objects with the current data from the server *asynchronously* and calls the given block.

Parameters

<i>objects</i>	The list of objects to fetch.
<i>block</i>	The block to execute. It should have the following argument signature: [^] (NSArray *objects, NSError *error).

6.43.2.28 + (void) fetchAllInBackground: (PF_NULLABLE NSArray *) objects target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.43.2.29 - (void) fetchFromLocalDatastore

Synchronously loads data from the local datastore into this object, if it has not been fetched from the server already.

6.43.2.30 - (void) fetchFromLocalDatastore: (NSError **) error

Synchronously loads data from the local datastore into this object, if it has not been fetched from the server already.

If the object is not stored in the local datastore, this `error` will be set to return `kPFCacheMiss`.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

6.43.2.31 - (BFTask *) fetchFromLocalDatastoreInBackground

Asynchronously loads data from the local datastore into this object, if it has not been fetched from the server already.

Returns

The task that encapsulates the work being done.

6.43.2.32 - (void) fetchFromLocalDatastoreInBackgroundWithBlock: (PF_NULLABLE PFOBJECTResultBlock) block

Asynchronously loads data from the local datastore into this object, if it has not been fetched from the server already.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: [^] (PFOBJECT *object, NSError *error).
--------------	--

6.43.2.33 - (PF_NULLABLE PFOBJECT *) fetchIfNeeded

Synchronously fetches the [PFOBJECT](#) data from the server if <isDataAvailable> is NO.

6.43.2.34 - (PF_NULLABLE PFOBJECT *) fetchIfNeeded: (NSError **) error

Synchronously fetches the [PFOBJECT](#) data from the server if <isDataAvailable> is NO.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

6.43.2.35 - (BFTask *) fetchIfNeededInBackground

Fetches the [PFOBJECT](#) data *asynchronously* if `isDataAvailable` is NO, then sets it as a result for the task.

Returns

The task that encapsulates the work being done.

6.43.2.36 - (void) fetchIfNeededInBackgroundWithBlock: (PF_NULLABLE PFOBJECTRESULTBLOCK) block

Fetches the [PFOBJECT](#) data *asynchronously* if <isDataAvailable> is NO, then calls the callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(PFOBJECT *object, NSError *error).</code>
--------------	--

6.43.2.37 - (void) fetchIfNeededInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.43.2.38 - (BFTask *) fetchInBackground

Fetches the [PFOBJECT](#) *asynchronously* and sets it as a result for the task.

Returns

The task that encapsulates the work being done.

6.43.2.39 - (void) fetchInBackgroundWithBlock: (PF_NULLABLE PFOBJECTRESULTBLOCK) block

Fetches the [PFOBJECT](#) *asynchronously* and executes the given callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(PFOBJECT *object, NSError *error).</code>
--------------	--

6.43.2.40 - (void) fetchInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.43.2.41 - (void) incrementKey: (NSString *) key

Increments the given key by 1.

Parameters

<i>key</i>	The key.
------------	----------

6.43.2.42 - (void) incrementKey: (NSString *) key byAmount:(NSNumber *) amount

Increments the given key by a number.

Parameters

<i>key</i>	The key.
<i>amount</i>	The amount to increment.

6.43.2.43 - (instancetype) init

Designated initializer for subclasses. This method can only be called on subclasses which conform to <PF↔Subclassing>. This method should not be overridden.

Provided by category [PFOBJECT\(Subclass\)](#).

6.43.2.44 - (instancetype) initWithClassName: (NSString *) newClassName

Initializes a new empty [PFOBJECT](#) instance with a class name.

Parameters

<i>newClassName</i>	A class name can be any alphanumeric string that begins with a letter. It represents an object in your app, like a 'User' or a 'Document'.
---------------------	--

Returns

Returns the object that is instantiated with the given class name.

6.43.2.45 - (BOOL) isDataAvailable

Gets whether the [PFOBJECT](#) has been fetched.

Returns

YES if the [PFOBJECT](#) is new or has been fetched or refreshed, otherwise NO.

6.43.2.46 - (BOOL) isDirty

Gets whether any key-value pair in this object (or its children) has been added/updated/removed and not saved yet.

Returns

Returns whether this object has been altered and not saved yet.

6.43.2.47 - (BOOL) isDirtyForKey: (NSString *) key

Get whether a value associated with a key has been added/updated/removed and not saved yet.

Parameters

<i>key</i>	The key to check for
------------	----------------------

Returns

Returns whether this key has been altered and not saved yet.

6.43.2.48 + (instancetype) object

Creates an instance of the registered subclass with this class's <parseClassName>.

This helps a subclass ensure that it can be subclassed itself. For example, [PFUser object] will return a MyUser object if MyUser is a registered subclass of PFUser. For this reason, [MyClass object] is preferred to [[MyClass alloc] init]. This method can only be called on subclasses which conform to PFSubclassing. A default implementation is provided by PFObject which should always be sufficient.

Provided by category PFObject(Subclass).

6.43.2.49 - (PF_NULLABLE_S id) objectForKey: (NSString *) key

Returns the value associated with a given key.

Parameters

<i>key</i>	The key for which to return the corresponding value.
------------	--

6.43.2.50 - (PF_NULLABLE_S id) objectForKeyedSubscript: (NSString *) key

Returns the value associated with a given key.

This method enables usage of literal syntax on PFObject. E.g. NSString *value = object[@"key"];

Parameters

<i>key</i>	The key for which to return the corresponding value.
------------	--

See also

- objectForKey:

6.43.2.51 + (instancetype) objectWithClassName: (NSString *) className

Creates a new PFObject with a class name.

Parameters

<i>className</i>	A class name can be any alphanumeric string that begins with a letter. It represents an object in your app, like a 'User' or a 'Document'.
------------------	--

Returns

Returns the object that is instantiated with the given class name.

6.43.2.52 + (PFOBJECT *) objectWithClassName: (NSString *) *className* dictionary:(PF_NULLABLE NSDictionary *) *dictionary*

Creates a new PFOBJECT with a class name, initialized with data constructed from the specified set of objects and keys.

Parameters

<i>className</i>	The object's class.
<i>dictionary</i>	An <code>NSDictionary</code> of keys and objects to set on the new <code>PFObjec</code> .

Returns

A `PFObjec` with the given class name and set with the given data.

6.43.2.53 + (instancetype) objectWithoutDataWithClassName: (NSString *) *className* objectId:(PF_NULLABLE NSString *) *objectId*

Creates a reference to an existing `PFObjec` for use in creating associations between `PFObjec`s.

Calling `<isDataAvailable>` on this object will return `NO` until `<fetchIfNeeded>` has been called. No network request will be made.

Parameters

<i>className</i>	The object's class.
<i>objectId</i>	The object id for the referenced object.

Returns

A `PFObjec` instance without data.

6.43.2.54 + (instancetype) objectWithoutDataWithObjectId: (PF_NULLABLE NSString *) *objectId*

Creates a reference to an existing `PFObjec` for use in creating associations between `PFObjec`s.

Calling `<isDataAvailable>` on this object will return `NO` until `<fetchIfNeeded>` or `<fetch>` has been called. This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by `PFObjec` which should always be sufficient. No network request will be made.

Parameters

<i>objectId</i>	The object id for the referenced object.
-----------------	--

Returns

An instance of `PFObjec` without data.

Provided by category `PFObjec(Subclass)`.

6.43.2.55 - (BOOL) pin

Synchronously stores the object and every object it points to in the local datastore, recursively, using a default pin name: `PFObjecDefaultPin`.

If those other objects have not been fetched from `Parse`, they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFObjec objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Returns

Returns whether the pin succeeded.

See also

- unpin:
PFOBJECTDefaultPin

6.43.2.56 - (BOOL) pin: (NSError **) error

Synchronously stores the object and every object it points to in the local datastore, recursively, using a default pin name: PFOBJECTDefaultPin.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a <PFQuery> that uses <[PFQuery fromLocalDatastore]>, or you can create an unfetched pointer with <[PFOBJECT objectWithoutDataWithClassName:objectId:> and then call <fetchFromLocalDatastore> on it.

Parameters

<i>error</i>	Pointer to an NSError that will be set if necessary.
--------------	--

Returns

Returns whether the pin succeeded.

See also

- unpin:
PFOBJECTDefaultPin

6.43.2.57 + (BOOL) pinAll: (PF_NULLABLE NSArray *) objects

Synchronously stores the objects and every object they point to in the local datastore, recursively, using a default pin name: PFOBJECTDefaultPin.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a <PFQuery> that uses <[PFQuery fromLocalDatastore]>, or you can create an unfetched pointer with [PFOBJECT objectWithoutDataWithClassName:objectId:] and then call fetchFromLocalDatastore: on it.

Parameters

<i>objects</i>	The objects to be pinned.
----------------	---------------------------

Returns

Returns whether the pin succeeded.

See also

+ unpinAll:
PFOBJECTDefaultPin

6.43.2.58 + (BOOL) pinAll: (PF_NULLABLE NSArray *) objects error:(NSError **) error

Synchronously stores the objects and every object they point to in the local datastore, recursively, using a default pin name: PFOBJECTDefaultPin.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a <PFQuery> that uses <[PFQuery fromLocalDatastore]>, or you can create an unfetched pointer with [PFOBJECT objectWithoutDataWithClassName:objectId:] and then call fetchFromLocalDatastore: on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the pin succeeded.

See also

+ [unpinAll:error:](#)
[PFObjecDefaultPin](#)

6.43.2.59 + (BOOL) pinAll: (PF_NULLABLE NSArray *) objects withName:(NSString *) name

Synchronously stores the objects and every object they point to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[P↔FQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `[PFObjec objectWithout↔DataWithClassName:objectId:]` and then call `fetchFromLocalDatastore:` on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>name</i>	The name of the pin.

Returns

Returns whether the pin succeeded.

See also

+ [unpinAll:withName:](#)

6.43.2.60 + (BOOL) pinAll: (PF_NULLABLE NSArray *) objects withName:(NSString *) name error:(NSError **) error

Synchronously stores the objects and every object they point to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[P↔FQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `[PFObjec objectWithout↔DataWithClassName:objectId:]` and then call `fetchFromLocalDatastore:` on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>name</i>	The name of the pin.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the pin succeeded.

See also

+ [unpinAll:withName:error:](#)

6.43.2.61 + (BFTask *) pinAllInBackground: (PF_NULLABLE NSArray *) objects

Asynchronously stores the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[P↔FQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `[PFOBJECT objectWithout↔DataWithClassName:objectId:]` and then call `fetchFromLocalDatastore:` on it.

Parameters

<i>objects</i>	The objects to be pinned.
----------------	---------------------------

Returns

The task that encapsulates the work being done.

See also

[+ unpinAllInBackground:
PFOBJECTDefaultPin](#)

6.43.2.62 + (void) pinAllInBackground: (PF_NULLABLE NSArray *) objects block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously stores the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[P↔FQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `[PFOBJECT objectWithout↔DataWithClassName:objectId:]` and then call `fetchFromLocalDatastore:` on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>block</i>	The block to execute. It should have the following argument signature: <code>^(BOOL↔L succeeded, NSError *error)</code> .

See also

[+ unpinAllInBackground:block:
PFOBJECTDefaultPin](#)

6.43.2.63 + (BFTask *) pinAllInBackground: (PF_NULLABLE NSArray *) objects withName:(NSString *) name

Asynchronously stores the objects and every object they point to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[P↔FQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `[PFOBJECT objectWithout↔DataWithClassName:objectId:]` and then call `fetchFromLocalDatastore:` on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>name</i>	The name of the pin.

Returns

The task that encapsulates the work being done.

See also

[+ unpinAllInBackground:withName:](#)

6.43.2.64 + (void) **pinAllInBackground:** (PF_NULLABLE NSArray *) *objects* withName:(NSString *) *name* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously stores the objects and every object they point to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a <PFQuery> that uses <[PFQuery fromLocalDatastore]>, or you can create an unfetched pointer with [PFObject objectWithoutDataWithClassName:objectId:] and then call fetchFromLocalDatastore: on it.

Parameters

<i>objects</i>	The objects to be pinned.
<i>name</i>	The name of the pin.
<i>block</i>	The block to execute. It should have the following argument signature: ^ (BOOL succeeded, NSError *error).

See also

[+ unpinAllInBackground:withName:block:](#)

6.43.2.65 - (BFTask *) **pinInBackground**

Asynchronously stores the object and every object it points to in the local datastore, recursively, using a default pin name: PFObjectDefaultPin.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a <PFQuery> that uses <[PFQuery fromLocalDatastore]>, or you can create an unfetched pointer with <[PFObject objectWithoutDataWithClassName:objectId:]> and then call <fetchFromLocalDatastore> on it.

Returns

The task that encapsulates the work being done.

See also

[- unpinInBackground](#)
[PFObjectDefaultPin](#)

6.43.2.66 - (void) pinInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Asynchronously stores the object and every object it points to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFOBJECT objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(BOOL ↵ L succeeded, NSError *error).</code>
--------------	--

See also

- [unpinInBackgroundWithBlock:](#)
`PFOBJECTDefaultPin`

6.43.2.67 - (BFTask *) pinInBackgroundWithName: (NSString *) name

Asynchronously stores the object and every object it points to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFOBJECT objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

The task that encapsulates the work being done.

See also

- [unpinInBackgroundWithName:](#)

6.43.2.68 - (void) pinInBackgroundWithName: (NSString *) name block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously stores the object and every object it points to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFOBJECT objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Parameters

<i>name</i>	The name of the pin.
<i>block</i>	The block to execute. It should have the following argument signature: <code>^(BOOL ↵ L succeeded, NSError *error).</code>

See also

- [unpinInBackgroundWithName:block:](#)

6.43.2.69 - (BOOL) pinWithName: (NSString *) name

Synchronously stores the object and every object it points to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFObject objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

Returns whether the pin succeeded.

See also

- [unpinWithName:](#)

6.43.2.70 - (BOOL) pinWithName: (NSString *) name error:(NSError **) error

Synchronously stores the object and every object it points to in the local datastore, recursively.

If those other objects have not been fetched from [Parse](#), they will not be stored. However, if they have changed data, all the changes will be retained. To get the objects back later, you can use a `<PFQuery>` that uses `<[PFQuery fromLocalDatastore]>`, or you can create an unfetched pointer with `<[PFObject objectWithoutDataWithClassName:objectId:]>` and then call `<fetchFromLocalDatastore>` on it.

Parameters

<i>name</i>	The name of the pin.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the pin succeeded.

See also

- [unpinWithName:](#)

6.43.2.71 + (PF_NULLABLE PFQuery *) query

Returns a query for objects of type `<parseClassName>`.

This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by `<PFObject>` which should always be sufficient.

Implemented in [PFInstallation](#).

Provided by category [PFObject\(Subclass\)](#).

6.43.2.72 + (PF_NULLABLE PFQuery *) queryWithPredicate: (PF_NULLABLE NSPredicate *) predicate

Returns a query for objects of type `<parseClassName>` with a given predicate.

A default implementation is provided by `<PFObject>` which should always be sufficient.

Warning

This method can only be called on subclasses which conform to `<PFSubclassing>`.

Parameters

<i>predicate</i>	The predicate to create conditions from.
------------------	--

Returns

An instance of `<PFQuery>`.

See also

[[PFQuery](#) queryWithClassName:predicate:]

Provided by category [PFOBJECT\(Subclass\)](#).

6.43.2.73 + (void) registerSubclass

Registers an Objective-C class for [Parse](#) to use for representing a given [Parse](#) class.

Once this is called on a [PFOBJECT](#) subclass, any [PFOBJECT](#) [Parse](#) creates with a class name that matches `[self parseClassName]` will be an instance of subclass. This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by [PFOBJECT](#) which should always be sufficient.

Provided by category [PFOBJECT\(Subclass\)](#).

6.43.2.74 - (PFRelation *) relationForKey: (NSString *) key

Returns the relation object associated with the given key.

Parameters

<i>key</i>	The key that the relation is associated with.
------------	---

6.43.2.75 - (PFRelation *) relationForKey: ("Please use -relationForKey: instead.") PARSE_DEPRECATED

Returns the relation object associated with the given key.

Parameters

<i>key</i>	The key that the relation is associated with.
------------	---

Deprecated Please use [[PFOBJECT](#) relationForKey:] instead.

6.43.2.76 - (void) removeObject: (id) object forKey:(NSString *) key

Removes all occurrences of an object from the array associated with a given key.

Parameters

<i>object</i>	The object to remove.
<i>key</i>	The key.

6.43.2.77 - (void) removeObjectForKey: (NSString *) *key*

Unsets a key on the object.

Parameters

<i>key</i>	The key.
------------	----------

6.43.2.78 - (void) removeObjectsinArray: (NSArray *) *objects* forKey:(NSString *) *key*

Removes all occurrences of the objects contained in another array from the array associated with a given key.

Parameters

<i>objects</i>	The array of objects to remove.
<i>key</i>	The key.

6.43.2.79 - (BOOL) save

Synchronously saves the [PFOBJECT](#).

Returns

Returns whether the save succeeded.

6.43.2.80 - (BOOL) save: (NSError **) *error*

Synchronously saves the [PFOBJECT](#) and sets an error if it occurs.

Parameters

<i>error</i>	Pointer to an NSError that will be set if necessary.
--------------	--

Returns

Returns whether the save succeeded.

6.43.2.81 + (BOOL) saveAll: (PF_NULLABLE NSArray *) *objects*

Saves a collection of objects **synchronously* all at once.

Parameters

<i>objects</i>	The array of objects to save.
----------------	-------------------------------

Returns

Returns whether the save succeeded.

6.43.2.82 + (BOOL) saveAll: (PF_NULLABLE NSArray *) *objects* error:(NSError **) *error*

Saves a collection of objects *synchronously* all at once and sets an error if necessary.

Parameters

<i>objects</i>	The array of objects to save.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the save succeeded.

6.43.2.83 + (BFTask *) saveAllInBackground: (PF_NULLABLE NSArray *) *objects*

Saves a collection of objects all at once *asynchronously*.

Parameters

<i>objects</i>	The array of objects to save.
----------------	-------------------------------

Returns

The task that encapsulates the work being done.

6.43.2.84 + (void) saveAllInBackground: (PF_NULLABLE NSArray *) *objects* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Saves a collection of objects all at once *asynchronously* and executes the block when done.

Parameters

<i>objects</i>	The array of objects to save.
<i>block</i>	The block to execute. It should have the following argument signature: $\wedge (BOOL \leftrightarrow L \text{ succeeded}, NSError *error)$.

6.43.2.85 + (void) saveAllInBackground: (PF_NULLABLE NSArray *) *objects* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.43.2.86 - (BFTask *) saveEventually

Saves this object to the server at some unspecified time in the future, even if [Parse](#) is currently inaccessible.

Use this when you may not have a solid network connection, and don't need to know when the save completes. If there is some problem with the object such that it can't be saved, it will be silently discarded. If the save completes successfully while the object is still in memory, then callback will be called.

Objects saved with this method will be stored locally in an on-disk cache until they can be delivered to [Parse](#). They will be sent immediately if possible. Otherwise, they will be sent the next time a network connection is available. Objects saved this way will persist even after the app is closed, in which case they will be sent the next time the app is opened. If more than 10MB of data is waiting to be sent, subsequent calls to `<saveEventually>` will cause old saves to be silently discarded until the connection can be re-established, and the queued objects can be saved.

Returns

The task that encapsulates the work being done.

6.43.2.87 - (void) saveEventually: (PF_NULLABLE PFBooleanResultBlock) *callback*

Saves this object to the server at some unspecified time in the future, even if [Parse](#) is currently inaccessible.

Use this when you may not have a solid network connection, and don't need to know when the save completes. If there is some problem with the object such that it can't be saved, it will be silently discarded. If the save completes successfully while the object is still in memory, then callback will be called.

Objects saved with this method will be stored locally in an on-disk cache until they can be delivered to [Parse](#). They will be sent immediately if possible. Otherwise, they will be sent the next time a network connection is available. Objects saved this way will persist even after the app is closed, in which case they will be sent the next time the app is opened. If more than 10MB of data is waiting to be sent, subsequent calls to `<saveEventually>` will cause old saves to be silently discarded until the connection can be re-established, and the queued objects can be saved.

Parameters

<i>callback</i>	The block to execute. It should have the following argument signature: \wedge (BOOL succeeded, NSError *error).
-----------------	---

6.43.2.88 - (BFTask *) saveInBackground

Saves the [PFObject](#) *asynchronously*.

Returns

The task that encapsulates the work being done.

6.43.2.89 - (void) saveInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Saves the [PFObject](#) *asynchronously* and executes the given callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL succeeded, NSError *error).
--------------	---

6.43.2.90 - (void) saveInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.43.2.91 - (void) setObject: (id) object forKey:(NSString *) key

Sets the object associated with a given key.

Parameters

<i>object</i>	The object for key. A strong reference to the object is maintained by PFObject . Raises an <code>NSInvalidArgumentException</code> if object is nil. If you need to represent a nil value - use <code>NSNull</code> .
<i>key</i>	The key for object. Raises an <code>NSInvalidArgumentException</code> if key is nil.

See also

- [setObject:forKeyedSubscript:](#)

6.43.2.92 - (void) setObject: (PF_NULLABLE_S id) object forKeyedSubscript:(NSString *) key

Returns the value associated with a given key.

This method enables usage of literal syntax on [PFObject](#). E.g. `object[@"key"] = @"value";`

Parameters

<i>object</i>	The object for key. A strong reference to the object is maintained by PFObject . Raises an <code>NSInvalidArgumentException</code> if object is nil. If you need to represent a nil value - use <code>NSNull</code> .
<i>key</i>	The key for object. Raises an <code>NSInvalidArgumentException</code> if key is nil.

See also

- [setObject:forKey:](#)

6.43.2.93 - (BOOL) unpin

Synchronously removes the object and every object it points to in the local datastore, recursively, using a default pin name: `PFObjectDefaultPin`.

Returns

Returns whether the unpin succeeded.

See also

- pin:
[PFObjectDefaultPin](#)

6.43.2.94 - (BOOL) unpin: (NSError **) error

Synchronously removes the object and every object it points to in the local datastore, recursively, using a default pin name: `PFObjectDefaultPin`.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns whether the unpin succeeded.

See also

- pin:
[PFObjectDefaultPin](#)

6.43.2.95 + (BOOL) unpinAll: (PF_NULLABLE NSArray *) objects

Synchronously removes the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFObjectDefaultPin`.

Parameters

<i>objects</i>	The objects.
----------------	--------------

Returns

Returns whether the unpin succeeded.

See also

+ pinAll:
[PFObjectDefaultPin](#)

6.43.2.96 + (BOOL) unpinAll: (PF_NULLABLE NSArray *) objects error:(NSError **) error

Synchronously removes the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFObjectDefaultPin`.

Parameters

<i>objects</i>	The objects.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the unpin succeeded.

See also

[+ pinAll:error:](#)
[PFOBJECTDefaultPin](#)

6.43.2.97 + (BOOL) unpinAll: (PF_NULLABLE NSArray *) *objects* withName:(NSString *) *name*

Synchronously removes the objects and every object they point to in the local datastore, recursively.

Parameters

<i>objects</i>	The objects.
<i>name</i>	The name of the pin.

Returns

Returns whether the unpin succeeded.

See also

[+ pinAll:withName:](#)

6.43.2.98 + (BOOL) unpinAll: (PF_NULLABLE NSArray *) *objects* withName:(NSString *) *name* error:(NSError **) *error*

Synchronously removes the objects and every object they point to in the local datastore, recursively.

Parameters

<i>objects</i>	The objects.
<i>name</i>	The name of the pin.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the unpin succeeded.

See also

[+ pinAll:withName:error:](#)

6.43.2.99 + (BFTask *) unpinAllInBackground: (PF_NULLABLE NSArray *) *objects*

Asynchronously removes the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

Parameters

<i>objects</i>	The objects.
----------------	--------------

Returns

The task that encapsulates the work being done.

See also

[+ pinAllInBackground:
PFObjectDefaultPin](#)

6.43.2.100 + (void) unpinAllInBackground: (PF_NULLABLE NSArray *) objects block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes the objects and every object they point to in the local datastore, recursively, using a default pin name: `PFObjectDefaultPin`.

Parameters

<i>objects</i>	The objects.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

See also

[+ pinAllInBackground:block:
PFObjectDefaultPin](#)

6.43.2.101 + (BFTask *) unpinAllInBackground: (PF_NULLABLE NSArray *) objects withName:(NSString *) name

Asynchronously removes the objects and every object they point to in the local datastore, recursively.

Parameters

<i>objects</i>	The objects.
<i>name</i>	The name of the pin.

Returns

The task that encapsulates the work being done.

See also

[+ pinAllInBackground:withName:](#)

6.43.2.102 + (void) unpinAllInBackground: (PF_NULLABLE NSArray *) objects withName:(NSString *) name block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes the objects and every object they point to in the local datastore, recursively.

Parameters

<i>objects</i>	The objects.
<i>name</i>	The name of the pin.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

See also

[+ pinAllInBackground:withName:block:](#)

6.43.2.103 + (BOOL) unpinAllObjects

Synchronously removes all objects in the local datastore using a default pin name: `PFOBJECTDefaultPin`.

Returns

Returns whether the unpin succeeded.

See also

[PFOBJECTDefaultPin](#)

6.43.2.104 + (BOOL) unpinAllObjects: (NSError **) error

Synchronously removes all objects in the local datastore using a default pin name: `PFOBJECTDefaultPin`.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns whether the unpin succeeded.

See also

[PFOBJECTDefaultPin](#)

6.43.2.105 + (BFTask *) unpinAllObjectsInBackground

Asynchronously removes all objects in the local datastore using a default pin name: `PFOBJECTDefaultPin`.

Returns

The task that encapsulates the work being done.

See also

[PFOBJECTDefaultPin](#)

6.43.2.106 + (void) unpinAllObjectsInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes all objects in the local datastore using a default pin name: `PFOBJECTDefaultPin`.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).
--------------	---

See also

[PFObjectDefaultPin](#)

6.43.2.107 + (BFTask *) unpinAllObjectsInBackgroundWithName: (NSString *) name

Asynchronously removes all objects with the specified pin name.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

The task that encapsulates the work being done.

6.43.2.108 + (void) unpinAllObjectsInBackgroundWithName: (NSString *) name block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes all objects with the specified pin name.

Parameters

<i>name</i>	The name of the pin.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

6.43.2.109 + (BOOL) unpinAllObjectsWithName: (NSString *) name

Synchronously removes all objects with the specified pin name.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

Returns whether the unpin succeeded.

6.43.2.110 + (BOOL) unpinAllObjectsWithName: (NSString *) name error:(NSError **) error

Synchronously removes all objects with the specified pin name.

Parameters

<i>name</i>	The name of the pin.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the unpin succeeded.

6.43.2.111 - (BFTask *) unpinInBackground

Asynchronously removes the object and every object it points to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

Returns

The task that encapsulates the work being done.

See also

[- pinInBackground](#)
[PFOBJECTDefaultPin](#)

6.43.2.112 - (void) unpinInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes the object and every object it points to in the local datastore, recursively, using a default pin name: `PFOBJECTDefaultPin`.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).
--------------	---

See also

[- pinInBackgroundWithBlock:](#)
[PFOBJECTDefaultPin](#)

6.43.2.113 - (BFTask *) unpinInBackgroundWithName: (NSString *) name

Asynchronously removes the object and every object it points to in the local datastore, recursively.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

The task that encapsulates the work being done.

See also

[- pinInBackgroundWithName:](#)

6.43.2.114 - (void) unpinInBackgroundWithName: (NSString *) name block:(PF_NULLABLE PFBooleanResultBlock) block

Asynchronously removes the object and every object it points to in the local datastore, recursively.

Parameters

<i>name</i>	The name of the pin.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

See also

[- pinInBackgroundWithName:block:](#)

6.43.2.115 - (BOOL) unpinWithName: (NSString *) name

Synchronously removes the object and every object it points to in the local datastore, recursively.

Parameters

<i>name</i>	The name of the pin.
-------------	----------------------

Returns

Returns whether the unpin succeeded.

See also

[- pinWithName:](#)

6.43.2.116 - (BOOL) unpinWithName: (NSString *) name error:(NSError **) error

Synchronously removes the object and every object it points to in the local datastore, recursively.

Parameters

<i>name</i>	The name of the pin.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the unpin succeeded.

See also

[- pinWithName:error:](#)

6.43.3 Member Data Documentation**6.43.3.1 - (BOOL) dirty** `[protected]`

Definition at line 34 of file `PFObject.h`.

6.43.3.2 - (NSMutableDictionary* PF_NULLABLE_S) estimatedData `[protected]`

Definition at line 47 of file `PFObject.h`.

6.43.3.3 - (NSMutableArray* PF_NULLABLE_S) operationSetQueue `[protected]`

Definition at line 43 of file `PFObject.h`.

6.43.4 Property Documentation**6.43.4.1 - (PFAcl*) ACL** `[read], [write], [nonatomic], [strong]`

The ACL for this object.

Definition at line 127 of file `PFObject.h`.

6.43.4.2 - (NSDate*) **createdAt** [read],[nonatomic],[strong]

When the object was created.

Definition at line 122 of file PFOBJECT.h.

6.43.4.3 - (NSString*) **objectId** [read],[write],[nonatomic],[strong]

The id of the object.

Definition at line 112 of file PFOBJECT.h.

6.43.4.4 - (NSString*) **parseClassName** [read],[atomic],[strong]

The class name of the object.

Definition at line 107 of file PFOBJECT.h.

6.43.4.5 - (NSDate*) **updatedAt** [read],[nonatomic],[strong]

When the object was last updated.

Definition at line 117 of file PFOBJECT.h.

The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFOBJECT.h

6.44 PFOBJECT(Subclass) Category Reference

```
#import <PFOBJECT+Subclass.h>
```

Methods for Subclasses

- (instancetype) - **init**
- (instancetype) + **object**
- (instancetype) + **objectWithoutDataWithObjectId:**
- (void) + **registerSubclass**
- (PF_NULLABLE PFQuery *) + **query**
- (PF_NULLABLE PFQuery *) + **queryWithPredicate:**

6.44.1 Detailed Description

Definition at line 62 of file PFOBJECT+Subclass.h.

6.44.2 Method Documentation

6.44.2.1 - (instancetype) **init**

Designated initializer for subclasses. This method can only be called on subclasses which conform to <PFOBJECT+Subclassing>. This method should not be overridden.

Extends class **PFOBJECT**.

6.44.2.2 + (instancetype) object

Creates an instance of the registered subclass with this class's `<parseClassName>`.

This helps a subclass ensure that it can be subclassed itself. For example, `[PFUser object]` will return a `MyUser` object if `MyUser` is a registered subclass of `PFUser`. For this reason, `[MyClass object]` is preferred to `[[MyClass alloc] init]`. This method can only be called on subclasses which conform to `PFSubclassing`. A default implementation is provided by `PFOBJECT` which should always be sufficient.

Extends class `PFOBJECT`.

6.44.2.3 + (instancetype) objectWithoutDataWithObjectId: (PF_NULLABLE NSString *) objectId

Creates a reference to an existing `PFOBJECT` for use in creating associations between `PFOBJECTS`.

Calling `<isDataAvailable>` on this object will return `NO` until `<fetchIfNeeded>` or `<fetch>` has been called. This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by `PFOBJECT` which should always be sufficient. No network request will be made.

Parameters

<i>objectId</i>	The object id for the referenced object.
-----------------	--

Returns

An instance of `PFOBJECT` without data.

Extends class `PFOBJECT`.

6.44.2.4 + (PF_NULLABLE PFQuery *) query

Returns a query for objects of type `<parseClassName>`.

This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by `<PFOBJECT>` which should always be sufficient.

Extends class `PFOBJECT`.

6.44.2.5 + (PF_NULLABLE PFQuery *) queryWithPredicate: (PF_NULLABLE NSPredicate *) predicate

Returns a query for objects of type `<parseClassName>` with a given predicate.

A default implementation is provided by `<PFOBJECT>` which should always be sufficient.

Warning

This method can only be called on subclasses which conform to `<PFSubclassing>`.

Parameters

<i>predicate</i>	The predicate to create conditions from.
------------------	--

Returns

An instance of `<PFQuery>`.

See also

`[PFQuery queryWithClassName:predicate:]`

Extends class `PFOBJECT`.

6.44.2.6 + (void) registerSubclass

Registers an Objective-C class for [Parse](#) to use for representing a given [Parse](#) class.

Once this is called on a [PFObject](#) subclass, any [PFObject](#) [Parse](#) creates with a class name that matches `[self parseClassName]` will be an instance of subclass. This method can only be called on subclasses which conform to `<PFSubclassing>`. A default implementation is provided by [PFObject](#) which should always be sufficient.

Extends class [PFObject](#).

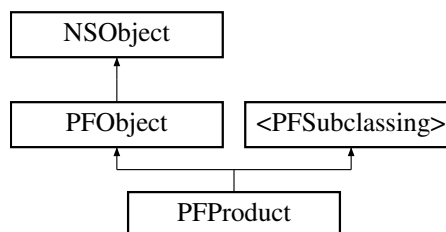
The documentation for this category was generated from the following file:

- [/Users/jscootrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFObject+Subclass.h](#)

6.45 PFProduct Class Reference

```
#import <PFProduct.h>
```

Inheritance diagram for PFProduct:



Properties

Product-specific Properties

- NSString * [productId](#)
- [PFFile](#) * [icon](#)
- NSString * [title](#)
- NSString * [subtitle](#)
- NSNumber * [order](#)
- NSString * [downloadName](#)

Additional Inherited Members

6.45.1 Detailed Description

The [PFProduct](#) class represents an in-app purchase product on the [Parse](#) server. By default, products can only be created via the Data Browser. Saving a [PFProduct](#) will result in error. However, the products' metadata information can be queried and viewed.

This class is currently for iOS only.

Definition at line 21 of file [PFProduct.h](#).

6.45.2 Property Documentation

6.45.2.1 - (NSString*) downloadName [read], [nonatomic], [strong]

The name of the associated download.

If there is no downloadable asset, it should be `nil`.

Definition at line 61 of file `PFProduct.h`.

6.45.2.2 - (PFFile*) icon [read], [write], [nonatomic], [strong]

The icon of the product.

Definition at line 37 of file `PFProduct.h`.

6.45.2.3 - (NSNumber*) order [read], [write], [nonatomic], [strong]

The order in which the product information is displayed in `<PFProductTableViewController>`.

The product with a smaller order is displayed earlier in the `<PFProductTableViewController>`.

Definition at line 54 of file `PFProduct.h`.

6.45.2.4 - (NSString*) productIdentifier [read], [write], [nonatomic], [strong]

The product identifier of the product.

This should match the product identifier in iTunes Connect exactly.

Definition at line 32 of file `PFProduct.h`.

6.45.2.5 - (NSString*) subtitle [read], [write], [nonatomic], [strong]

The subtitle of the product.

Definition at line 47 of file `PFProduct.h`.

6.45.2.6 - (NSString*) title [read], [write], [nonatomic], [strong]

The title of the product.

Definition at line 42 of file `PFProduct.h`.

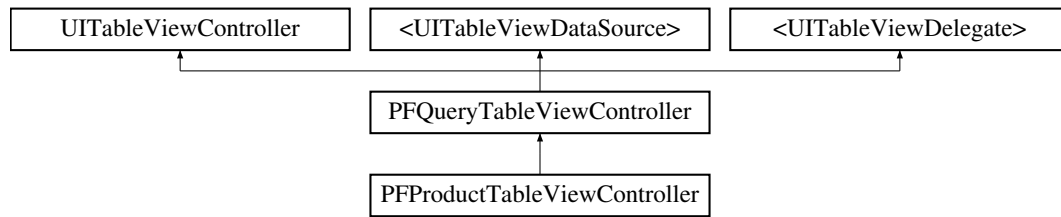
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFProduct.h`

6.46 PFProductTableViewController Class Reference

```
#import <PFProductTableViewController.h>
```

Inheritance diagram for `PFProductTableViewController`:



Instance Methods

- (instancetype) - [initWithStyle:](#)

Additional Inherited Members

6.46.1 Detailed Description

[PFProductTableViewController](#) displays in-app purchase products stored on [Parse](#). In addition to setting up in-app purchases in iTunes Connect, the app developer needs to register product information on [Parse](#), in the Product class.

Definition at line 34 of file `PFProductTableViewController.h`.

6.46.2 Method Documentation

6.46.2.1 - (instancetype) initWithStyle: (UITableViewStyle) *NS_DESIGNATED_INITIALIZER*

Initializes a product table view controller.

Parameters

<i>style</i>	The UITableViewStyle for the table
--------------	------------------------------------

Returns

An initialized [PFProductTableViewController](#) object or `nil` if the object couldn't be created.

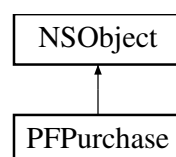
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFProductTableViewController.h`

6.47 PFPurchase Class Reference

```
#import <PFPurchase.h>
```

Inheritance diagram for PFPurchase:



Class Methods

- (void) + [addObserverForProduct:block:](#)
- (void) + [buyProduct:block:](#)
- (void) + [downloadAssetForTransaction:completion:](#)
- (void) + [downloadAssetForTransaction:completion:progress:](#)
- (void) + [restore](#)
- (NSString *) + [assetContentPathForProduct:](#)

6.47.1 Detailed Description

[PFPurchase](#) provides a set of APIs for working with in-app purchases.

This class is currently for iOS only.

Definition at line 19 of file PFPurchase.h.

6.47.2 Method Documentation

6.47.2.1 + (void) addObserverForProduct: (NSString *) *productIdentifier* block:(void (^)(SKPaymentTransaction *transaction)) *block*

Add application logic block which is run when buying a product.

This method should be called once for each product, and should be called before calling <buyProduct:block:>. All invocations to <addObserverForProduct:block:> should happen within the same method, and on the main thread. It is recommended to place all invocations of this method in `application:didFinishLaunchingWithOptions:`.

Parameters

<i>productIdentifier</i>	the product identifier
<i>block</i>	The block to be run when buying a product.

6.47.2.2 + (NSString *) assetContentPathForProduct: (PFProduct *) *product*

6.47.2.3 + (void) buyProduct: (NSString *) *productIdentifier* block:(void (^)(NSError *error)) *block*

Asynchronously initiates the purchase for the product.

Parameters

<i>productIdentifier</i>	the product identifier
<i>block</i>	the completion block.

6.47.2.4 + (void) downloadAssetForTransaction: (SKPaymentTransaction *) *transaction* completion:(void (^)(NSString *filePath, NSError *error)) *completion*

Asynchronously download the purchased asset, which is stored on [Parse](#)'s server.

[Parse](#) verifies the receipt with Apple and delivers the content only if the receipt is valid.

Parameters

<i>transaction</i>	the transaction, which contains the receipt.
<i>completion</i>	the completion block.

6.47.2.5 + (void) downloadAssetForTransaction: (SKPaymentTransaction *) *transaction* completion:(void (^)(NSString *filePath, NSError *error)) *completion* progress:(PFProgressBlock) *progress*

Asynchronously download the purchased asset, which is stored on [Parse](#)'s server.

[Parse](#) verifies the receipt with Apple and delivers the content only if the receipt is valid.

Parameters

<i>transaction</i>	the transaction, which contains the receipt.
<i>completion</i>	the completion block.
<i>progress</i>	the progress block, which is called multiple times to reveal progress of the download.

6.47.2.6 + (void) restore

Asynchronously restore completed transactions for the current user.

Only nonconsumable purchases are restored. If observers for the products have been added before calling this method, invoking the method reruns the application logic associated with the purchase.

Warning

This method is only important to developers who want to preserve purchase states across different installations of the same app.

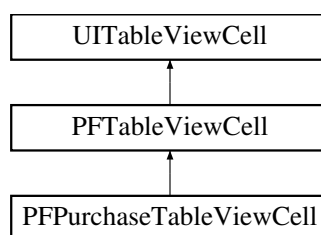
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFPurchase.h](#)

6.48 PFPurchaseTableViewCell Class Reference

```
#import <PFPurchaseTableViewCell.h>
```

Inheritance diagram for PFPurchaseTableViewCell:



Properties

- PFPurchaseTableViewCellState [state](#)
- UILabel * [priceLabel](#)
- UIProgressView * [progressView](#)

6.48.1 Detailed Description

[PFPurchaseTableViewCell](#) is a subclass [<PFTableViewCell>](#) that is used to show products in a [<PFProductTableViewController>](#).

See also

[PFProductTableViewController](#)

Definition at line 48 of file PFPurchaseTableViewCell.h.

6.48.2 Property Documentation

6.48.2.1 - (UILabel*) `priceLabel` [read], [nonatomic], [strong]

Label where price of the product is displayed.

Definition at line 59 of file PFPurchaseTableViewCell.h.

6.48.2.2 - (UIProgressView*) `progressView` [read], [nonatomic], [strong]

Progress view that is shown, when the product is downloading.

Definition at line 64 of file PFPurchaseTableViewCell.h.

6.48.2.3 - (PFPurchaseTableViewCellState) `state` [read], [write], [nonatomic], [assign]

State of the cell.

See also

[PFPurchaseTableViewCellState](#)

Definition at line 54 of file PFPurchaseTableViewCell.h.

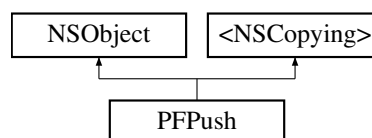
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/[PFPurchaseTableViewCell.h](#)

6.49 PFPush Class Reference

```
#import <PFPush.h>
```

Inheritance diagram for PFPush:



Instance Methods

Configuring a Push Notification

- (void) - [setChannel:](#)
- (void) - [setChannels:](#)
- (void) - [setQuery:](#)
- (void) - [setMessage:](#)
- (void) - [setData:](#)

- (void) - [setPushToAndroid:](#)
- (void) - [setPushToIOS:](#)
- (void) - [expireAtDate:](#)
- (void) - [expireAfterTimeInterval:](#)
- (void) - [clearExpiration](#)

Class Methods

Creating a Push Notification

- ([PFPush](#) *) + [push](#)

Handling Notifications

- (void) + [handlePush:](#)

Managing Channel Subscriptions

- (void) + [storeDeviceToken:](#)
- (NSSet *) + [getSubscribedChannels:](#)
- ([BFTask](#) *) + [getSubscribedChannelsInBackground](#)
- (void) + [getSubscribedChannelsInBackgroundWithBlock:](#)
- (void) + [getSubscribedChannelsInBackgroundWithTarget:selector:](#)
- (BOOL) + [subscribeToChannel:error:](#)
- ([BFTask](#) *) + [subscribeToChannelInBackground:](#)
- (void) + [subscribeToChannelInBackground:block:](#)
- (void) + [subscribeToChannelInBackground:target:selector:](#)
- (BOOL) + [unsubscribeFromChannel:error:](#)
- ([BFTask](#) *) + [unsubscribeFromChannelInBackground:](#)
- (void) + [unsubscribeFromChannelInBackground:block:](#)
- (void) + [unsubscribeFromChannelInBackground:target:selector:](#)

Sending Push Notifications

- (BOOL) + [sendPushMessageToChannel:withMessage:error:](#)
- ([BFTask](#) *) + [sendPushMessageToChannelInBackground:withMessage:](#)
- (void) + [sendPushMessageToChannelInBackground:withMessage:block:](#)
- (void) + [sendPushMessageToChannelInBackground:withMessage:target:selector:](#)
- (BOOL) + [sendPushMessageToQuery:withMessage:error:](#)
- ([BFTask](#) *) + [sendPushMessageToQueryInBackground:withMessage:](#)
- (void) + [sendPushMessageToQueryInBackground:withMessage:block:](#)
- (BOOL) + [sendPushDataToChannel:withData:error:](#)
- ([BFTask](#) *) + [sendPushDataToChannelInBackground:withData:](#)
- (void) + [sendPushDataToChannelInBackground:withData:block:](#)
- (void) + [sendPushDataToChannelInBackground:withData:target:selector:](#)
- (BOOL) + [sendPushDataToQuery:withData:error:](#)
- ([BFTask](#) *) + [sendPushDataToQueryInBackground:withData:](#)
- (void) + [sendPushDataToQueryInBackground:withData:block:](#)
- (BOOL) - [sendPush:](#)
- ([BFTask](#) *) - [sendPushInBackground](#)
- (void) - [sendPushInBackgroundWithBlock:](#)
- (void) - [sendPushInBackgroundWithTarget:selector:](#)

6.49.1 Detailed Description

The `PFPush` class defines a push notification that can be sent from a client device.

The preferred way of modifying or retrieving channel subscriptions is to use the `<PFInstallation>` class, instead of the class methods in `PFPush`.

This class is currently for iOS only. `Parse` does not handle Push Notifications to `Parse` applications running on OS X. Push Notifications can be sent from OS X applications via Cloud Code or the REST API to push-enabled devices (e.g. iOS or Android).

Definition at line 24 of file `PFPush.h`.

6.49.2 Method Documentation

6.49.2.1 - (void) clearExpiration

Clears both expiration values, indicating that the notification should never expire.

6.49.2.2 - (void) expireAfterTimeInterval: (NSTimeInterval) *timeInterval*

Sets the time interval after which this notification should expire.

This notification will be sent to devices which are either online at the time the notification is sent, or which come online within the given time interval of the notification being received by `Parse`'s server. An interval which is less than or equal to zero indicates that the message should only be sent to devices which are currently online.

Parameters

<i>timeInterval</i>	The interval after which the notification should expire.
---------------------	--

6.49.2.3 - (void) expireAtDate: (NSDate *) *date*

Sets the expiration time for this notification.

The notification will be sent to devices which are either online at the time the notification is sent, or which come online before the expiration time is reached. Because device clocks are not guaranteed to be accurate, most applications should instead use `<expireAfterTimeInterval:>`.

See also

- [expireAfterTimeInterval:](#)

Parameters

<i>date</i>	The time at which the notification should expire.
-------------	---

6.49.2.4 + (NSSet *) getSubscribedChannels: (NSError **) *error*

Synchronously get all the channels that this device is subscribed to.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns an `NSSet` containing all the channel names this device is subscribed to.

6.49.2.5 + (BFTask *) getSubscribedChannelsInBackground

Asynchronously get all the channels that this device is subscribed to.

Returns

The task, that encapsulates the work being done.

6.49.2.6 + (void) getSubscribedChannelsInBackgroundWithBlock: (PFSetResultBlock) block

Asynchronously get all the channels that this device is subscribed to.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(NSSet *channels, NSError *error).</code>
--------------	---

6.49.2.7 + (void) getSubscribedChannelsInBackgroundWithTarget: (id) target selector:(SEL) selector**6.49.2.8 + (void) handlePush: (NSDictionary *) userInfo**

A default handler for push notifications while the app is active that could be used to mimic the behavior of iOS push notifications while the app is backgrounded or not running.

Call this from `application:didReceiveRemoteNotification:`.

Parameters

<i>userInfo</i>	The <code>userInfo</code> dictionary you get in <code>application:didReceiveRemoteNotification:</code> .
-----------------	--

6.49.2.9 + (PFPush *) push**6.49.2.10 - (BOOL) sendPush: (NSError **) error**

Synchronously send this push message.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns whether the send succeeded.

6.49.2.11 + (BOOL) sendPushDataToChannel: (NSString *) channel withData:(NSDictionary *) data error:(NSError **) error

Synchronously send a push message with arbitrary data to a channel.

See the guide for information about the dictionary structure.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>data</i>	The data to send.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

Returns whether the send succeeded.

6.49.2.12 + (BFTask *) sendPushDataToChannelInBackground: (NSString *) *channel* withData:(NSDictionary *) *data*

Asynchronously send a push message with arbitrary data to a channel.

See the guide for information about the dictionary structure.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>data</i>	The data to send.

Returns

The task, that encapsulates the work being done.

6.49.2.13 + (void) sendPushDataToChannelInBackground: (NSString *) *channel* withData:(NSDictionary *) *data* block:(PFBooleanResultBlock) *block*

Asynchronously sends a push message with arbitrary data to a channel and calls the given block.

See the guide for information about the dictionary structure.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>data</i>	The data to send.
<i>block</i>	The block to execute. It should have the following argument signature: $\wedge (BOOL \leftarrow L \text{ succeeded}, NSError *error)$.

6.49.2.14 + (void) sendPushDataToChannelInBackground: (NSString *) *channel* withData:(NSDictionary *) *data* target:(id) *target* selector:(SEL) *selector*

6.49.2.15 + (BOOL) sendPushDataToQuery: (PFQuery *) *query* withData:(NSDictionary *) *data* error:(NSError **) *error*

Synchronously send a push message with arbitrary data to a query.

See the guide for information about the dictionary structure.

Parameters

<i>query</i>	The query to send to. The query must be a <PFInstallation> query created with <[PFInstallation query]>.
<i>data</i>	The data to send.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the send succeeded.

6.49.2.16 + (BFTask *) sendPushDataToQueryInBackground: (PFQuery *) *query* withData:(NSDictionary *) *data*

Asynchronously send a push message with arbitrary data to a query.

See the guide for information about the dictionary structure.

Parameters

<i>query</i>	The query to send to. The query must be a <PFInstallation> query created with <[PFInstallation query]>.
<i>data</i>	The data to send.

Returns

The task, that encapsulates the work being done.

6.49.2.17 + (void) sendPushDataToQueryInBackground: (PFQuery *) query withData:(NSDictionary *) data block:(PFBooleanResultBlock) block

Asynchronously sends a push message with arbitrary data to a query and calls the given block.

See the guide for information about the dictionary structure.

Parameters

<i>query</i>	The query to send to. The query must be a <PFInstallation> query created with <[PFInstallation query]>.
<i>data</i>	The data to send.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL succeeded, NSError *error).

6.49.2.18 - (BFTask *) sendPushInBackground

Asynchronously send this push message.

Returns

The task, that encapsulates the work being done.

6.49.2.19 - (void) sendPushInBackgroundWithBlock: (PFBooleanResultBlock) block

Asynchronously send this push message and executes the given callback block.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL succeeded, NSError *error).
--------------	---

6.49.2.20 - (void) sendPushInBackgroundWithTarget: (id) target selector:(SEL) selector

6.49.2.21 + (BOOL) sendPushMessageToChannel: (NSString *) channel withMessage:(NSString *) message error:(NSError **) error

Synchronously send a push message to a channel.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>message</i>	The message to send.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the send succeeded.

6.49.2.22 + (BFTask *) sendPushMessageToChannelInBackground: (NSString *) *channel* withMessage:(NSString *) *message*

Asynchronously send a push message to a channel.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>message</i>	The message to send.

Returns

The task, that encapsulates the work being done.

6.49.2.23 + (void) sendPushMessageToChannelInBackground: (NSString *) *channel* withMessage:(NSString *) *message* block:(PFBooleanResultBlock) *block*

Asynchronously sends a push message to a channel and calls the given block.

Parameters

<i>channel</i>	The channel to send to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>message</i>	The message to send.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftarrow L succeeded, NSError *error)

6.49.2.24 + (void) sendPushMessageToChannelInBackground: (NSString *) *channel* withMessage:(NSString *) *message* target:(id) *target* selector:(SEL) *selector*

6.49.2.25 + (BOOL) sendPushMessageToQuery: (PFQuery *) *query* withMessage:(NSString *) *message* error:(NSError **) *error*

Send a push message to a query.

Parameters

<i>query</i>	The query to send to. The query must be a <PFIInstallation> query created with <[PFInstallation query]>.
<i>message</i>	The message to send.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the send succeeded.

6.49.2.26 + (BFTask *) sendPushMessageToQueryInBackground: (PFQuery *) *query* withMessage:(NSString *) *message*

Asynchronously send a push message to a query.

Parameters

<i>query</i>	The query to send to. The query must be a <PFInstallation> query created with <[PFInstallation query]>.
<i>message</i>	The message to send.

Returns

The task, that encapsulates the work being done.

6.49.2.27 + (void) sendPushMessageToQueryInBackground: (PFQuery *) *query* withMessage:(NSString *) *message* block:(PFBooleanResultBlock) *block*

Asynchronously sends a push message to a query and calls the given block.

Parameters

<i>query</i>	The query to send to. The query must be a PFInstallation query created with [PFInstallation query] .
<i>message</i>	The message to send.
<i>block</i>	The block to execute. It should have the following argument signature: $\wedge (BOOL \leftarrow L \text{ succeeded}, NSError *error)$

6.49.2.28 - (void) setChannel: (NSString *) *channel*

Sets the channel on which this push notification will be sent.

Parameters

<i>channel</i>	The channel to set for this push. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
----------------	---

6.49.2.29 - (void) setChannels: (NSArray *) *channels*

Sets the array of channels on which this push notification will be sent.

Parameters

<i>channels</i>	The array of channels to set for this push. Each channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
-----------------	--

6.49.2.30 - (void) setData: (NSDictionary *) *data*

Sets an arbitrary data payload for this push notification.

See the guide for information about the dictionary structure.

Warning

This will overwrite any data specified in setMessage.

Parameters

<i>data</i>	The data to send in this push.
-------------	--------------------------------

6.49.2.31 - (void) setMessage: (NSString *) message

Sets an alert message for this push notification.

Warning

This will overwrite any data specified in setData.

Parameters

<i>message</i>	The message to send in this push.
----------------	-----------------------------------

6.49.2.32 - (void) setPushToAndroid: ("Please use a with a constraint on deviceType.") PARSE_DEPRECATED[PFInstallation query]

Sets whether this push will go to Android devices.

Parameters

<i>pushToAndroid</i>	Whether this push will go to Android devices.
----------------------	---

Deprecated Please use a [PFInstallation query] with a constraint on deviceType instead.

6.49.2.33 - (void) setPushToiOS: ("Please use a with a constraint on deviceType.") PARSE_DEPRECATED[PFInstallation query]

Sets whether this push will go to iOS devices.

Parameters

<i>pushToiOS</i>	Whether this push will go to iOS devices.
------------------	---

Deprecated Please use a [PFInstallation query] with a constraint on deviceType instead.

6.49.2.34 - (void) setQuery: (PFQuery *) query

Sets an installation query to which this push notification will be sent.

The query should be created via <[PFInstallation query]> and should not specify a skip, limit, or order.

Parameters

<i>query</i>	The installation query to set for this push.
--------------	--

6.49.2.35 + (void) storeDeviceToken: (id) deviceToken

Store the device token locally for push notifications.

Usually called from your main app delegate's didRegisterForRemoteNotificationsWithDeviceToken:.

Parameters

<i>deviceToken</i>	Either as an NSData straight from <code>application:didRegisterForRemoteNotificationsWithDeviceToken:</code> or as an NSString if you converted it yourself.
--------------------	--

6.49.2.36 + (BOOL) `subscribeToChannel:` (NSString *) *channel* error:(NSError **) *error*

Synchronously subscribes the device to a channel of push notifications.

Parameters

<i>channel</i>	The channel to subscribe to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the subscribe succeeded.

6.49.2.37 + (BFTask *) `subscribeToChannelInBackground:` (NSString *) *channel*

Asynchronously subscribes the device to a channel of push notifications.

Parameters

<i>channel</i>	The channel to subscribe to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
----------------	--

Returns

The task, that encapsulates the work being done.

6.49.2.38 + (void) `subscribeToChannelInBackground:` (NSString *) *channel* block:(PFBooleanResultBlock) *block*

Asynchronously subscribes the device to a channel of push notifications and calls the given block.

Parameters

<i>channel</i>	The channel to subscribe to. The channel name must start with a letter and contain only letters, numbers, dashes, and underscores.
<i>block</i>	The block to execute. It should have the following argument signature: <code>^(BOOL succeeded, NSError *error)</code>

6.49.2.39 + (void) `subscribeToChannelInBackground:` (NSString *) *channel* target:(id) *target* selector:(SEL) *selector*6.49.2.40 + (BOOL) `unsubscribeFromChannel:` (NSString *) *channel* error:(NSError **) *error*

Synchronously unsubscribes the device to a channel of push notifications.

Parameters

<i>channel</i>	The channel to unsubscribe from.
<i>error</i>	Pointer to an NSError that will be set if necessary.

Returns

Returns whether the unsubscribe succeeded.

6.49.2.41 + (BFTask *) unsubscribeFromChannelInBackground: (NSString *) *channel*

Asynchronously unsubscribes the device from a channel of push notifications.

Parameters

<i>channel</i>	The channel to unsubscribe from.
----------------	----------------------------------

Returns

The task, that encapsulates the work being done.

6.49.2.42 + (void) unsubscribeFromChannelInBackground: (NSString *) *channel* block:(PFBooleanResultBlock) *block*

Asynchronously unsubscribes the device from a channel of push notifications and calls the given block.

Parameters

<i>channel</i>	The channel to unsubscribe from.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL succeeded, NSError *error).

6.49.2.43 + (void) unsubscribeFromChannelInBackground: (NSString *) *channel* target:(id) *target* selector:(SEL) *selector*

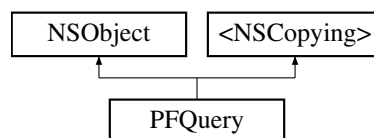
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFPush.h

6.50 PFQuery Class Reference

```
#import <PFQuery.h>
```

Inheritance diagram for PFQuery:



Instance Methods

Adding Basic Constraints

- (instancetype) - [includeKey:](#)
- (instancetype) - [selectKeys:](#)
- (instancetype) - [whereKeyExists:](#)
- (instancetype) - [whereKeyDoesNotExist:](#)
- (instancetype) - [whereKey:equalTo:](#)
- (instancetype) - [whereKey:lessThan:](#)
- (instancetype) - [whereKey:lessThanOrEqualTo:](#)
- (instancetype) - [whereKey:greaterThan:](#)

- (instancetype) - [whereKey:greaterThanOrEqualTo:](#)
- (instancetype) - [whereKey:notEqualTo:](#)
- (instancetype) - [whereKey:containedIn:](#)
- (instancetype) - [whereKey:notContainedIn:](#)
- (instancetype) - [whereKey:containsAllObjectsInArray:](#)

Adding Location Constraints

- (instancetype) - [whereKey:nearGeoPoint:](#)
- (instancetype) - [whereKey:nearGeoPoint:withinMiles:](#)
- (instancetype) - [whereKey:nearGeoPoint:withinKilometers:](#)
- (instancetype) - [whereKey:nearGeoPoint:withinRadians:](#)
- (instancetype) - [whereKey:withinGeoBoxFromSouthwest:toNortheast:](#)

Adding String Constraints

- (instancetype) - [whereKey:matchesRegex:](#)
- (instancetype) - [whereKey:matchesRegex:modifiers:](#)
- (instancetype) - [whereKey:containsString:](#)
- (instancetype) - [whereKey:hasPrefix:](#)
- (instancetype) - [whereKey:hasSuffix:](#)

Sorting

- (instancetype) - [orderByAscending:](#)
- (instancetype) - [addAscendingOrder:](#)
- (instancetype) - [orderByDescending:](#)
- (instancetype) - [addDescendingOrder:](#)
- (instancetype) - [orderBySortDescriptor:](#)
- (instancetype) - [orderBySortDescriptors:](#)

Getting all Matches for a Query

- (PF_NULLABLE NSArray *) - [findObjects](#)
- (PF_NULLABLE NSArray *) - [findObjects:](#)
- (BFTask *) - [findObjectsInBackground](#)
- (void) - [findObjectsInBackgroundWithBlock:](#)
- (void) - [findObjectsInBackgroundWithTarget:selector:](#)

Getting the First Match in a Query

- (PF_NULLABLE PFObjct *) - [getFirstObject](#)
- (PF_NULLABLE PFObjct *) - [getFirstObject:](#)
- (BFTask *) - [getFirstObjectInBackground](#)
- (void) - [getFirstObjectInBackgroundWithBlock:](#)
- (void) - [getFirstObjectInBackgroundWithTarget:selector:](#)

Counting the Matches in a Query

- (NSInteger) - [countObjects](#)
- (NSInteger) - [countObjects:](#)
- (BFTask *) - [countObjectsInBackground](#)
- (void) - [countObjectsInBackgroundWithBlock:](#)
- (void) - [countObjectsInBackgroundWithTarget:selector:](#)

Cancelling a Query

- (void) - [cancel](#)

Query Source

- (instancetype) - [fromLocalDatastore](#)
- (instancetype) - [fromPin](#)
- (instancetype) - [fromPinWithName:](#)
- (instancetype) - [ignoreACLs](#)

Class Methods

Getting User Objects

- (PF_NULLABLE PFUser *) + [getUserObjectWithId:](#)
- (PF_NULLABLE PFUser *) + [getUserObjectWithId:error:](#)
- ("Use [PFUser query] instead.") + [PARSE_DEPRECATED](#)

Properties

Paginating Results

- NSInteger [limit](#)
- NSInteger [skip](#)

Advanced Settings

- BOOL [trace](#)

Creating a Query for a Class

- NSString * [parseClassName](#)
- (instancetype) + [queryWithClassName:](#)
- (instancetype) + [queryWithClassName:predicate:](#)
- (instancetype) - [initWithClassName:](#)

Adding Subqueries

- (instancetype) + [orQueryWithSubqueries:](#)
- (instancetype) - [whereKey:matchesKey:inQuery:](#)
- (instancetype) - [whereKey:doesNotMatchKey:inQuery:](#)
- (instancetype) - [whereKey:matchesQuery:](#)
- (instancetype) - [whereKey:doesNotMatchQuery:](#)

Getting Objects by ID

- (PF_NULLABLE PFObject *) + [getObjectOfClass:objectId:](#)
- (PF_NULLABLE PFObject *) + [getObjectOfClass:objectId:error:](#)
- (PF_NULLABLE PFObject *) - [getObjectWithId:](#)
- (PF_NULLABLE PFObject *) - [getObjectWithId:error:](#)
- (BFTask *) - [getObjectInBackgroundWithId:](#)
- (void) - [getObjectInBackgroundWithId:block:](#)
- (void) - [getObjectInBackgroundWithId:target:selector:](#)

Controlling Caching Behavior

- PFCachePolicy [cachePolicy](#)
- NSTimeInterval [maxCacheAge](#)
- (void) + [clearAllCachedResults](#)
- (BOOL) - [hasCachedResult](#)
- (void) - [clearCachedResult](#)

6.50.1 Detailed Description

The `PFQuery` class defines a query that is used to query for `<PFOBJECT>`s.

Definition at line 28 of file `PFQuery.h`.

6.50.2 Method Documentation

6.50.2.1 `-(instancetype) addAscendingOrder: (NSString *) key`

Additionally sort in *ascending* order by the given key.

The previous keys provided will precedence over this key.

Parameters

<i>key</i>	The key to order by.
------------	----------------------

6.50.2.2 `-(instancetype) addDescendingOrder: (NSString *) key`

Additionally sort in *descending* order by the given key.

The previous keys provided will precedence over this key.

Parameters

<i>key</i>	The key to order by.
------------	----------------------

6.50.2.3 `-(void) cancel`

Cancels the current network request (if any). Ensures that callbacks won't be called.

6.50.2.4 `+(void) clearAllCachedResults`

Clears the cached results for all queries.

6.50.2.5 `-(void) clearCachedResult`

Clears the cached result for this query. If there is no cached result, this is a noop.

6.50.2.6 `-(NSInteger) countObjects`

Counts objects *synchronously* based on the constructed query.

Returns

Returns the number of `<PFOBJECT>` objects that match the query, or `-1` if there is an error.

6.50.2.7 `-(NSInteger) countObjects: (NSError **) error`

Counts objects *synchronously* based on the constructed query and sets an error if there was one.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns the number of `<PFObjec>` objects that match the query, or `-1` if there is an error.

6.50.2.8 - (BFTask *) countObjectsInBackground

Counts objects *asynchronously* and sets `NSNumber` with count as a result of the task.

Returns

The task, that encapsulates the work being done.

6.50.2.9 - (void) countObjectsInBackgroundWithBlock: (PF_NULLABLE PFIntegerResultBlock) block

Counts objects *asynchronously* and calls the given block with the counts.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(int count, NSError *error)</code>
--------------	--

6.50.2.10 - (void) countObjectsInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.50.2.11 - (PF_NULLABLE NSArray *) findObjects

Finds objects *synchronously* based on the constructed query.

Returns

Returns an array of `<PFObjec>` objects that were found.

6.50.2.12 - (PF_NULLABLE NSArray *) findObjects: (NSError **) error

Finds objects *synchronously* based on the constructed query and sets an error if there was one.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns an array of `<PFObjec>` objects that were found.

6.50.2.13 - (BFTask *) findObjectsInBackground

Finds objects *asynchronously* and sets the `NSArray` of `<PFObjec>` objects as a result of the task.

Returns

The task, that encapsulates the work being done.

6.50.2.14 - (void) findObjectsInBackgroundWithBlock: (PF_NULLABLE PFArrayResultBlock) *block*

Finds objects *asynchronously* and calls the given block with the results.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(NSArray *objects, NSError *error)</code>
--------------	---

6.50.2.15 - (void) `findObjectsInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector`

6.50.2.16 - (instancetype) `fromLocalDatastore`

Change the source of this query to all pinned objects.

Warning

Requires Local Datastore to be enabled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

See also

[cachePolicy](#)

6.50.2.17 - (instancetype) `fromPin`

Change the source of this query to the default group of pinned objects.

Warning

Requires Local Datastore to be enabled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

See also

[PFOBJECTDefaultPin](#)
[cachePolicy](#)

6.50.2.18 - (instancetype) `fromPinWithName: (PF_NULLABLE NSString *) name`

Change the source of this query to a specific group of pinned objects.

Warning

Requires Local Datastore to be enabled.

Parameters

<i>name</i>	The pinned group.
-------------	-------------------

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

See also

[PObjectDefaultPin
cachePolicy](#)

6.50.2.19 - (PF_NULLABLE PObject *) getFirstObject

Gets an object *synchronously* based on the constructed query.

Warning

This method mutates the query. It will reset the limit to 1.

Returns

Returns a <PObject>, or `nil` if none was found.

6.50.2.20 - (PF_NULLABLE PObject *) getFirstObject: (NSError **) error

Gets an object *synchronously* based on the constructed query and sets an error if any occurred.

Warning

This method mutates the query. It will reset the limit to 1.

Parameters

<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.
--------------	---

Returns

Returns a <PObject>, or `nil` if none was found.

6.50.2.21 - (BFTask *) getFirstObjectInBackground

Gets an object *asynchronously* and sets it as a result of the task.

Warning

This method mutates the query. It will reset the limit to 1.

Returns

The task, that encapsulates the work being done.

6.50.2.22 - (void) `getFirstObjectInBackgroundWithBlock:` (PF_NULLABLE PObjectResultBlock) *block*

Gets an object *asynchronously* and calls the given block with the result.

Warning

This method mutates the query. It will reset the limit to 1.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: <code>^(PObject *object, NSError *error). result will be nil if error is set OR no object was found matching the query. error will be nil if result is set OR if the query succeeded, but found no results.</code>
--------------	---

6.50.2.23 - (void) `getFirstObjectInBackgroundWithTarget:` (PF_NULLABLE_S id) *target selector*:(PF_NULLABLE_S SEL) *selector*

6.50.2.24 - (BFTask *) `getObjectInBackgroundWithId:` (NSString *) *objectId*

Gets a <PObject> asynchronously and calls the given block with the result.

Warning

This method mutates the query. It will reset limit to 1, skip to 0 and remove all conditions, leaving only `objectId`.

Parameters

<i>objectId</i>	The id of the object that is being requested.
-----------------	---

Returns

The task, that encapsulates the work being done.

6.50.2.25 - (void) `getObjectInBackgroundWithId:` (NSString *) *objectId* `block:`(PF_NULLABLE PObjectResultBlock) *block*

Gets a <PObject> asynchronously and calls the given block with the result.

Warning

This method mutates the query. It will reset limit to 1, skip to 0 and remove all conditions, leaving only `objectId`.

Parameters

<i>objectId</i>	The id of the object that is being requested.
<i>block</i>	The block to execute. The block should have the following argument signature: <code>^(NSArray *object, NSError *error)</code>

6.50.2.26 - (void) `getObjectInBackgroundWithId:` (NSString *) *objectId* `target:`(PF_NULLABLE_S id) *target selector:(PF_NULLABLE_S SEL) *selector**

6.50.2.27 + (PF_NULLABLE PObject *) `getObjectOfClass:` (NSString *) *objectClass* `objectId:`(NSString *) *objectId*

Returns a <PObject> with a given class and id.

Parameters

<i>objectClass</i>	The class name for the object that is being requested.
<i>objectId</i>	The id of the object that is being requested.

Returns

The <PFObj> if found. Returns `nil` if the object isn't found, or if there was an error.

6.50.2.28 + (PF_NULLABLE PFObj *) getObjectOfClass: (NSString *) *objectClass* objectId:(NSString *) *objectId* error:(NSError **) *error*

Returns a <PFObj> with a given class and id and sets an error if necessary.

Parameters

<i>objectClass</i>	The class name for the object that is being requested.
<i>objectId</i>	The id of the object that is being requested.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

The <PFObj> if found. Returns `nil` if the object isn't found, or if there was an error.

6.50.2.29 - (PF_NULLABLE PFObj *) getObjectWithId: (NSString *) *objectId*

Returns a <PFObj> with the given id.

Warning

This method mutates the query. It will reset limit to 1, skip to 0 and remove all conditions, leaving only `objectId`.

Parameters

<i>objectId</i>	The id of the object that is being requested.
-----------------	---

Returns

The <PFObj> if found. Returns `nil` if the object isn't found, or if there was an error.

6.50.2.30 - (PF_NULLABLE PFObj *) getObjectWithId: (NSString *) *objectId* error:(NSError **) *error*

Returns a <PFObj> with the given id and sets an error if necessary.

Warning

This method mutates the query. It will reset limit to 1, skip to 0 and remove all conditions, leaving only `objectId`.

Parameters

<i>objectId</i>	The id of the object that is being requested.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

The `<PFObj>` if found. Returns nil if the object isn't found, or if there was an error.

6.50.2.31 + (PF_NULLABLE PFUser *) getUserObjectWithId: (NSString *) objectId

Returns a `<PFUser>` with a given id.

Parameters

<i>objectId</i>	The id of the object that is being requested.
-----------------	---

Returns

The `PFUser` if found. Returns nil if the object isn't found, or if there was an error.

6.50.2.32 + (PF_NULLABLE PFUser *) getUserObjectWithId: (NSString *) objectId error:(NSError **) error

Returns a `PFUser` with a given class and id and sets an error if necessary.

Parameters

<i>objectId</i>	The id of the object that is being requested.
<i>error</i>	Pointer to an <code>NSError</code> that will be set if necessary.

Returns

The `PFUser` if found. Returns nil if the object isn't found, or if there was an error.

6.50.2.33 - (BOOL) hasCachedResult

Returns whether there is a cached result for this query.

Returns

YES if there is a cached result for this query, otherwise NO.

6.50.2.34 - (instancetype) ignoreACLs

Ignore ACLs when querying from the Local Datastore.

This is particularly useful when querying for objects with Role based ACLs set on them.

Warning

Requires Local Datastore to be enabled.

Returns

The same instance of `PFQuery` as the receiver. This allows method chaining.

6.50.2.35 - (instancetype) includeKey: (NSString *) key

Make the query include PFObjets that have a reference stored at the provided key.

This has an effect similar to a join. You can use dot notation to specify which fields in the included object are also fetch.

Parameters

<i>key</i>	The key to load child <PFObjets>s for.
------------	--

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.36 - (instancetype) initWithClassName: (NSString *) newClassName

Initializes the query with a class name.

Parameters

<i>newClassName</i>	The class name.
---------------------	-----------------

6.50.2.37 - (instancetype) orderByAscending: (NSString *) key

Sort the results in *ascending* order with the given key.

Parameters

<i>key</i>	The key to order by.
------------	----------------------

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.38 - (instancetype) orderByDescending: (NSString *) key

Sort the results in *descending* order with the given key.

Parameters

<i>key</i>	The key to order by.
------------	----------------------

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.39 - (instancetype) orderBySortDescriptor: (NSSortDescriptor *) sortDescriptor

Sort the results using a given sort descriptor.

Parameters

<i>sortDescriptor</i>	The NSSortDescriptor to use to sort the results of the query.
-----------------------	---

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.40 - (instancetype) orderBySortDescriptors: (PF_NULLABLE NSArray *) *sortDescriptors*

Sort the results using a given array of sort descriptors.

Parameters

<i>sortDescriptors</i>	An array of <code>NSSortDescriptor</code> objects to use to sort the results of the query.
------------------------	--

Returns

The same instance of `PFQuery` as the receiver. This allows method chaining.

6.50.2.41 + (instancetype) orQueryWithSubqueries: (NSArray *) queries

Returns a `PFQuery` that is the `or` of the passed in queries.

Parameters

<i>queries</i>	The list of queries to or together.
----------------	-------------------------------------

Returns

An instance of `PFQuery` that is the `or` of the passed in queries.

6.50.2.42 + ("Use [PFUser query] instead.") PARSE_DEPRECATED

Deprecated Please use `[PFUser query]` instead.

6.50.2.43 + (instancetype) queryWithClassName: (NSString *) className

Returns a `PFQuery` for a given class.

Parameters

<i>className</i>	The class to query on.
------------------	------------------------

Returns

A `PFQuery` object.

6.50.2.44 + (instancetype) queryWithClassName: (NSString *) className predicate:(PF_NULLABLE NSPredicate *) predicate

Creates a `PFQuery` with the constraints given by predicate.

The following types of predicates are supported:

- Simple comparisons such as `=`, `!=`, `<`, `>`, `<=`, `>=`, and `BETWEEN` with a key and a constant.
- Containment predicates, such as `x IN {1, 2, 3}`.
- Key-existence predicates, such as `x IN SELF`.
- `BEGINSWITH` expressions.
- Compound predicates with `AND`, `OR`, and `NOT`.
- SubQueries with `key IN %@, subquery`.

The following types of predicates are NOT supported:

- Aggregate operations, such as `ANY`, `SOME`, `ALL`, or `NONE`.
- Regular expressions, such as `LIKE`, `MATCHES`, `CONTAINS`, or `ENDSWITH`.
- Predicates comparing one key to another.
- Complex predicates with many `ORed` clauses.

Parameters

<i>className</i>	The class to query on.
<i>predicate</i>	The predicate to create conditions from.

6.50.2.45 - (instancetype) selectKeys: (NSArray *) keys

Make the query restrict the fields of the returned `<PFObj>s` to include only the provided keys.

If this is called multiple times, then all of the keys specified in each of the calls will be included.

Parameters

<i>keys</i>	The keys to include in the result.
-------------	------------------------------------

Returns

The same instance of `PFQuery` as the receiver. This allows method chaining.

6.50.2.46 - (instancetype) whereKey: (NSString *) key containedIn:(NSArray *) array

Add a constraint to the query that requires a particular key's object to be contained in the provided array.

Parameters

<i>key</i>	The key to be constrained.
<i>array</i>	The possible values for the key's object.

Returns

The same instance of `PFQuery` as the receiver. This allows method chaining.

6.50.2.47 - (instancetype) whereKey: (NSString *) key containsAllObjectsInArray:(NSArray *) array

Add a constraint to the query that requires a particular key's array contains every element of the provided array.

Parameters

<i>key</i>	The key to be constrained.
<i>array</i>	The array of values to search for.

Returns

The same instance of `PFQuery` as the receiver. This allows method chaining.

6.50.2.48 - (instancetype) whereKey: (NSString *) key containsString:(PF_NULLABLE NSString *) substring

Add a constraint for finding string values that contain a provided substring.

Warning

This will be slow for large datasets.

Parameters

<i>key</i>	The key that the string to match is stored in.
<i>substring</i>	The substring that the value must contain.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.49 - (instancetype) whereKey: (NSString *) key doesNotMatchKey:(NSString *) otherKey inQuery:(PFQuery *) query

Adds a constraint that requires that a key's value **NOT** match a value in another key in objects returned by a sub query.

Parameters

<i>key</i>	The key that the value is stored.
<i>otherKey</i>	The key in objects in the returned by the sub query whose value should match.
<i>query</i>	The query to run.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.50 - (instancetype) whereKey: (NSString *) key doesNotMatchQuery:(PFQuery *) query

Add a constraint that requires that a key's value to not match a [PFQuery](#) constraint.

Warning

This only works where the key's values are <PFOBJECT>s or arrays of <PFOBJECT>s.

Parameters

<i>key</i>	The key that the value is stored in
<i>query</i>	The query the value should not match

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.51 - (instancetype) whereKey: (NSString *) key equalTo:(id) object

Add a constraint to the query that requires a particular key's object to be equal to the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that must be equalled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.52 - (instancetype) whereKey: (NSString *) key greaterThan:(id) object

Add a constraint to the query that requires a particular key's object to be greater than the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that must be equalled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.53 - (instancetype) whereKey: (NSString *) key greaterThanOrEqualTo:(id) object

Add a constraint to the query that requires a particular key's object to be greater than or equal to the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that must be equalled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.54 - (instancetype) whereKey: (NSString *) key hasPrefix:(PF_NULLABLE NSString *) prefix

Add a constraint for finding string values that start with a provided prefix.

This will use smart indexing, so it will be fast for large datasets.

Parameters

<i>key</i>	The key that the string to match is stored in.
<i>prefix</i>	The substring that the value must start with.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.55 - (instancetype) whereKey: (NSString *) key hasSuffix:(PF_NULLABLE NSString *) suffix

Add a constraint for finding string values that end with a provided suffix.

Warning

This will be slow for large datasets.

Parameters

<i>key</i>	The key that the string to match is stored in.
<i>suffix</i>	The substring that the value must end with.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.56 - (instancetype) whereKey: (NSString *) key lessThan:(id) object

Add a constraint to the query that requires a particular key's object to be less than the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that provides an upper bound.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.57 - (instancetype) whereKey: (NSString *) key lessThanOrEqualTo:(id) object

Add a constraint to the query that requires a particular key's object to be less than or equal to the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that must be equalled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.58 - (instancetype) whereKey: (NSString *) key matchesKey:(NSString *) otherKey inQuery:(PFQuery *) query

Adds a constraint that requires that a key's value matches a value in another key in objects returned by a sub query.

Parameters

<i>key</i>	The key that the value is stored.
<i>otherKey</i>	The key in objects in the returned by the sub query whose value should match.
<i>query</i>	The query to run.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.59 - (instancetype) whereKey: (NSString *) key matchesQuery:(PFQuery *) query

Add a constraint that requires that a key's value matches a [PFQuery](#) constraint.

Warning

This only works where the key's values are `<PFOBJECT>`s or arrays of `<PFOBJECT>`s.

Parameters

<i>key</i>	The key that the value is stored in
<i>query</i>	The query the value should match

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.60 - (instancetype) whereKey: (NSString *) key matchesRegex:(NSString *) regex

Add a regular expression constraint for finding string values that match the provided regular expression.

Warning

This may be slow for large datasets.

Parameters

<i>key</i>	The key that the string to match is stored in.
<i>regex</i>	The regular expression pattern to match.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.61 - (instancetype) whereKey: (NSString *) key matchesRegex:(NSString *) regex modifiers:(PF_NULLABLE NSString *) modifiers

Add a regular expression constraint for finding string values that match the provided regular expression.

Warning

This may be slow for large datasets.

Parameters

<i>key</i>	The key that the string to match is stored in.
<i>regex</i>	The regular expression pattern to match.
<i>modifiers</i>	Any of the following supported PCRE modifiers: <ul style="list-style-type: none"> • <code>i</code> - Case insensitive search • <code>m</code> - Search across multiple lines of input

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.62 - (instancetype) whereKey: (NSString *) key nearGeoPoint:(PFGeoPoint *) geopoint

Add a constraint to the query that requires a particular key's coordinates (specified via `<PFGeoPoint>`) be near a reference point.

Distance is calculated based on angular distance on a sphere. Results will be sorted by distance from reference point.

Parameters

<i>key</i>	The key to be constrained.
<i>geopoint</i>	The reference point represented as a <PGeoPoint>.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.63 - (instancetype) whereKey: (NSString *) *key* nearGeoPoint:(PGeoPoint *) *geopoint* withinKilometers:(double) *maxDistance*

Add a constraint to the query that requires a particular key's coordinates (specified via <PGeoPoint>) be near a reference point and within the maximum distance specified (in kilometers).

Distance is calculated based on a spherical coordinate system. Results will be sorted by distance (nearest to farthest) from the reference point.

Parameters

<i>key</i>	The key to be constrained.
<i>geopoint</i>	The reference point represented as a <PGeoPoint>.
<i>maxDistance</i>	Maximum distance in kilometers.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.64 - (instancetype) whereKey: (NSString *) *key* nearGeoPoint:(PGeoPoint *) *geopoint* withinMiles:(double) *maxDistance*

Add a constraint to the query that requires a particular key's coordinates (specified via <PGeoPoint>) be near a reference point and within the maximum distance specified (in miles).

Distance is calculated based on a spherical coordinate system. Results will be sorted by distance (nearest to farthest) from the reference point.

Parameters

<i>key</i>	The key to be constrained.
<i>geopoint</i>	The reference point represented as a <PGeoPoint>.
<i>maxDistance</i>	Maximum distance in miles.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.65 - (instancetype) whereKey: (NSString *) *key* nearGeoPoint:(PGeoPoint *) *geopoint* withinRadians:(double) *maxDistance*

Add a constraint to the query that requires a particular key's coordinates (specified via <PGeoPoint>) be near a reference point and within the maximum distance specified (in radians). Distance is calculated based on angular distance on a sphere. Results will be sorted by distance (nearest to farthest) from the reference point.

Parameters

<i>key</i>	The key to be constrained.
<i>geopoint</i>	The reference point as a <PFGeoPoint>.
<i>maxDistance</i>	Maximum distance in radians.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.66 - (instancetype) whereKey: (NSString *) key notContainedIn:(NSArray *) array

Add a constraint to the query that requires a particular key's object not be contained in the provided array.

Parameters

<i>key</i>	The key to be constrained.
<i>array</i>	The list of values the key's object should not be.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.67 - (instancetype) whereKey: (NSString *) key notEqualTo:(id) object

Add a constraint to the query that requires a particular key's object to be not equal to the provided object.

Parameters

<i>key</i>	The key to be constrained.
<i>object</i>	The object that must not be equalled.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.68 - (instancetype) whereKey: (NSString *) key withinGeoBoxFromSouthwest:(PFGeoPoint *) southwest toNortheast:(PFGeoPoint *) northeast

Add a constraint to the query that requires a particular key's coordinates (specified via <PFGeoPoint>) be contained within a given rectangular geographic bounding box.

Parameters

<i>key</i>	The key to be constrained.
<i>southwest</i>	The lower-left inclusive corner of the box.
<i>northeast</i>	The upper-right inclusive corner of the box.

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.69 - (instancetype) whereKeyDoesNotExist: (NSString *) key

Add a constraint that requires a key not exist.

Parameters

<i>key</i>	The key that should not exist.
------------	--------------------------------

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.2.70 - (instancetype) whereKeyExists: (NSString *) *key*

Add a constraint that requires a particular key exists.

Parameters

<i>key</i>	The key that should exist.
------------	----------------------------

Returns

The same instance of [PFQuery](#) as the receiver. This allows method chaining.

6.50.3 Property Documentation**6.50.3.1** - (PFCachePolicy) cachePolicy [read],[write],[atomic],[assign]

The cache policy to use for requests.

Not allowed when Pinning is enabled.

See also

- [fromLocalDatastore](#)
- [fromPin](#)
- [fromPinWithName:](#)

Definition at line 796 of file PFQuery.h.

6.50.3.2 - (NSInteger) limit [read],[write],[nonatomic],[assign]

A limit on the number of objects to return. The default limit is 100, with a maximum of 1000 results being returned at a time.

Warning

If you are calling `findObjects` with `limit = 1`, you may find it easier to use `getFirst` instead.

Definition at line 776 of file PFQuery.h.

6.50.3.3 - (NSTimeInterval) maxCacheAge [read],[write],[atomic],[assign]

Definition at line 801 of file PFQuery.h.

6.50.3.4 - (NSString*) parseClassName [read],[write],[nonatomic],[strong]

The class name to query for

Definition at line 76 of file PFQuery.h.

6.50.3.5 - (NSInteger) skip [read], [write], [nonatomic], [assign]

The number of objects to skip before returning any.

Definition at line 781 of file PFQuery.h.

6.50.3.6 - (BOOL) trace [read], [write], [nonatomic], [assign]

Whether or not performance tracing should be done on the query.

Warning

This should not be set to YES in most cases.

Definition at line 881 of file PFQuery.h.

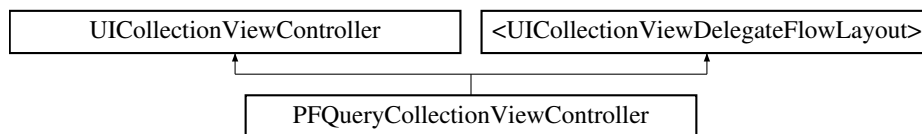
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFQuery.h

6.51 PFQueryCollectionViewController Class Reference

```
#import <PFQueryCollectionViewController.h>
```

Inheritance diagram for PFQueryCollectionViewController:



Instance Methods

Creating a PFQueryCollectionViewController

- (instancetype) - initWithClassName:
- (instancetype) - initWithCollectionViewLayout:className:

Responding to Events

- (void) - NS_REQUIRES_SUPER
- (void) - objectsDidLoad:

Loading Data

- (void) - loadObjects
- (void) - loadObjects:clear:
- (void) - loadNextPage
- (void) - clear

Querying

- (PFQuery *) - queryForCollection

Data Source Methods

- (PFUI_NULLABLE PFCollectionViewCell *) - collectionView:cellForItemAtIndexPath:object:
- (PFUI_NULLABLE UICollectionViewCell *) - collectionViewReusableCollectionViewForNextPageAction:

Properties

- IBInspectable NSString * [parseClassName](#)
- IBInspectable BOOL [loadingViewEnabled](#)
- IBInspectable BOOL [pullToRefreshEnabled](#)
- IBInspectable BOOL [paginationEnabled](#)
- IBInspectable NSUInteger [objectsPerPage](#)
- BOOL [loading](#)

Accessing Results

- NSArray * [objects](#)
- (PFUI_NULLABLE PFObjct *) - [objectAtIndexPath:](#)

6.51.1 Detailed Description

This class allows you to think about a one-to-one mapping between a <PFObjct> and a `UICollectionViewCell`, rather than having to juggle index paths.

You also get the following features out of the box:

- Pagination with a cell that can be tapped to load the next page.
- Pull-to-refresh collection view header.
- Automatic downloading and displaying of remote images in cells.
- Loading screen, shown before any data is loaded.
- Automatic loading and management of the objects array.
- Various methods that can be overridden to customize behavior at major events in the data cycle.

See also

[PFCollectionViewCell](#)

Definition at line 47 of file `PFQueryCollectionViewController.h`.

6.51.2 Method Documentation

6.51.2.1 - (void) clear

Clears the collection view of all objects.

6.51.2.2 - (PFUI_NULLABLE PFCollectionViewCell *) collectionView: (UICollectionView *) collectionView cellForItemAtIndexPath:(NSIndexPath *) indexPath object:(PFUI_NULLABLE PFObjct *) object

Override this method to customize each cell given a <PFObjct> that is loaded.

Warning

The cell should inherit from <PFCollectionViewCell> which is a subclass of `UICollectionViewCell`.

Parameters

<i>collectionView</i>	The collection view object associated with this controller.
<i>indexPath</i>	The indexPath of the cell.
<i>object</i>	The <PFObj> that is associated with the cell.

Returns

The cell that represents this object.

6.51.2.3 - (PFUI_NULLABLE UICollectionViewReusableV *) collectionViewReusableVForNextPageAction: (UICollectionView *) collectionView

Override this method to customize the view that allows the user to load the next page when pagination is turned on.

Parameters

<i>collectionView</i>	The collection view object associated with this controller.
-----------------------	---

Returns

The view that allows the user to paginate.

6.51.2.4 - (instancetype) initWithClassName: (PFUI_NULLABLE NSString *) className

Initializes a view controller with a UICollectionViewFlowLayout and a class name of <PFObj> that will be associated with this collection.

Parameters

<i>className</i>	The class name of the instances of <PFObj> that this table will display.
------------------	--

Returns

An initialized PFQueryCollectionViewController object or nil if the object couldn't be created.

6.51.2.5 - (instancetype) initWithCollectionViewLayout: (UICollectionViewLayout *) layout className:(PFUI_NULLABLE NSString *) NS_DESIGNATED_INITIALIZER

Initializes a view controller with a class name of <PFObj> that will be associated with this collection.

Parameters

<i>layout</i>	Layout for collection view to use.
<i>className</i>	The class name of the instances of <PFObj> that this table will display.

Returns

An initialized PFQueryCollectionViewController object or nil if the object couldn't be created.

6.51.2.6 - (void) loadNextPage

Loads the next page of objects, appends to table, and refreshes.

6.51.2.7 - (void) loadObjects

Clears the collection view and loads the first page of objects.

6.51.2.8 - (void) loadObjects: (NSInteger) page clear:(BOOL) clear

Loads the objects of the `<parseClassName>` at the specified page and appends it to the objects already loaded and refreshes the collection.

Parameters

<i>page</i>	The page of objects to load.
<i>clear</i>	Whether to clear the collection view after receiving the objects.

6.51.2.9 - (void) NS_REQUIRES_SUPER

Called when objects will be loaded from [Parse](#). If you override this method, you must call `[super objectsWillLoad]` in your implementation.

6.51.2.10 - (PFUI_NULLABLE PFObject *) objectAtIndex: (PFUI_NULLABLE NSIndexPath *) indexPath

Returns an object at a particular indexPath.

The default implementation returns the object at `indexPath.item`. If you want to return objects in a different indexPath order, like for sections, override this method.

Parameters

<i>indexPath</i>	An instance of <code>NSIndexPath</code> .
------------------	---

Returns

The object at the specified indexPath.

6.51.2.11 - (void) objectsDidLoad: (PFUI_NULLABLE NSError *) NS_REQUIRES_SUPER

Called when objects have loaded from [Parse](#). If you override this method, you must call `[super objectsDidLoad:]` in your implementation.

Parameters

<i>error</i>	The Parse error from running the PFQuery , if there was any.
--------------	--

6.51.2.12 - (PFQuery *) queryForCollection

Override to construct your own custom `<PFQuery>` to get the objects.

Returns

An instance of `<PFQuery>` that `<loadObjects>` method will use to the objects for this collection.

6.51.3 Property Documentation**6.51.3.1 - (BOOL) loading [read], [write], [nonatomic], [assign]**

Whether the collection is actively loading new data from the server.

Definition at line 77 of file PFQueryCollectionViewController.h.

6.51.3.2 - (IBInspectable BOOL) loadingViewEnabled [read], [write], [nonatomic], [assign]

Whether the collection should use the default loading view. Default - YES.

Definition at line 57 of file PFQueryCollectionViewController.h.

6.51.3.3 - (NSArray*) objects [read], [nonatomic], [copy]

The array of instances of <PFOBJECT> that is used as a data source.

Definition at line 128 of file PFQueryCollectionViewController.h.

6.51.3.4 - (IBInspectable NSInteger) objectsPerPage [read], [write], [nonatomic], [assign]

The number of objects to show per page. Default - 25.

Definition at line 72 of file PFQueryCollectionViewController.h.

6.51.3.5 - (IBInspectable BOOL) paginationEnabled [read], [write], [nonatomic], [assign]

Whether the collection should use the built-in pagination feature. Default - YES.

Definition at line 67 of file PFQueryCollectionViewController.h.

6.51.3.6 - (IBInspectable NSString*) parseClassName [read], [write], [nonatomic], [copy]

The class name of the <PFOBJECT> this collection will use as a datasource.

Definition at line 52 of file PFQueryCollectionViewController.h.

6.51.3.7 - (IBInspectable BOOL) pullToRefreshEnabled [read], [write], [nonatomic], [assign]

Whether the collection should use the built-in pull-to-refresh feature. Default - YES.

Definition at line 62 of file PFQueryCollectionViewController.h.

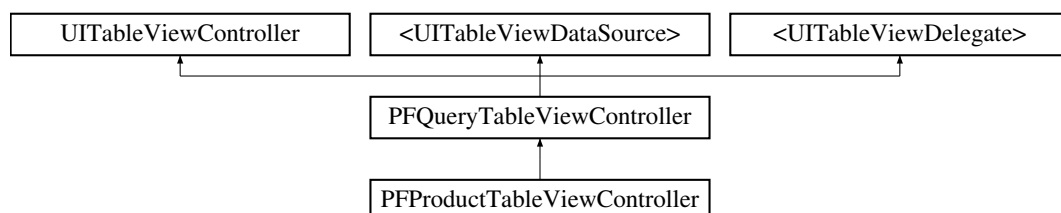
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/[PFQueryCollectionViewController.h](#)

6.52 PFQueryTableViewController Class Reference

```
#import <PFQueryTableViewController.h>
```

Inheritance diagram for PFQueryTableViewController:



Instance Methods

Creating a PFQueryTableViewController

- (instancetype) - [initWithStyle:className:](#)
- (instancetype) - [initWithClassName:](#)

Responding to Events

- (void) - [objectsWillLoad](#)
- (void) - [objectsDidLoad:](#)

Querying

- ([PFQuery *](#)) - [queryForTable](#)

Data Source Methods

- ([PFUI_NULLABLE PFTableViewCell *](#)) - [tableView:cellForRowAtIndexPath:object:](#)
- ([PFUI_NULLABLE PFTableViewCell *](#)) - [tableView:cellForNextPageAtIndexPath:](#)

Properties

Configuring Behavior

- IBInspectable NSString * [parseClassName](#)
- IBInspectable NSString * [textKey](#)
- IBInspectable NSString * [imageKey](#)
- IBInspectable UIImage * [placeholderImage](#)
- IBInspectable BOOL [loadingViewEnabled](#)
- IBInspectable BOOL [pullToRefreshEnabled](#)
- IBInspectable BOOL [paginationEnabled](#)
- IBInspectable NSUInteger [objectsPerPage](#)
- BOOL [loading](#)

Accessing Results

- NSArray * [objects](#)
- ([PFUI_NULLABLE PFObjct *](#)) - [objectAtIndexPath:](#)
- (void) - [clear](#)
- (void) - [loadObjects](#)
- (void) - [loadObjects:clear:](#)
- (void) - [loadNextPage](#)

6.52.1 Detailed Description

This class allows you to think about a one-to-one mapping between a `<PFObjct>` and a `UITableViewCell`, rather than having to juggle index paths.

You also get the following features out of the box:

- Pagination with a cell that can be tapped to load the next page.
- Pull-to-refresh table view header.
- Automatic downloading and displaying of remote images in cells.
- Loading screen, shown before any data is loaded.

- Automatic loading and management of the objects array.
- Various methods that can be overridden to customize behavior at major events in the data cycle.

Definition at line 45 of file PFQueryTableViewController.h.

6.52.2 Method Documentation

6.52.2.1 - (void) clear

Clears the table of all objects.

6.52.2.2 - (instancetype) initWithClassName: (PFUI_NULLABLE NSString *) className

Initializes with a class name of the PFObjets that will be associated with this table.

Parameters

<i>className</i>	The class name of the instances of <PFObjets> that this table will display.
------------------	---

Returns

An initialized [PFQueryTableViewController](#) object or `nil` if the object couldn't be created.

6.52.2.3 - (instancetype) initWithStyle: (UITableViewStyle) style className:(PFUI_NULLABLE NSString *) NS_DESIGNATED_INITIALIZER

Initializes with a class name of the <PFObjets> that will be associated with this table.

Parameters

<i>style</i>	The UITableViewStyle for the table
<i>className</i>	The class name of the instances of <PFObjets> that this table will display.

Returns

An initialized [PFQueryTableViewController](#) object or `nil` if the object couldn't be created.

6.52.2.4 - (void) loadNextPage

Loads the next page of objects, appends to table, and refreshes.

6.52.2.5 - (void) loadObjects

Clears the table and loads the first page of objects.

6.52.2.6 - (void) loadObjects: (NSInteger) page clear:(BOOL) clear

Loads the objects of the className at the specified page and appends it to the objects already loaded and refreshes the table.

Parameters

<i>page</i>	The page of objects to load.
<i>clear</i>	Whether to clear the table after receiving the objects.

6.52.2.7 - (PFUI_NULLABLE PFOBJECT *) objectAtIndexPath: (PFUI_NULLABLE NSIndexPath *) indexPath

Returns an object at a particular indexPath.

The default implementation returns the object at `indexPath.row`. If you want to return objects in a different indexPath order, like for sections, override this method.

Parameters

<i>indexPath</i>	The indexPath.
------------------	----------------

Returns

The object at the specified index

6.52.2.8 - (void) objectsDidLoad: (PFUI_NULLABLE NSError *) error

Called when objects have loaded from [Parse](#). If you override this method, you must call `[super objectsDidLoad:]` in your implementation.

Parameters

<i>error</i>	The Parse error from running the PFQuery , if there was any.
--------------	--

6.52.2.9 - (void) objectsWillLoad

Called when objects will loaded from [Parse](#). If you override this method, you must call `[super objectsWillLoad]` in your implementation.

6.52.2.10 - (PFQuery *) queryForTable

Override to construct your own custom [PFQuery](#) to get the objects.

Returns

[PFQuery](#) that loadObjects will use to the objects for this table.

**6.52.2.11 - (PFUI_NULLABLE PFTableViewCell *) tableView: (UITableView *) tableView
cellForNextPageAtIndexPath:(NSIndexPath *) indexPath**

Override this method to customize the cell that allows the user to load the next page when pagination is turned on.

Parameters

<i>tableView</i>	The table view object associated with this controller.
<i>indexPath</i>	The indexPath of the cell.

Returns

The cell that allows the user to paginate.

6.52.2.12 - (PFUI_NULLABLE PFTableViewCell *) tableView: (UITableView *) *tableView*
cellForRowAtIndexPath:(NSIndexPath *) *indexPath* object:(PFUI_NULLABLE PFObjec *) *object*

Override this method to customize each cell given a [PFObjec](#) that is loaded.

If you don't override this method, it will use a default style cell and display either the first data key from the object, or it will display the key as specified with `textKey`, `imageKey`.

Warning

The cell should inherit from <PFTableViewCell> which is a subclass of UITableViewCell.

Parameters

<i>tableView</i>	The table view object associated with this controller.
<i>indexPath</i>	The indexPath of the cell.
<i>object</i>	The PFObjec that is associated with the cell.

Returns

The cell that represents this object.

6.52.3 Property Documentation

6.52.3.1 - (IBInspectable NSString*) *imageKey* [read], [write], [nonatomic], [copy]

The key to use to display for the cell image view.

This won't apply if you override <tableView:cellForRowAtIndexPath:object:>

Definition at line 92 of file PFQueryTableViewCell.h.

6.52.3.2 - (BOOL) *loading* [read], [write], [nonatomic], [assign]

Whether the table is actively loading new data from the server.

Definition at line 124 of file PFQueryTableViewCell.h.

6.52.3.3 - (IBInspectable BOOL) *loadingViewEnabled* [read], [write], [nonatomic], [assign]

Whether the table should use the default loading view. Default - YES.

Definition at line 104 of file PFQueryTableViewCell.h.

6.52.3.4 - (NSArray*) *objects* [read], [nonatomic], [copy]

The array of instances of <PFObjec> that is used as a data source.

Definition at line 150 of file PFQueryTableViewCell.h.

6.52.3.5 - (IBInspectable NSUInteger) *objectsPerPage* [read], [write], [nonatomic], [assign]

The number of objects to show per page. Default - 25.

Definition at line 119 of file PFQueryTableViewCell.h.

6.52.3.6 - (IBInspectable BOOL) `paginationEnabled` [read], [write], [nonatomic], [assign]

Whether the table should use the built-in pagination feature. Default - YES.

Definition at line 114 of file `PFQueryTableViewController.h`.

6.52.3.7 - (IBInspectable NSString*) `parseClassName` [read], [write], [nonatomic], [copy]

The class name of the <PFObj> this table will use as a datasource.

Definition at line 78 of file `PFQueryTableViewController.h`.

6.52.3.8 - (IBInspectable UIImage*) `placeholderImage` [read], [write], [nonatomic], [strong]

The image to use as a placeholder for the cell images.

This won't apply if you override <tableView:cellForRowAtIndexPath:object:>

Definition at line 99 of file `PFQueryTableViewController.h`.

6.52.3.9 - (IBInspectable BOOL) `pullToRefreshEnabled` [read], [write], [nonatomic], [assign]

Whether the table should use the built-in pull-to-refresh feature. Default - YES.

Definition at line 109 of file `PFQueryTableViewController.h`.

6.52.3.10 - (IBInspectable NSString*) `textKey` [read], [write], [nonatomic], [copy]

The key to use to display for the cell text label.

This won't apply if you override <tableView:cellForRowAtIndexPath:object:>

Definition at line 85 of file `PFQueryTableViewController.h`.

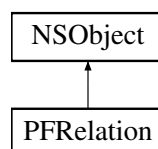
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFQueryTableViewController.h`

6.53 PFRelation Class Reference

```
#import <PFRelation.h>
```

Inheritance diagram for `PFRelation`:



Instance Methods

Accessing Objects

- (PF_NULLABLE PFQuery *) - `query`

Modifying Relations

- (void) - [addObject:](#)
- (void) - [removeObject:](#)

Properties

- NSString * [targetClass](#)

6.53.1 Detailed Description

The [PFRelation](#) class that is used to access all of the children of a many-to-many relationship. Each instance of [PFRelation](#) is associated with a particular parent object and key.

Definition at line 25 of file [PFRelation.h](#).

6.53.2 Method Documentation

6.53.2.1 - (void) addObject: (PFOBJECT *) object

Adds a relation to the passed in object.

Parameters

<i>object</i>	A <PFOBJECT> object to add relation to.
---------------	---

6.53.2.2 - (PF_NULLABLE PFQuery *) query

Returns a <PFQuery> object that can be used to get objects in this relation.

6.53.2.3 - (void) removeObject: (PFOBJECT *) object

Removes a relation to the passed in object.

Parameters

<i>object</i>	A <PFOBJECT> object to add relation to.
---------------	---

6.53.3 Property Documentation

6.53.3.1 - (NSString*) targetClass [read],[write],[nonatomic],[strong]

The name of the class of the target child objects.

Definition at line 30 of file [PFRelation.h](#).

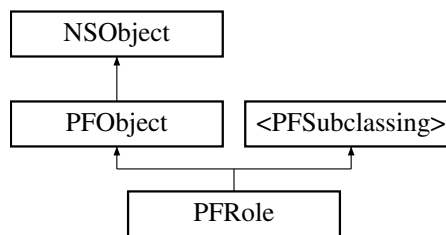
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFRelation.h](#)

6.54 PFRole Class Reference

```
#import <PFRole.h>
```

Inheritance diagram for `PFRole`:



Properties

Role-specific Properties

- `NSString *` `name`
- `PFRelation *` `users`
- `PFRelation *` `roles`

Creating a New Role

- (instancetype) - `initWithName:`
- (instancetype) - `initWithName:acl:`
- (instancetype) + `roleWithName:`
- (instancetype) + `roleWithName:acl:`

Additional Inherited Members

6.54.1 Detailed Description

The `PFRole` class represents a Role on the `Parse` server. `PFRoles` represent groupings of `<PFUser>` objects for the purposes of granting permissions (e.g. specifying a `<PFACL>` for a `<PFObject>`). Roles are specified by their sets of child users and child roles, all of which are granted any permissions that the parent role has.

Roles must have a name (which cannot be changed after creation of the role), and must specify an ACL.

Definition at line 30 of file `PFRole.h`.

6.54.2 Method Documentation

6.54.2.1 - (instancetype) initWithName: (NSString *) name

Constructs a new `PFRole` with the given name. If no default ACL has been specified, you must provide an ACL for the role.

Parameters

<i>name</i>	The name of the Role to create.
-------------	---------------------------------

6.54.2.2 - (instancetype) initWithName: (NSString *) name acl:(PF_NULLABLE PFACL *) acl

Constructs a new `PFRole` with the given name.

Parameters

<i>name</i>	The name of the Role to create.
<i>acl</i>	The ACL for this role. Roles must have an ACL.

6.54.2.3 + (instancetype) roleName: (NSString *) *name*

Constructs a new `PFRole` with the given name.

If no default ACL has been specified, you must provide an ACL for the role.

Parameters

<i>name</i>	The name of the Role to create.
-------------	---------------------------------

6.54.2.4 + (instancetype) roleName: (NSString *) *name* acl:(PF_NULLABLE PFACL *) *acl*

Constructs a new `PFRole` with the given name.

Parameters

<i>name</i>	The name of the Role to create.
<i>acl</i>	The ACL for this role. Roles must have an ACL.

6.54.3 Property Documentation

6.54.3.1 -(NSString*) name [read],[write],[nonatomic],[copy]

Gets or sets the name for a role.

This value must be set before the role has been saved to the server, and cannot be set once the role has been saved.

Warning

A role's name can only contain alphanumeric characters, `_`, `-`, and spaces.

Definition at line 81 of file `PFRole.h`.

6.54.3.2 -(PFRelation*) roles [read],[nonatomic],[strong]

Gets the `<PFRelation>` for the `PFRole` objects that are direct children of this role.

These roles' users are granted any privileges that this role has been granted (e.g. read or write access through ACLs). You can add or remove child roles from this role through this relation.

Definition at line 99 of file `PFRole.h`.

6.54.3.3 -(PFRelation*) users [read],[nonatomic],[strong]

Gets the `<PFRelation>` for the `<PFUser>` objects that are direct children of this role.

These users are granted any privileges that this role has been granted (e.g. read or write access through ACLs). You can add or remove users from the role through this relation.

Definition at line 90 of file `PFRole.h`.

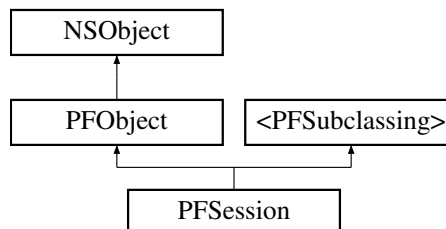
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/P↵FRole.h](#)

6.55 PFSession Class Reference

```
#import <PFSession.h>
```

Inheritance diagram for PFSession:



Class Methods

- (BFTask *) + [getCurrentSessionInBackground](#)
- (void) + [getCurrentSessionInBackgroundWithBlock:](#)

Properties

- NSString * [sessionToken](#)

Additional Inherited Members

6.55.1 Detailed Description

[PFSession](#) is a local representation of a session. This class is a subclass of a [<PFObject>](#), and retains the same functionality as any other subclass of [<PFObject>](#).

Definition at line 28 of file [PFSession.h](#).

6.55.2 Method Documentation

6.55.2.1 + (BFTask *) [getCurrentSessionInBackground](#)

Asynchronously* fetches a [PFSession](#) object related to the current user.

Returns

A task that is `completed` with an instance of [PFSession](#) class or is `faulted` if the operation fails.

6.55.2.2 + (void) [getCurrentSessionInBackgroundWithBlock:](#) (PF_NULLABLE PFSessionResultBlock) *block*

Asynchronously* fetches a [PFSession](#) object related to the current user.

Parameters

<i>block</i>	The block to execute when the operation completes. It should have the following argument signature: <code>^(PFSession *session, NSError *error).</code>
--------------	---

6.55.3 Property Documentation

6.55.3.1 `-(NSString*) sessionToken` `[read], [nonatomic], [copy]`

The session token string for this session.

Definition at line 33 of file `PFSession.h`.

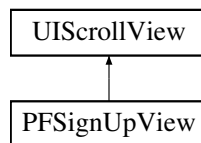
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFSession.h`

6.56 PFSignUpView Class Reference

```
#import <PFSignUpView.h>
```

Inheritance diagram for PFSignUpView:



Properties

Customizing the Logo

- `UIView *` `logo`

Configure Username Behaviour

- `BOOL` `emailAsUsername`

Sign Up Elements

- `PFSignUpFields` `fields`
- `PFTextField *` `usernameField`
- `PFTextField *` `passwordField`
- `PFTextField *` `emailField`
- `PFTextField *` `additionalField`
- `UIButton *` `signUpButton`
- `UIButton *` `dismissButton`

Creating Sign Up View

- `UIViewController *` `presentingViewController`
- `(instancetype) - initWithFields:`

6.56.1 Detailed Description

The `PFSignUpView` class provides a standard sign up interface for authenticating a `<PFUser>`.

Definition at line 57 of file `PFSignUpView.h`.

6.56.2 Method Documentation

6.56.2.1 `-(instancetype) initWithFields: (PFSignUpFields) fields`

Initializes the view with the specified sign up elements.

Parameters

<i>fields</i>	A bitmask specifying the sign up elements which are enabled in the view
---------------	---

Returns

An initialized `PFSignUpView` object or `nil` if the object couldn't be created.

See also

`PFSignUpFields`

6.56.3 Property Documentation

6.56.3.1 `-(PUITextField*) additionalField` `[read]`, `[nonatomic]`, `[strong]`

The additional text field. It is `nil` if the element is not enabled.

This field is intended to be customized.

Definition at line 130 of file `PFSignUpView.h`.

6.56.3.2 `-(UIButton*) dismissButton` `[read]`, `[nonatomic]`, `[strong]`

The dismiss button. It is `nil` if the element is not enabled.

Definition at line 140 of file `PFSignUpView.h`.

6.56.3.3 `-(BOOL) emailAsUsername` `[read]`, `[write]`, `[nonatomic]`, `[assign]`

If email should be used to log in, instead of username

By default, this is set to `NO`.

Definition at line 99 of file `PFSignUpView.h`.

6.56.3.4 `-(PUITextField*) emailField` `[read]`, `[nonatomic]`, `[strong]`

The email text field. It is `nil` if the element is not enabled.

Definition at line 123 of file `PFSignUpView.h`.

6.56.3.5 `-(PFSignUpFields) fields` `[read]`, `[nonatomic]`, `[assign]`

The bitmask which specifies the enabled sign up elements in the view

Definition at line 108 of file `PFSignUpView.h`.

6.56.3.6 `-(UIView*) logo` `[read], [write], [nonatomic], [strong]`

The logo. By default, it is the [Parse](#) logo.

Definition at line 88 of file PFSignUpView.h.

6.56.3.7 `-(PUITextField*) passwordField` `[read], [nonatomic], [strong]`

The password text field.

Definition at line 118 of file PFSignUpView.h.

6.56.3.8 `-(UIViewController*) presentingViewController` `[read], [write], [nonatomic], [weak]`

The view controller that will present this view.

Used to lay out elements correctly when the presenting view controller has translucent elements.

Definition at line 79 of file PFSignUpView.h.

6.56.3.9 `-(UIButton*) signUpButton` `[read], [nonatomic], [strong]`

The sign up button. It is `nil` if the element is not enabled.

Definition at line 135 of file PFSignUpView.h.

6.56.3.10 `-(PUITextField*) usernameField` `[read], [nonatomic], [strong]`

The username text field.

Definition at line 113 of file PFSignUpView.h.

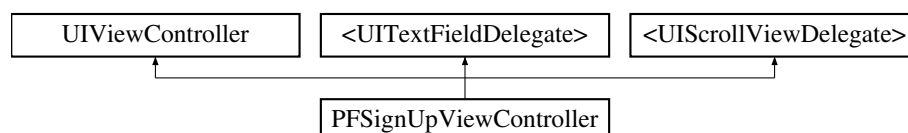
The documentation for this class was generated from the following file:

- `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFSignUpView.h`

6.57 PFSignUpViewController Class Reference

```
#import <PFSignUpViewController.h>
```

Inheritance diagram for PFSignUpViewController:



Properties

Configuring Sign Up Elements

- PFSignUpFields [fields](#)
- [PFSignUpView](#) * [signUpView](#)

Configuring Sign Up Behaviors

- id< [PFSignUpViewControllerDelegate](#) > [delegate](#)
- NSUInteger [minPasswordLength](#)
- BOOL [emailAsUsername](#)

6.57.1 Detailed Description

The [PFSignUpViewController](#) class that presents and manages a standard authentication interface for signing up a <PFUser>.

Definition at line 36 of file [PFSignUpViewController.h](#).

6.57.2 Property Documentation

6.57.2.1 - (id<[PFSignUpViewControllerDelegate](#)>) [delegate](#) [read], [write], [nonatomic], [weak]

The delegate that responds to the control events of [PFSignUpViewController](#).

See also

[PFSignUpViewControllerDelegate](#)

Definition at line 65 of file [PFSignUpViewController.h](#).

6.57.2.2 - (BOOL) [emailAsUsername](#) [read], [write], [nonatomic], [assign]

Whether to use the email as username on the attached <signUpView>.

If set to YES, we'll hide the email field, prompt for the email in the username field, and save the email into both username and email fields on the new <PFUser> object. By default, this is set to NO.

Definition at line 79 of file [PFSignUpViewController.h](#).

6.57.2.3 - (PFSignUpFields) [fields](#) [read], [write], [nonatomic], [assign]

A bitmask specifying the log in elements which are enabled in the view.

See also

[PFSignUpFields](#)

Definition at line 47 of file [PFSignUpViewController.h](#).

6.57.2.4 - (NSUInteger) [minPasswordLength](#) [read], [write], [nonatomic], [assign]

Minimum required password length for user signups, defaults to 0.

Definition at line 70 of file [PFSignUpViewController.h](#).

6.57.2.5 - (PFSignUpView*) [signUpView](#) [read], [nonatomic], [strong]

The sign up view. It contains all the enabled log in elements.

See also

[PFSignUpView](#)

Definition at line 54 of file PFSignUpViewController.h.

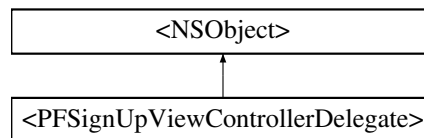
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/[PFSignUpViewController.h](#)

6.58 <PFSignUpViewControllerDelegate> Protocol Reference

```
#import <PFSignUpViewController.h>
```

Inheritance diagram for <PFSignUpViewControllerDelegate>:



Instance Methods

Customizing Behavior

- (BOOL) - [signUpViewController:shouldBeginSignUp:](#)

Responding to Actions

- (void) - [signUpViewController:didSignUpUser:](#)
- (void) - [signUpViewController:didFailToSignUpWithError:](#)
- (void) - [signUpViewControllerDidCancelSignUp:](#)

6.58.1 Detailed Description

The [PFLoginViewControllerDelegate](#) protocol defines methods a delegate of a <PFSignUpViewController> should implement. All methods of this protocol are optional.

Definition at line 112 of file PFSignUpViewController.h.

6.58.2 Method Documentation

6.58.2.1 - (void) signUpViewController: (PFSignUpViewController *) *signUpController* didFailToSignUpWithError:(PFUI_NULLABLE NSError *) *error*

Sent to the delegate when the sign up attempt fails.

Parameters

<i>signUpController</i>	The signup view controller where signup failed.
-------------------------	---

<i>error</i>	<code>NSError</code> object representing the error that occurred.
--------------	---

6.58.2.2 - (void) signUpViewController: (PFSignUpViewController *) *signUpController* didSignUpUser:(PFUser *) *user*

Sent to the delegate when a <PFUser> is signed up.

Parameters

<i>signUpController</i>	The signup view controller where signup finished.
<i>user</i>	<PFUser> object that is a result of the sign up.

6.58.2.3 - (BOOL) signUpViewController: (PFSignUpViewController *) *signUpController* shouldBeginSignUp:(NSDictionary *) *info*

Sent to the delegate to determine whether the sign up request should be submitted to the server.

Parameters

<i>signUpController</i>	The signup view controller that is requesting the data.
<i>info</i>	An <code>NSDictionary</code> instance which contains all sign up information that the user entered.

Returns

A `BOOL` indicating whether the sign up should proceed.

6.58.2.4 - (void) signUpViewControllerDidCancelSignUp: (PFSignUpViewController *) *signUpController*

Sent to the delegate when the sign up screen is cancelled.

Parameters

<i>signUpController</i>	The signup view controller where signup was cancelled.
-------------------------	--

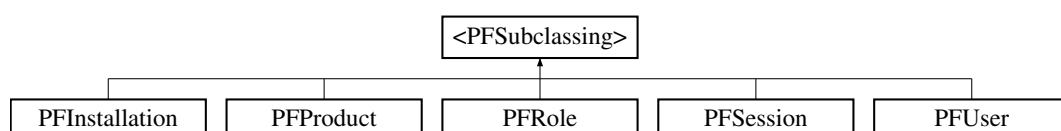
The documentation for this protocol was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/[PFSignUpViewController.h](#)

6.59 <PFSubclassing> Protocol Reference

```
#import <PFSubclassing.h>
```

Inheritance diagram for <PFSubclassing>:



Class Methods

- (instancetype) + [object](#)

- (instancetype) + [objectWithoutDataWithObjectId:](#)
- (NSString *) + [parseClassName](#)
- (PF_NULLABLE PFQuery *) + [query](#)
- (PF_NULLABLE PFQuery *) + [queryWithPredicate:](#)
- (void) + [registerSubclass](#)

6.59.1 Detailed Description

If a subclass of <PFObjec> conforms to [PFSubclassing](#) and calls <registerSubclass>, [Parse](#) framework will be able to use that class as the native class for a [Parse](#) cloud object.

Classes conforming to this protocol should subclass <PFObjec> and include `PFObjec+Subclass.h` in their implementation file. This ensures the methods in the Subclass category of <PFObjec> are exposed in its subclasses only.

Definition at line 27 of file `PFSubclassing.h`.

6.59.2 Method Documentation

6.59.2.1 + (instancetype) object

Constructs an object of the most specific class known to implement <parseClassName>.

This method takes care to help <PFObjec> subclasses be subclassed themselves. For example, `[PFUser object]` returns a <PFUser> by default but will return an object of a registered subclass instead if one is known. A default implementation is provided by <PFObjec> which should always be sufficient.

Returns

Returns the object that is instantiated.

6.59.2.2 + (instancetype) objectWithoutDataWithObjectId: (PF_NULLABLE NSString *) objectId

Creates a reference to an existing [PFObjec](#) for use in creating associations between [PFObjec](#)s.

Calling <[[PFObjec](#) isDataAvailable]> on this object will return NO until <[[PFObjec](#) fetchIfNeeded]> has been called. No network request will be made. A default implementation is provided by [PFObjec](#) which should always be sufficient.

Parameters

<i>objectId</i>	The object id for the referenced object.
-----------------	--

Returns

A new <PFObjec> without data.

6.59.2.3 + (NSString *) parseClassName

The name of the class as seen in the REST API.

6.59.2.4 + (PF_NULLABLE PFQuery *) query

Create a query which returns objects of this type.

A default implementation is provided by <PFObjec> which should always be sufficient.

Reimplemented in [PFInstallation](#).

6.59.2.5 + (PF_NULLABLE PFQuery *) queryWithPredicate: (PF_NULLABLE NSPredicate *) predicate

Returns a query for objects of this type with a given predicate.

A default implementation is provided by <PFObj> which should always be sufficient.

Parameters

<i>predicate</i>	The predicate to create conditions from.
------------------	--

Returns

An instance of <PFQuery>.

See also

[[PFQuery](#) queryWithClassName:predicate:]

6.59.2.6 + (void) registerSubclass

Lets [Parse](#) know this class should be used to instantiate all objects with class type <parseClassName>.

Warning

This method must be called before <[[Parse](#) setApplicationId:clientKey:]>

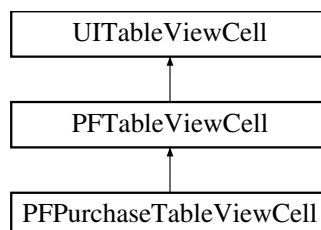
The documentation for this protocol was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[P↵FSubclassing.h](#)

6.60 PFTableViewCell Class Reference

```
#import <PFTableViewCell.h>
```

Inheritance diagram for PFTableViewCell:



Properties

- [PFIImageView](#) * *imageView*

6.60.1 Detailed Description

The [PFTableViewCell](#) class represents a table view cell which can download and display remote images stored on [Parse](#).

When used in a <PFQueryTableViewController> - downloading and displaying of the remote images are automatically managed by the <PFQueryTableViewController>.

Definition at line 35 of file PFTableViewCell.h.

6.60.2 Property Documentation

6.60.2.1 -(PFIImageView*) imageView [read],[nonatomic],[strong]

The imageView of the table view cell.

See also

[PFIImageView](#)

Definition at line 42 of file PFTableViewCell.h.

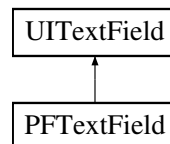
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/↔ Headers/[PFTableViewCell.h](#)

6.61 PUITextField Class Reference

```
#import <PUITextField.h>
```

Inheritance diagram for PUITextField:



Instance Methods

- (instancetype) - [initWithFrame:separatorStyle:](#)

Properties

- PUITextFieldSeparatorStyle [separatorStyle](#)
- UIColor *separatorColor [UI_APPEARANCE_SELECTOR](#)

6.61.1 Detailed Description

[PUITextField](#) class serves as a stylable subclass of [UITextField](#). It includes styles that are specific to ParseUI framework and allows advanced customization.

Definition at line 47 of file PUITextField.h.

6.61.2 Method Documentation

6.61.2.1 -(instancetype) initWithFrame: (CGRect) frame separatorStyle:(PUITextFieldSeparatorStyle) separatorStyle

This method is a convenience initializer that sets both `frame` and `separatorStyle` for an instance of [PF↵extField](#).

Parameters

<i>frame</i>	The frame rectangle for the view, measured in points.
<i>separatorStyle</i>	Initial separator style to use.

Returns

An initialized instance of [PTextField](#) or `nil` if it couldn't be created.

6.61.3 Property Documentation

6.61.3.1 - ([PTextFieldSeparatorStyle](#)) `separatorStyle` [read], [write], [nonatomic], [assign]

Separator style bitmask that should be applied to this textfield.

Default: `<PTextFieldSeparatorStyleNone>`

See also

[PTextFieldSeparatorStyle](#)

Definition at line 56 of file [PTextField.h](#).

6.61.3.2 - ([UIColor*](#) `separatorColor`) `UI_APPEARANCE_SELECTOR` [read], [write], [nonatomic], [strong]

Color that should be used for the separators, if they are visible.

Default: `227, 227, 227, 1.0`.

Definition at line 63 of file [PTextField.h](#).

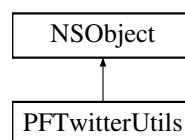
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PTextField.h](#)

6.62 PFTwitterUtils Class Reference

```
#import <PFTwitterUtils.h>
```

Inheritance diagram for PFTwitterUtils:



Class Methods

Interacting With Twitter

- ([PF_NULLABLE PF_Twitter *](#)) + [twitter](#)
- (void) + [initWithConsumerKey:consumerSecret:](#)
- (BOOL) + [isLinkedWithUser:](#)

Logging In & Creating Twitter-Linked Users

- (BFTask *) + [loginInBackground](#)
- (void) + [loginWithBlock:](#)
- (void) + [loginWithTarget:selector:](#)
- (BFTask *) + [loginWithTwitterIdInBackground:screenName:authToken:authTokenSecret:](#)
- (void) + [loginWithTwitterId:screenName:authToken:authTokenSecret:block:](#)
- (void) + [loginWithTwitterId:screenName:authToken:authTokenSecret:target:selector:](#)

Linking Users with Twitter

- (void) + [linkUser:](#)
- (BFTask *) + [linkUserInBackground:](#)
- (void) + [linkUser:block:](#)
- (void) + [linkUser:target:selector:](#)
- (BFTask *) + [linkUserInBackground:twitterId:screenName:authToken:authTokenSecret:](#)
- (void) + [linkUser:twitterId:screenName:authToken:authTokenSecret:block:](#)
- (void) + [linkUser:twitterId:screenName:authToken:authTokenSecret:target:selector:](#)

Unlinking Users from Twitter

- (BOOL) + [unlinkUser:](#)
- (BOOL) + [unlinkUser:error:](#)
- (BFTask *) + [unlinkUserInBackground:](#)
- (void) + [unlinkUserInBackground:block:](#)
- (void) + [unlinkUserInBackground:target:selector:](#)

6.62.1 Detailed Description

The [PFTwitterUtils](#) class provides utility functions for working with Twitter in a [Parse](#) application.

This class is currently for iOS only.

Definition at line 22 of file [PFTwitterUtils.h](#).

6.62.2 Method Documentation

6.62.2.1 + (void) initializeWithConsumerKey: (NSString *) consumerKey consumerSecret:(NSString *) consumerSecret

Initializes the Twitter singleton.

Warning

You must invoke this in order to use the Twitter functionality in [Parse](#).

Parameters

<i>consumerKey</i>	Your Twitter application's consumer key.
<i>consumerSecret</i>	Your Twitter application's consumer secret.

6.62.2.2 + (BOOL) isLinkedWithUser: (PF_NULLABLE PFUser *) user

Whether the user has their account linked to Twitter.

Parameters

<i>user</i>	User to check for a Twitter link. The user must be logged in on this device.
-------------	--

Returns

YES if the user has their account linked to Twitter, otherwise NO.

6.62.2.3 + (void) linkUser: ("Please use +linkUserInBackground: instead.") *PARSE_DEPRECATED*

Asynchronously links Twitter to an existing [PFUser](#).

This method delegates to Twitter to authenticate the user, and then automatically links the account to the <PF↔User>.

Parameters

<i>user</i>	User to link to Twitter.
-------------	--------------------------

Deprecated Please use [[PFTwitterUtils linkUserInBackground:](#)] instead.

6.62.2.4 + (void) linkUser: (PFUser *) *user* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously links Twitter to an existing <PFUser>.

This method delegates to Twitter to authenticate the user, and then automatically links the account to the <PF↔User>.

Parameters

<i>user</i>	User to link to Twitter.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOO↔L *success, NSError *error).

6.62.2.5 + (void) linkUser: (PFUser *) *user* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.62.2.6 + (void) linkUser: (PFUser *) *user* twitterId:(NSString *) *twitterId* screenName:(NSString *) *screenName* authToken:(NSString *) *authToken* authTokenSecret:(NSString *) *authTokenSecret* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Asynchronously links Twitter to an existing <PFUser>.

you to handle user login to Twitter, then provide authentication data to link the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Twitter.
<i>twitterId</i>	The id of the Twitter user being linked.
<i>screenName</i>	The screen name of the Twitter user being linked.
<i>authToken</i>	The auth token for the user's session.
<i>authTokenSecret</i>	The auth token secret for the user's session.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOO↔L *success, NSError *error).

6.62.2.7 + (void) linkUser: (PFUser *) *user* twitterId:(NSString *) *twitterId* screenName:(NSString *) *screenName* authToken:(NSString *) *authToken* authTokenSecret:(NSString *) *authTokenSecret* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

6.62.2.8 + (BFTask *) linkUserInBackground: (PFUser *) user

Asynchronously links Twitter to an existing <PFUser>.

This method delegates to Twitter to authenticate the user, and then automatically links the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Twitter.
-------------	--------------------------

Returns

The task, that encapsulates the work being done.

6.62.2.9 + (BFTask *) linkUserInBackground: (PFUser *) user twitterId:(NSString *) twitterId screenName:(NSString *) screenName authToken:(NSString *) authToken authTokenSecret:(NSString *) authTokenSecret

Asynchronously links Twitter to an existing [PFUser](#) asynchronously.

Allows you to handle user login to Twitter, then provide authentication data to link the account to the <PFUser>.

Parameters

<i>user</i>	User to link to Twitter.
<i>twitterId</i>	The id of the Twitter user being linked.
<i>screenName</i>	The screen name of the Twitter user being linked.
<i>authToken</i>	The auth token for the user's session.
<i>authTokenSecret</i>	The auth token secret for the user's session.

Returns

The task, that encapsulates the work being done.

6.62.2.10 + (BFTask *) loginInBackground

Asynchronously logs in a user using Twitter.

This method delegates to Twitter to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) a <PFUser>.

Returns

The task, that encapsulates the work being done.

6.62.2.11 + (void) loginWithBlock: (PF_NULLABLE PFUserResultBlock) block

Asynchronously logs in a user using Twitter.

This method delegates to Twitter to authenticate the user, and then automatically logs in (or creates, in the case where it is a new user) <PFUser>.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (PFUser *user, NSError *error).
--------------	--

6.62.2.12 + (void) loginWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.62.2.13 + (void) loginWithTwitterId: (NSString *) twitterId screenName:(NSString *) screenName authToken:(NSString *) authToken authTokenSecret:(NSString *) authTokenSecret block:(PF_NULLABLE PFUserResultBlock) block

Logs in a user using Twitter.

Allows you to handle user login to Twitter, then provide authentication data to log in (or create, in the case where it is a new user) the <PFUser>.

Parameters

<i>twitterId</i>	The id of the Twitter user being linked
<i>screenName</i>	The screen name of the Twitter user being linked
<i>authToken</i>	The auth token for the user's session
<i>authTokenSecret</i>	The auth token secret for the user's session
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (PFUser *user, NSError *error).

6.62.2.14 + (void) loginWithTwitterId: (NSString *) twitterId screenName:(NSString *) screenName authToken:(NSString *) authToken authTokenSecret:(NSString *) authTokenSecret target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.62.2.15 + (BFTask *) loginWithTwitterIdInBackground: (NSString *) twitterId screenName:(NSString *) screenName authToken:(NSString *) authToken authTokenSecret:(NSString *) authTokenSecret

Asynchronously logs in a user using Twitter.

Allows you to handle user login to Twitter, then provide authentication data to log in (or create, in the case where it is a new user) the <PFUser>.

Parameters

<i>twitterId</i>	The id of the Twitter user being linked.
<i>screenName</i>	The screen name of the Twitter user being linked.
<i>authToken</i>	The auth token for the user's session.
<i>authTokenSecret</i>	The auth token secret for the user's session.

Returns

The task, that encapsulates the work being done.

6.62.2.16 + (PF_NULLABLE PF_Twitter *) twitter

Gets the instance of the <PF_Twitter> object that [Parse](#) uses.

Returns

An instance of <PF_Twitter> object.

6.62.2.17 + (BOOL) unlinkUser: (PFUser *) user

Synchronously unlinks the <PFUser> from a Twitter account.

Parameters

<i>user</i>	User to unlink from Twitter.
-------------	------------------------------

Returns

Returns true if the unlink was successful.

6.62.2.18 + (BOOL) unlinkUser: (PFUser *) *user* error:(NSError **) *error*

Synchronously unlinks the [PFUser](#) from a Twitter account.

Parameters

<i>user</i>	User to unlink from Twitter.
<i>error</i>	Error object to set on error.

Returns

Returns YES if the unlink was successful, otherwise NO.

6.62.2.19 + (BFTask *) unlinkUserInBackground: (PFUser *) *user*

Makes an *asynchronous* request to unlink a user from a Twitter account.

Parameters

<i>user</i>	User to unlink from Twitter.
-------------	------------------------------

Returns

The task, that encapsulates the work being done.

6.62.2.20 + (void) unlinkUserInBackground: (PFUser *) *user* block:(PF_NULLABLE PFBooleanResultBlock) *block*

Makes an *asynchronous* request to unlink a user from a Twitter account.

Parameters

<i>user</i>	User to unlink from Twitter.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

6.62.2.21 + (void) unlinkUserInBackground: (PFUser *) *user* target:(PF_NULLABLE_S id) *target* selector:(PF_NULLABLE_S SEL) *selector*

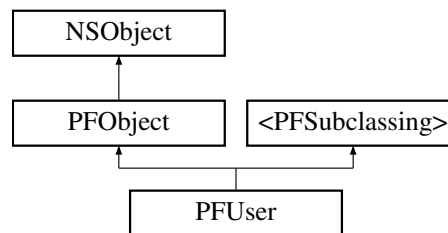
The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/[PFTwitterUtils.h](#)

6.63 PFUser Class Reference

```
#import <PFUser.h>
```

Inheritance diagram for PFUser:



Class Methods

Logging In

- (PF_NULLABLE instancetype) + [logInWithUsername:password:](#)
- (PF_NULLABLE instancetype) + [logInWithUsername:password:error:](#)
- (BFTask *) + [logInWithUsernameInBackground:password:](#)
- (void) + [logInWithUsernameInBackground:password:target:selector:](#)
- (void) + [logInWithUsernameInBackground:password:block:](#)

Becoming a User

- (PF_NULLABLE instancetype) + [become:](#)
- (PF_NULLABLE instancetype) + [become:error:](#)
- (BFTask *) + [becomeInBackground:](#)
- (void) + [becomeInBackground:target:selector:](#)
- (void) + [becomeInBackground:block:](#)

Revocable Session

- (BFTask *) + [enableRevocableSessionInBackground](#)
- (void) + [enableRevocableSessionInBackgroundWithBlock:](#)

Logging Out

- (void) + [logOut](#)
- (BFTask *) + [logOutInBackground](#)
- (void) + [logOutInBackgroundWithBlock:](#)

Requesting a Password Reset

- (BOOL) + [requestPasswordResetForEmail:](#)
- (BOOL) + [requestPasswordResetForEmail:error:](#)
- (BFTask *) + [requestPasswordResetForEmailInBackground:](#)
- (void) + [requestPasswordResetForEmailInBackground:target:selector:](#)
- (void) + [requestPasswordResetForEmailInBackground:block:](#)

Accessing the Current User

- NSString * [sessionToken](#)
- BOOL [isNew](#)
- (PF_NULLABLE instancetype) + [currentUser](#)
- (BOOL) - [isAuthenticated](#)

Creating a New User

- NSString * [username](#)
- NSString * [password](#)
- NSString * [email](#)
- (PFUser *) + [user](#)
- (void) + [enableAutomaticUser](#)
- (BOOL) - [signUp](#)
- (BOOL) - [signUp:](#)
- (BFTask *) - [signUpInBackground](#)
- (void) - [signUpInBackgroundWithBlock:](#)
- (void) - [signUpInBackgroundWithTarget:selector:](#)

Additional Inherited Members

6.63.1 Detailed Description

The [PFUser](#) class is a local representation of a user persisted to the [Parse](#) Data. This class is a subclass of a [<PFOBJECT>](#), and retains the same functionality of a [<PFOBJECT>](#), but also extends it with various user specific methods, like authentication, signing up, and validation uniqueness.

Many APIs responsible for linking a [PFUser](#) with Facebook or Twitter have been deprecated in favor of dedicated utilities for each social network. See [<PFFacebookUtils>](#), [<PFTwitterUtils>](#) and [<PFAnonymousUtils>](#) for more information.

Definition at line 36 of file [PFUser.h](#).

6.63.2 Method Documentation

6.63.2.1 + (PF_NULLABLEinstancetype) become: (NSString *) *sessionToken*

Makes a *synchronous* request to become a user with the given session token.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to [<currentUser>](#) will use the latest logged in user.

Parameters

<i>sessionToken</i>	The session token for the user.
---------------------	---------------------------------

Returns

Returns an instance of the [PFUser](#) on success. If becoming a user fails due to incorrect token, it returns *nil*.

6.63.2.2 + (PF_NULLABLEinstancetype) become: (NSString *) *sessionToken* error:(NSError **) *error*

Makes a *synchronous* request to become a user with the given session token.

Returns an instance of the successfully logged in [PFUser](#). This will also cache the user locally so that calls to [<currentUser>](#) will use the latest logged in user.

Parameters

<i>sessionToken</i>	The session token for the user.
<i>error</i>	The error object to set on error.

Returns

Returns an instance of the [PFUser](#) on success. If becoming a user fails due to incorrect token, it returns `nil`.

6.63.2.3 + (BFTask *) becomeInBackground: (NSString *) sessionToken

Makes an *asynchronous* request to become a user with the given session token.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to `<currentUser>` will use the latest logged in user.

Parameters

<i>sessionToken</i>	The session token for the user.
---------------------	---------------------------------

Returns

The task, that encapsulates the work being done.

6.63.2.4 + (void) becomeInBackground: (NSString *) sessionToken block:(PF_NULLABLE PFUserResultBlock) block

Makes an *asynchronous* request to become a user with the given session token.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to `<currentUser>` will use the latest logged in user.

Parameters

<i>sessionToken</i>	The session token for the user.
<i>block</i>	The block to execute. The block should have the following argument signature: (PFUser *user, NSError *error)

6.63.2.5 + (void) becomeInBackground: (NSString *) sessionToken target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector**6.63.2.6 + (PF_NULLABLEinstancetype) currentUser**

Gets the currently logged in user from disk and returns an instance of it.

Returns

Returns a [PFUser](#) that is the currently logged in user. If there is none, returns `nil`.

6.63.2.7 + (void) enableAutomaticUser

Enables automatic creation of anonymous users.

After calling this method, `<currentUser>` will always have a value. The user will only be created on the server once the user has been saved, or once an object with a relation to that user or an ACL that refers to the user has been saved.

Warning

<[PFObject saveEventually]> will not work on if an item being saved has a relation to an automatic user that has never been saved.

6.63.2.8 + (BFTask *) enableRevocableSessionInBackground

Enables revocable sessions and migrates the currentUser session token to use revocable session if needed.

This method is required if you want to use <PFSession> APIs and you application's 'Require Revocable Session' setting is turned off on <http://parse.com> app settings. After returned BFTask completes - <PFSession> class and APIs will be available for use.

Returns

An instance of BFTask that is completed when revocable sessions are enabled and currentUser token is migrated.

6.63.2.9 + (void) enableRevocableSessionInBackgroundWithBlock: (PF_NULLABLE PFUserSessionUpgradeResult↔ Block) block

Enables revocable sessions and upgrades the currentUser session token to use revocable session if needed.

This method is required if you want to use <PFSession> APIs and legacy sessions are enabled in your application settings on <http://parse.com/>. After returned BFTask completes - <PFSession> class and APIs will be available for use.

Parameters

<i>block</i>	Block that will be called when revocable sessions are enabled and currentUser token is migrated.
--------------	--

6.63.2.10 - (BOOL) isAuthenticated

Whether the user is an authenticated object for the device.

An authenticated PFUser is one that is obtained via a <signUp> or <login> method. An authenticated object is required in order to save (with altered values) or delete it.

Returns

Returns whether the user is authenticated.

6.63.2.11 + (PF_NULLABLEinstancetype) loginWithUsername: (NSString *) username password:(NSString *) password

Makes a *synchronous* request to login a user with specified credentials.

Returns an instance of the successfully logged in PFUser. This also caches the user locally so that calls to <currentUser> will use the latest logged in user.

Parameters

<i>username</i>	The username of the user.
-----------------	---------------------------

<i>password</i>	The password of the user.
-----------------	---------------------------

Returns

Returns an instance of the [PFUser](#) on success. If login failed for either wrong password or wrong username, returns `nil`.

6.63.2.12 + (PF_NULLABLEinstancetype) loginWithUsername: (NSString *) *username* password:(NSString *) *password* error:(NSError **) *error*

Makes a *synchronous* request to login a user with specified credentials.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to `<currentUser>` will use the latest logged in user.

Parameters

<i>username</i>	The username of the user.
<i>password</i>	The password of the user.
<i>error</i>	The error object to set on error.

Returns

Returns an instance of the [PFUser](#) on success. If login failed for either wrong password or wrong username, returns `nil`.

6.63.2.13 + (BFTask *) loginWithUsernameInBackground: (NSString *) *username* password:(NSString *) *password*

Makes an *asynchronous* request to login a user with specified credentials.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to `<currentUser>` will use the latest logged in user.

Parameters

<i>username</i>	The username of the user.
<i>password</i>	The password of the user.

Returns

The task, that encapsulates the work being done.

6.63.2.14 + (void) loginWithUsernameInBackground: (NSString *) *username* password:(NSString *) *password* block:(PF_NULLABLE PFUserResultBlock) *block*

Makes an *asynchronous* request to log in a user with specified credentials.

Returns an instance of the successfully logged in [PFUser](#). This also caches the user locally so that calls to `<currentUser>` will use the latest logged in user.

Parameters

<i>username</i>	The username of the user.
-----------------	---------------------------

<i>password</i>	The password of the user.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (PFUser *user, NSError *error).

6.63.2.15 + (void) loginWithUsernameInBackground: (NSString *) username password:(NSString *) password
target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.63.2.16 + (void) logOut

Synchronously logs out the currently logged in user on disk.

6.63.2.17 + (BFTask *) logOutInBackground

Asynchronously logs out the currentlt logged in user.

This will also remove the session from disk, log out of linked services and all future calls to <currentUser> will return *nil*. This is preferable to using <logOut>, unless your code is already running from a background thread.

Returns

An instance of *BFTask*, that is resolved with *nil* result when logging out completes.

6.63.2.18 + (void) logOutInBackgroundWithBlock: (PF_NULLABLE PFUserLogoutResultBlock) block

Asynchronously logs out the currentlt logged in user.

This will also remove the session from disk, log out of linked services and all future calls to <currentUser> will return *nil*. This is preferable to using <logOut>, unless your code is already running from a background thread.

Parameters

<i>block</i>	A block that will be called when logging out completes or fails.
--------------	--

6.63.2.19 + (BOOL) requestPasswordResetForEmail: (NSString *) email

Synchronously Send a password reset request for a specified email.

If a user account exists with that email, an email will be sent to that address with instructions on how to reset their password.

Parameters

<i>email</i>	Email of the account to send a reset password request.
--------------	--

Returns

Returns YES if the reset email request is successful. NO - if no account was found for the email address.

6.63.2.20 + (BOOL) requestPasswordResetForEmail: (NSString *) email error:(NSError **) error

Synchronously send a password reset request for a specified email and sets an error object.

If a user account exists with that email, an email will be sent to that address with instructions on how to reset their password.

Parameters

<i>email</i>	Email of the account to send a reset password request.
<i>error</i>	Error object to set on error.

Returns

Returns YES if the reset email request is successful. NO - if no account was found for the email address.

6.63.2.21 + (BFTask *) requestPasswordResetForEmailInBackground: (NSString *) email

Send a password reset request asynchronously for a specified email and sets an error object. If a user account exists with that email, an email will be sent to that address with instructions on how to reset their password.

Parameters

<i>email</i>	Email of the account to send a reset password request.
--------------	--

Returns

The task, that encapsulates the work being done.

6.63.2.22 + (void) requestPasswordResetForEmailInBackground: (NSString *) email block:(PF_NULLABLE PFBooleanResultBlock) block

Send a password reset request *asynchronously* for a specified email.

If a user account exists with that email, an email will be sent to that address with instructions on how to reset their password.

Parameters

<i>email</i>	Email of the account to send a reset password request.
<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).

6.63.2.23 + (void) requestPasswordResetForEmailInBackground: (NSString *) email target:(PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector

6.63.2.24 - (BOOL) signUp

Signs up the user *synchronously*.

This will also enforce that the username isn't already taken.

Warning

Make sure that password and username are set before calling this method.

Returns

Returns YES if the sign up was successful, otherwise NO.

6.63.2.25 - (BOOL) signUp: (NSError **) error

Signs up the user *synchronously*.

This will also enforce that the username isn't already taken.

Warning

Make sure that password and username are set before calling this method.

Parameters

<i>error</i>	Error object to set on error.
--------------	-------------------------------

Returns

Returns whether the sign up was successful.

6.63.2.26 - (BFTask *) signUpInBackground

Signs up the user *asynchronously*.

This will also enforce that the username isn't already taken.

Warning

Make sure that password and username are set before calling this method.

Returns

The task, that encapsulates the work being done.

6.63.2.27 - (void) signUpInBackgroundWithBlock: (PF_NULLABLE PFBooleanResultBlock) block

Signs up the user *asynchronously*.

This will also enforce that the username isn't already taken.

Warning

Make sure that password and username are set before calling this method.

Parameters

<i>block</i>	The block to execute. It should have the following argument signature: \wedge (BOOL \leftrightarrow L succeeded, NSError *error).
--------------	---

6.63.2.28 - (void) signUpInBackgroundWithTarget: (PF_NULLABLE_S id) target selector:(PF_NULLABLE_S SEL) selector**6.63.2.29 + (PFUser *) user**

Creates a new [PFUser](#) object.

Returns

Returns a new [PFUser](#) object.

6.63.3 Property Documentation

6.63.3.1 `-(NSString*) email` `[read], [write], [nonatomic], [strong]`

The email for the [PFUser](#).

Definition at line 112 of file PFUser.h.

6.63.3.2 `-(BOOL) isNew` `[read], [atomic], [assign]`

Whether the [PFUser](#) was just created from a request.

This is only set after a Facebook or Twitter login.

Definition at line 61 of file PFUser.h.

6.63.3.3 `-(NSString*) password` `[read], [write], [nonatomic], [strong]`

! The password for the [PFUser](#).

This will not be filled in from the server with the password. It is only meant to be set.

Definition at line 107 of file PFUser.h.

6.63.3.4 `-(NSString*) sessionToken` `[read], [nonatomic], [copy]`

The session token for the [PFUser](#).

This is set by the server upon successful authentication.

Definition at line 54 of file PFUser.h.

6.63.3.5 `-(NSString*) username` `[read], [write], [nonatomic], [strong]`

The username for the [PFUser](#).

Definition at line 99 of file PFUser.h.

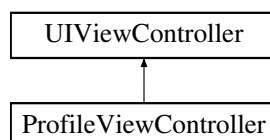
The documentation for this class was generated from the following file:

- [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFUser.h](#)

6.64 ProfileViewController Class Reference

```
#import <ProfileViewController.h>
```

Inheritance diagram for ProfileViewController:



Instance Methods

- (IBAction) - [logoutButton:](#)
logout current user and go to login page
- (IBAction) - [deleteButton:](#)
delete account if user quits the game
- (IBAction) - [saveChanges:](#)
Update User name.

Properties

- IBOutlet UILabel * [userNameLabel](#)
- IBOutlet UILabel * [emailLabel](#)
- IBOutlet UILabel * [highScoreLabel](#)
- IBOutlet UITextField * [userNameField](#)

6.64.1 Detailed Description

Definition at line 12 of file ProfileViewController.h.

6.64.2 Method Documentation

6.64.2.1 - (IBAction) deleteButton: (id) sender

delete account if user quits the game

Definition at line 55 of file ProfileViewController.m.

6.64.2.2 - (IBAction) logoutButton: (id) sender

logout current user and go to login page

Definition at line 49 of file ProfileViewController.m.

6.64.2.3 - (IBAction) saveChanges: (id) sender

Update User name.

Definition at line 63 of file ProfileViewController.m.

6.64.3 Property Documentation

6.64.3.1 - (IBOutlet UILabel*) emailLabel [read], [write], [nonatomic], [weak]

Definition at line 14 of file ProfileViewController.h.

6.64.3.2 - (IBOutlet UILabel*) highScoreLabel [read], [write], [nonatomic], [weak]

Definition at line 15 of file ProfileViewController.h.

6.64.3.3 - (IBOutlet UITextField*) userNameField [read], [write], [nonatomic], [weak]

Definition at line 19 of file ProfileViewController.h.

6.64.3.4 - (IBOutlet UILabel*) userNameLabel [read], [write], [nonatomic], [weak]

Definition at line 13 of file ProfileViewController.h.

The documentation for this class was generated from the following files:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Profile↔
ViewController.h
- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Profile↔
ViewController.m

6.65 ProfileViewController() Category Reference

This is the method file for [ProfileViewController.m](#). Implement codes for viewing account info.

6.65.1 Detailed Description

This is the method file for [ProfileViewController.m](#). Implement codes for viewing account info.

[ProfileViewController.m](#)

Author

JScott Richards

Copyright

2015 JScott Richards

Definition at line 13 of file ProfileViewController.m.

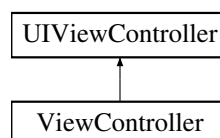
The documentation for this category was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Profile↔
ViewController.m

6.66 ViewController Class Reference

```
#import <ViewController.h>
```

Inheritance diagram for ViewController:



6.66.1 Detailed Description

Definition at line 12 of file ViewController.h.

The documentation for this class was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/View↔
Controller.h

6.67 ViewController() Category Reference

6.67.1 Detailed Description

Definition at line 11 of file ViewController.m.

The documentation for this category was generated from the following file:

- /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/[View↔Controller.m](#)

Chapter 7

File Documentation

7.1 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFAppLink.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFAppLink](#)

Variables

- FOUNDATION_EXPORT NSString* const [BFAppLinkVersion](#)

7.1.1 Variable Documentation

7.1.1.1 FOUNDATION_EXPORT NSString* const BFAppLinkVersion

The version of the App Link protocol that this library supports

Definition at line 14 of file BFAppLink.h.

7.2 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFAppLinkNavigation.h File Reference

```
#import <Foundation/Foundation.h>  
#import <Bolts/BFAppLink.h>
```

Classes

- class [BFAppLinkNavigation](#)

Functions

- typedef [NS_ENUM](#) (NSInteger, BFAppLinkNavigationType)

7.2.1 Function Documentation

7.2.1.1 typedef NS_ENUM (NSInteger , BFAppLinkNavigationType)

The result of calling navigate on a [BFAppLinkNavigation](#) Indicates that the navigation failed and no app was opened

Indicates that the navigation succeeded by opening the URL in the browser

Indicates that the navigation succeeded by opening the URL in an app on the device

Definition at line 18 of file BFAppLinkNavigation.h.

7.3 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↔ Bolts.framework/Headers/BFAppLinkResolving.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- protocol [<BFAppLinkResolving>](#)

7.4 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↔ Bolts.framework/Headers/BFAppLinkReturnToRefererController.h File Reference

```
#import <Foundation/Foundation.h>
#import <UIKit/UIKit.h>
#import <Bolts/BFAppLinkReturnToRefererView.h>
```

Classes

- protocol [<BFAppLinkReturnToRefererControllerDelegate>](#)
- class [BFAppLinkReturnToRefererController](#)

7.5 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↔ Bolts.framework/Headers/BFAppLinkReturnToRefererView.h File Reference

```
#import <Foundation/Foundation.h>
#import <UIKit/UIKit.h>
#import <Bolts/BFAppLinkNavigation.h>
```

Classes

- protocol [<BFAppLinkReturnToRefererViewDelegate>](#)
- class [BFAppLinkReturnToRefererView](#)

Typedefs

- typedef enum [BFIncludeStatusBarInSize](#) [BFIncludeStatusBarInSize](#)

Enumerations

- enum [BFIncludeStatusBarInSize](#) { [BFIncludeStatusBarInSizeNever](#), [BFIncludeStatusBarInSizeIOS7AndLater](#), [BFIncludeStatusBarInSizeAlways](#) }

7.5.1 Typedef Documentation

7.5.1.1 typedef enum [BFIncludeStatusBarInSize](#) [BFIncludeStatusBarInSize](#)

7.5.2 Enumeration Type Documentation

7.5.2.1 enum [BFIncludeStatusBarInSize](#)

Enumerator

[BFIncludeStatusBarInSizeNever](#)

[BFIncludeStatusBarInSizeIOS7AndLater](#)

[BFIncludeStatusBarInSizeAlways](#)

Definition at line 19 of file [BFAppLinkReturnToRefererView.h](#).

7.6 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFAppLinkTarget.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFAppLinkTarget](#)

7.7 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFExecutor.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFExecutor](#)

7.8 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFMeasurementEvent.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFMeasurementEvent](#)

Variables

- FOUNDATION_EXPORT NSString *const [BFMeasurementEventNotificationName](#)
- FOUNDATION_EXPORT NSString *const [BFMeasurementEventNameKey](#)
- FOUNDATION_EXPORT NSString *const [BFMeasurementEventArgsKey](#)
- FOUNDATION_EXPORT NSString *const [BFAppLinkParseEventName](#)
- FOUNDATION_EXPORT NSString *const [BFAppLinkNavigateInEventName](#)
- FOUNDATION_EXPORT NSString *const [BFAppLinkNavigateOutEventName](#)
- FOUNDATION_EXPORT NSString *const [BFAppLinkNavigateBackToReferrerEventName](#)

7.8.1 Variable Documentation

7.8.1.1 FOUNDATION_EXPORT NSString* const BFAppLinkNavigateBackToReferrerEventName

The event raised when the user navigates out from your app and back to the referrer app. e.g when the user leaves your app after tapping the back-to-referrer navigation bar

Definition at line 42 of file BFMeasurementEvent.h.

7.8.1.2 FOUNDATION_EXPORT NSString* const BFAppLinkNavigateInEventName

The name of the event posted when [[BFURL URLWithInboundURL:](#)] is called successfully. This represents parsing an inbound app link URL from a different application

Definition at line 33 of file BFMeasurementEvent.h.

7.8.1.3 FOUNDATION_EXPORT NSString* const BFAppLinkNavigateOutEventName

The event raised when the user navigates from your app to other apps

Definition at line 36 of file BFMeasurementEvent.h.

7.8.1.4 FOUNDATION_EXPORT NSString* const BFAppLinkParseEventName

Bolts Events raised by [BFMeasurementEvent](#) for Applink

The name of the event posted when [[BFURL URLWithURL:](#)] is called successfully. This represents the successful parsing of an app link URL.

Definition at line 27 of file BFMeasurementEvent.h.

7.8.1.5 FOUNDATION_EXPORT NSString* const BFMeasurementEventArgsKey

The dictionary field for the arguments of the event
Definition at line 20 of file BFMeasurementEvent.h.

7.8.1.6 FOUNDATION_EXPORT NSString* const BFMeasurementEventNameKey

Defines keys in the userInfo object for the notification named BFMeasurementEventNotificationName
The string field for the name of the event
Definition at line 18 of file BFMeasurementEvent.h.

7.8.1.7 FOUNDATION_EXPORT NSString* const BFMeasurementEventNotificationName

The name of the notification posted by [BFMeasurementEvent](#)
Definition at line 14 of file BFMeasurementEvent.h.

7.9 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BFTask.h File Reference ↩↪

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFTask](#)

Typedefs

- typedef id(^ [BFContinuationBlock](#)) ([BFTask](#) *task)

Variables

- NSString *const [BFTaskErrorDomain](#)
- NSString *const [BFTaskMultipleExceptionsException](#)

7.9.1 Typedef Documentation

7.9.1.1 typedef id(^ BFContinuationBlock) (BFTask *task)

A block that can act as a continuation for a task.
Definition at line 29 of file BFTask.h.

7.9.2 Variable Documentation

7.9.2.1 NSString* const BFTaskErrorDomain

Error domain used if there was multiple errors on [<BFTask taskForCompletionOfAllTasks:>](#).

7.9.2.2 NSString* const BFTaskMultipleExceptionsException

An exception that is thrown if there was multiple exceptions on `<BFTask taskForCompletionOfAllTasks:>`.

7.10 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFTaskCompletionSource.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFTaskCompletionSource](#)

7.11 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFURL.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [BFURL](#)

7.12 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/BFWebViewAppLinkResolver.h File Reference

```
#import <Foundation/Foundation.h>
#import <Bolts/BFAppLinkResolving.h>
```

Classes

- class [BFWebViewAppLinkResolver](#)

7.13 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ Bolts.framework/Headers/Bolts.h File Reference

```
#import <Bolts/BoltsVersion.h>
#import <Bolts/BFExecutor.h>
#import <Bolts/BFTask.h>
#import <Bolts/BFTaskCompletionSource.h>
```

Classes

- class [Bolts](#)

Variables

- NSInteger const [kBFMultipleErrorsError](#)

7.13.1 Variable Documentation

7.13.1.1 NSInteger const kBFMultipleErrorsError

80175001: There were multiple errors.

7.14 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bolts.framework/Headers/BoltsVersion.h File Reference ↩

Macros

- #define [BOLTS_VERSION](#) @"1.1.4"

7.14.1 Macro Definition Documentation

7.14.1.1 #define BOLTS_VERSION @"1.1.4"

Definition at line 1 of file BoltsVersion.h.

7.15 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/AppDelegate.h File Reference ↩

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

Classes

- class [AppDelegate](#)

7.16 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/AppDelegate.m File Reference ↩

```
#import "AppDelegate.h"
#import <Parse/Parse.h>
```

7.17 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Bound Breaker/Game.h File Reference ↩

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

Classes

- class [Game](#)

Variables

- bool [Right](#)
This is the header file for [Game.m](#). It contains methods and variable declarations.
- bool [Left](#)
- int [randomLongObstacle1Placement](#)
- int [randomLongObstacle2Placement](#)
- int [randomLongObstacle3Placement](#)
- int [randomLongObstacle4Placement](#)
- int [ScoreNumber](#)
- NSInteger [HighScoreNumber](#)

7.17.1 Variable Documentation

7.17.1.1 NSInteger HighScoreNumber

Definition at line 19 of file [Game.h](#).

7.17.1.2 bool Left

Definition at line 13 of file [Game.h](#).

7.17.1.3 int randomLongObstacle1Placement

Definition at line 14 of file [Game.h](#).

7.17.1.4 int randomLongObstacle2Placement

Definition at line 15 of file [Game.h](#).

7.17.1.5 int randomLongObstacle3Placement

Definition at line 16 of file [Game.h](#).

7.17.1.6 int randomLongObstacle4Placement

Definition at line 17 of file [Game.h](#).

7.17.1.7 bool Right

This is the header file for [Game.m](#). It contains methods and variable declarations.

[Game.h](#)

Author

Victor Palacios

Copyright

2015 Victor Palacios

Variables

Definition at line 12 of file Game.h.

7.17.1.8 int ScoreNumber

Definition at line 18 of file Game.h.

7.18 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ Bound Breaker/Game.m File Reference ↩

```
#import "Game.h"
#import <Parse/Parse.h>
```

Variables

- int `screenTop` = -30
- int `screenBottom` = 645
- int `screenRight` = 346
- int `screenLeft` = 20
- int `ballSpeed` = 3
- int `defaultBallYCoord` = 525
- int `recoverySpeed` = 1
- int `obstacleSpeed` = 1
- int `numberOfObstaclesOnScreen` = 2
- int `obstacleDistance` = 300
- int `obstacleGap` = 100
- int `longObstacleWidth` = 290
- int `obstacleGeneratorBuffer` = 10

7.18.1 Variable Documentation

7.18.1.1 int ballSpeed = 3

Variables that deal with the ball/player

Definition at line 25 of file Game.m.

7.18.1.2 int defaultBallYCoord = 525

Definition at line 26 of file Game.m.

7.18.1.3 int longObstacleWidth = 290

Definition at line 33 of file Game.m.

7.18.1.4 int numberOfObstaclesOnScreen = 2

Definition at line 30 of file Game.m.

7.18.1.5 int obstacleDistance = 300

Definition at line 31 of file Game.m.

7.18.1.6 int obstacleGap = 100

Definition at line 32 of file Game.m.

7.18.1.7 int obstacleGeneratorBuffer = 10

Variable needed for proper obstacle placement mechanism, used to narrow the obstacle regeneration area to a small portion of the screen

Definition at line 36 of file Game.m.

7.18.1.8 int obstacleSpeed = 1

Variables that deal with the obstacles

Definition at line 29 of file Game.m.

7.18.1.9 int recoverySpeed = 1

Definition at line 27 of file Game.m.

7.18.1.10 int screenBottom = 645

Definition at line 21 of file Game.m.

7.18.1.11 int screenLeft = 20

Definition at line 23 of file Game.m.

7.18.1.12 int screenRight = 346

Definition at line 22 of file Game.m.

7.18.1.13 int screenTop = -30

[Game](#) settings - change to obtain correct obstacle placement, gameover mechanism, etc.

Screen top declared as negative to have obstacles regenerate offscreen

Definition at line 20 of file Game.m.

7.19 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/leaderBoardViewController.h File Reference

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

Classes

- class [leaderBoardViewController](#)

7.20 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/leaderBoardViewController.m File Reference

```
#import "leaderBoardViewController.h"
#import <Parse/Parse.h>
```

7.21 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/LoginViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <Parse/Parse.h>
```

Classes

- class [LoginViewController](#)

7.22 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/LoginViewController.m File Reference

```
#import "LoginViewController.h"
```

7.23 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/main.m File Reference

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

Functions

- int [main](#) (int argc, char *argv[])

7.23.1 Function Documentation

7.23.1.1 `int main (int argc, char * argv[])`

Definition at line 12 of file main.m.

7.24 `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵` `Bound Breaker/ProfileViewController.h` File Reference

```
#import <UIKit/UIKit.h>
#import <Parse/Parse.h>
```

Classes

- class [ProfileViewController](#)

7.25 `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵` `Bound Breaker/ProfileViewController.m` File Reference

```
#import "ProfileViewController.h"
```

7.26 `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵` `Bound Breaker/ViewController.h` File Reference

```
#import <UIKit/UIKit.h>
#import <Parse/Parse.h>
```

Classes

- class [ViewController](#)

7.27 `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵` `Bound Breaker/ViewController.m` File Reference

```
#import "ViewController.h"
```

7.28 `/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵` `Bound BreakerTests/Bound_BreakerTests.m` File Reference

```
#import <UIKit/UIKit.h>
#import <XCTest/XCTest.h>
```


Classes

- class [Bound_BreakerTests](#)

7.29 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/Parse.h File Reference ↩

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFACL.h>
#import <ParseOSX/PFAnalytics.h>
#import <ParseOSX/PFAnonymousUtils.h>
#import <ParseOSX/PFCloud.h>
#import <ParseOSX/PFConfig.h>
#import <ParseOSX/PFConstants.h>
#import <ParseOSX/PFFile.h>
#import <ParseOSX/PFGeoPoint.h>
#import <ParseOSX/PFNullability.h>
#import <ParseOSX/PFObject+Subclass.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFQuery.h>
#import <ParseOSX/PFRelation.h>
#import <ParseOSX/PFRole.h>
#import <ParseOSX/PFSession.h>
#import <ParseOSX/PFSubclassing.h>
#import <ParseOSX/PFUser.h>
```

Classes

- class [Parse](#)

7.30 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PF_Twitter.h File Reference ↩

```
#import <Foundation/Foundation.h>
#import <Parse/PFNullability.h>
```

Classes

- class [PF_Twitter](#)

7.31 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFACL.h File Reference ↩

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
```

Classes

- class [PFACL](#)

7.32 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵ Parse.framework/Headers/PFAnalytics.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
```

Classes

- class [PFAnalytics](#)

7.33 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵ Parse.framework/Headers/PFAnonymousUtils.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
#import <ParseOSX/PFUser.h>
```

Classes

- class [PFAnonymousUtils](#)

7.34 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵ Parse.framework/Headers/PFCloud.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
```

Classes

- class [PFCloud](#)

7.35 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↵ Parse.framework/Headers/PFConfig.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
```

Classes

- class [PFConfig](#)

Typedefs

- typedef void(^ [PFConfigResultBlock](#)) (PFConfig *[PF_NULLABLE_S](#) config, NSError *[PF_NULLABLE_S](#) error)

7.35.1 Typedef Documentation

7.35.1.1 typedef void(^ [PFConfigResultBlock](#)) (PFConfig *[PF_NULLABLE_S](#) config, NSError *[PF_NULLABLE_S](#) error)

Definition at line 20 of file [PFConfig.h](#).

7.36 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFConstants.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
#import <Cocoa/Cocoa.h>
```

Macros

Deprecated Macros

- #define [PARSE_DEPRECATED](#)(_MSG)

Typedefs

Blocks

- typedef void(^ [PFBooleanResultBlock](#)) (BOOL succeeded, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFIntegerResultBlock](#)) (int number, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFArrayResultBlock](#)) (NSArray *[PF_NULLABLE_S](#) objects, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFObjectResultBlock](#)) (PFOBJECT *[PF_NULLABLE_S](#) object, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFSetResultBlock](#)) (NSSet *[PF_NULLABLE_S](#) channels, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFUserResultBlock](#)) (PFUser *[PF_NULLABLE_S](#) user, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFDataResultBlock](#)) (NSData *[PF_NULLABLE_S](#) data, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFDataStreamResultBlock](#)) (NSInputStream *[PF_NULLABLE_S](#) stream, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFStringResultBlock](#)) (NSString *[PF_NULLABLE_S](#) string, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFIdResultBlock](#)) ([PF_NULLABLE_S](#) id object, NSError *[PF_NULLABLE_S](#) error)
- typedef void(^ [PFProgressBlock](#)) (int percentDone)

Functions

Cache Policies

- typedef [NS_ENUM](#) (uint8_t, PFCachePolicy)

Logging Levels

- typedef [NS_ENUM](#) (uint8_t, PFLogLevel)

Variables

Server

- NSString *const [PF_NONNULL_S](#) kPFParseServer

Version

- #define [PARSE_VERSION](#) @"1.7.1"
- NSInteger const [PARSE_API_VERSION](#)

Platform

- #define [PARSE_IOS_ONLY](#) (TARGET_OS_IPHONE)
- #define [PARSE_OSX_ONLY](#) (TARGET_OS_MAC && !(TARGET_OS_IPHONE))
- NSString *const [PF_NONNULL_S](#) kPFDeviceType
- compatibility_alias UIImage [NSImage](#)
- compatibility_alias UIColor [NSColor](#)
- compatibility_alias UIView [NSView](#)

Errors

- NSString *const [PF_NONNULL_S](#) kPFParseErrorDomain
- typedef [NS_ENUM](#) (NSInteger, PFErrorCode)

7.36.1 Macro Definition Documentation

7.36.1.1 #define PARSE_DEPRECATED(_MSG)

Definition at line 400 of file PFConstants.h.

7.36.1.2 #define PARSE_IOS_ONLY (TARGET_OS_IPHONE)

Definition at line 27 of file PFConstants.h.

7.36.1.3 #define PARSE_OSX_ONLY (TARGET_OS_MAC && !(TARGET_OS_IPHONE))

Definition at line 28 of file PFConstants.h.

7.36.1.4 #define PARSE_VERSION @"1.7.1"

Definition at line 19 of file PFConstants.h.

7.36.2 Typedef Documentation

7.36.2.1 `typedef void(^ PFArrayResultBlock) (NSArray *PF_NULLABLE_S objects, NSError *PF_NULLABLE_S error)`

Definition at line 379 of file PFConstants.h.

7.36.2.2 `typedef void(^ PFBooleanResultBlock) (BOOL succeeded, NSError *PF_NULLABLE_S error)`

Definition at line 377 of file PFConstants.h.

7.36.2.3 `typedef void(^ PFDataResultBlock) (NSData *PF_NULLABLE_S data, NSError *PF_NULLABLE_S error)`

Definition at line 383 of file PFConstants.h.

7.36.2.4 `typedef void(^ PFDataStreamResultBlock) (NSInputStream *PF_NULLABLE_S stream, NSError *PF_NULLABLE_S error)`

Definition at line 384 of file PFConstants.h.

7.36.2.5 `typedef void(^ PFIdResultBlock) (PF_NULLABLE_S id object, NSError *PF_NULLABLE_S error)`

Definition at line 386 of file PFConstants.h.

7.36.2.6 `typedef void(^ PFIntegerResultBlock) (int number, NSError *PF_NULLABLE_S error)`

Definition at line 378 of file PFConstants.h.

7.36.2.7 `typedef void(^ PFObjectResultBlock) (PFObject *PF_NULLABLE_S object, NSError *PF_NULLABLE_S error)`

Definition at line 380 of file PFConstants.h.

7.36.2.8 `typedef void(^ PFProgressBlock) (int percentDone)`

Definition at line 387 of file PFConstants.h.

7.36.2.9 `typedef void(^ PFSetResultBlock) (NSSet *PF_NULLABLE_S channels, NSError *PF_NULLABLE_S error)`

Definition at line 381 of file PFConstants.h.

7.36.2.10 `typedef void(^ PFStringResultBlock) (NSString *PF_NULLABLE_S string, NSError *PF_NULLABLE_S error)`

Definition at line 385 of file PFConstants.h.

7.36.2.11 `typedef void(^ PFUserResultBlock) (PFUser *PF_NULLABLE_S user, NSError *PF_NULLABLE_S error)`

Definition at line 382 of file PFConstants.h.

7.36.3 Function Documentation

7.36.3.1 `typedef NS_ENUM (uint8_t, PFCachePolicy)`

`PFCachePolicy` specifies different caching policies that could be used with `<PFQuery>`.

This lets you show data when the user's device is offline, or when the app has just started and network requests have not yet had time to complete. [Parse](#) takes care of automatically flushing the cache when it takes up too much space.

Warning

Cache policy could only be set when Local Datastore is not enabled.

See also

[PFQuery](#)

The query does not load from the cache or save results to the cache. This is the default cache policy.

The query only loads from the cache, ignoring the network. If there are no cached results, this causes a `NSError` with `kPFCacheMiss` code.

The query does not load from the cache, but it will save results to the cache.

The query first tries to load from the cache, but if that fails, it loads results from the network. If there are no cached results, this causes a `NSError` with `kPFCacheMiss` code.

The query first tries to load from the network, but if that fails, it loads results from the cache. If there are no cached results, this causes a `NSError` with `kPFCacheMiss` code.

The query first loads from the cache, then loads from the network. The callback will be called twice - first with the cached results, then with the network results. Since it returns two results at different times, this cache policy cannot be used with synchronous or task methods.

Definition at line 62 of file `PFConstants.h`.

7.36.3.2 `typedef NS_ENUM (uint8_t, PFLogLevel)`

`PFLogLevel` enum specifies different levels of logging that could be used to limit or display more messages in logs.

See also

[\[Parse setLogLevel:\]](#)

[\[Parse logLevel\]](#)

Log level that disables all logging.

Log level that if set is going to output error messages to the log.

Log level that if set is going to output the following messages to log:

- Errors
- Warnings

Log level that if set is going to output the following messages to log:

- Errors
- Warnings

- Informational messages

Log level that if set is going to output the following messages to log:

- Errors
- Warnings
- Informational messages
- Debug messages

Definition at line 105 of file PFConstants.h.

7.36.3.3 typedef NS_ENUM (NSInteger , PFCErrorCode)

`PFCErrorCode` enum contains all custom error codes that are used as `code` for `NSError` for callbacks on all classes.

These codes are used when `domain` of `NSError` that you receive is set to `PFParseErrorDomain`. Internal server error. No information available.

The connection to the [Parse](#) servers failed.

Object doesn't exist, or has an incorrect password.

You tried to find values matching a datatype that doesn't support exact database matching, like an array or a dictionary.

Missing or invalid classname. Classnames are case-sensitive. They must start with a letter, and `a-zA-Z0-9_` are the only valid characters.

Missing object id.

Invalid key name. Keys are case-sensitive. They must start with a letter, and `a-zA-Z0-9_` are the only valid characters.

Malformed pointer. Pointers must be arrays of a classname and an object id.

Malformed json object. A json dictionary is expected.

Tried to access a feature only available internally.

Field set to incorrect type.

Invalid channel name. A channel name is either an empty string (the broadcast channel) or contains only `a-zA-Z0-9_` characters and starts with a letter.

Invalid device token.

Push is misconfigured. See details to find out how.

The object is too large.

That operation isn't allowed for clients.

The results were not found in the cache.

Keys in `NSDictionary` values may not include `$` or `..`

Invalid file name. A file name can contain only `a-zA-Z0-9_.` characters and should be between 1 and 36 characters.

Invalid ACL. An ACL with an invalid format was saved. This should not happen if you use `<PFACL>`.

The request timed out on the server. Typically this indicates the request is too expensive.

The email address was invalid.

A unique field was given a value that is already taken.

Role's name is invalid.

Exceeded an application quota. Upgrade to resolve.

Cloud Code script had an error.

Cloud Code validation failed.

Product purchase receipt is missing.

Product purchase receipt is invalid.

Payment is disabled on this device.

The product identifier is invalid.

The product is not found in the App Store.

The Apple server response is not valid.

Product fails to download due to file system error.

Fail to convert data to image.

Unsaved file.

Fail to delete file.

Application has exceeded its request limit.

Invalid event name.

Username is missing or empty.

Password is missing or empty.

Username has already been taken.

Email has already been taken.

The email is missing, and must be specified.

A user with the specified email was not found.

The user cannot be altered by a client without the session.

Users can only be created through sign up.

An existing Facebook account already linked to another user.

An existing account already linked to another user.

Error code indicating that the current session token is invalid.

Facebook id missing from request.

Linked id missing from request.

Invalid Facebook session.

Invalid linked session.

Definition at line 148 of file PFConstants.h.

7.36.4 Variable Documentation

7.36.4.1 `NSString* const PF_NONNULL_S kPFDeviceType`

7.36.4.2 `NSString* const PF_NONNULL_S kPFParseServer`

7.36.4.3 `compatibility_alias UIColor NSColor`

Definition at line 37 of file PFConstants.h.

7.36.4.4 compatibility_alias UIImage UIImage

Definition at line 36 of file PFConstants.h.

7.36.4.5 compatibility_alias UIView UIView

Definition at line 38 of file PFConstants.h.

7.36.4.6 NSInteger const PARSE_API_VERSION

7.36.4.7 NSString* const PF_NONNULL_S PFParseErrorDomain

7.37 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFFile.h File Reference ↩

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
```

Classes

- class [PFFile](#)

7.38 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFGeoPoint.h File Reference ↩

```
#import <Foundation/Foundation.h>
#import <CoreLocation/CoreLocation.h>
#import <ParseOSX/PFNullability.h>
```

Classes

- class [PFGeoPoint](#)

Typedefs

- typedef void(^ [PFGeoPointResultBlock](#)) (PFGeoPoint *[PF_NULLABLE_S](#) geoPoint, NSError *[PF_NULLABLE_S](#) error) ↩

7.38.1 Typedef Documentation

7.38.1.1 typedef void(^ [PFGeoPointResultBlock](#)) (PFGeoPoint *[PF_NULLABLE_S](#) geoPoint, NSError *[PF_NULLABLE_S](#) error)

Definition at line 20 of file PFGeoPoint.h.

7.39 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFInstallation.h](#) File Reference

```
#import <Foundation/Foundation.h>
#import <Parse/PFNullability.h>
#import <Parse/PFObject.h>
#import <Parse/PFSubclassing.h>
```

Classes

- class [PFInstallation](#)

7.40 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFNetworkActivityIndicatorManager.h](#) File Reference

```
#import <Foundation/Foundation.h>
#import <Parse/PFNullability.h>
```

Classes

- class [PFNetworkActivityIndicatorManager](#)

7.41 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFNullability.h](#) File Reference

Macros

Nullability Annotation Support

- #define [PF_NONNULL](#)
- #define [PF_NONNULL_S](#)
- #define [PF_NULLABLE](#)
- #define [PF_NULLABLE_S](#)
- #define [PF_NULLABLE_PROPERTY](#)
- #define [PF_ASSUME_NONNULL_BEGIN](#)
- #define [PF_ASSUME_NONNULL_END](#)

7.41.1 Macro Definition Documentation

7.41.1.1 #define [PF_ASSUME_NONNULL_BEGIN](#)

Definition at line 40 of file [PFNullability.h](#).

7.41.1.2 #define [PF_ASSUME_NONNULL_END](#)

Definition at line 41 of file [PFNullability.h](#).

7.41.1.3 #define PF_NONNULL

Definition at line 21 of file PFNullability.h.

7.41.1.4 #define PF_NONNULL_S

Definition at line 22 of file PFNullability.h.

7.41.1.5 #define PF_NULLABLE

Definition at line 23 of file PFNullability.h.

7.41.1.6 #define PF_NULLABLE_PROPERTY

Definition at line 25 of file PFNullability.h.

7.41.1.7 #define PF_NULLABLE_S

Definition at line 24 of file PFNullability.h.

7.42 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFObject+Subclass.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
#import <ParseOSX/PFObject.h>
```

Classes

- category [PFObject\(Subclass\)](#)

7.43 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFObject.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFACL.h>
#import <ParseOSX/PFConstants.h>
```

Classes

- class [PFObject](#)

Variables

- NSString *const [PFObjectDefaultPin](#)

7.43.1 Variable Documentation

7.43.1.1 NSString* const PFObjDefaultPin

The name of the default pin that for [PFObj](#) local data store.

7.44 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFProduct.h](#) File Reference

```
#import <Parse/PFFile.h>
#import <Parse/PFNullability.h>
#import <Parse/PFObject.h>
#import <Parse/PFSubclassing.h>
```

Classes

- class [PFProduct](#)

7.45 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFPurchase.h](#) File Reference

```
#import <Foundation/Foundation.h>
#import <UIKit/UIKit.h>
#import <Parse/PFConstants.h>
```

Classes

- class [PFPurchase](#)

7.46 [/Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFPush.h](#) File Reference

```
#import <Foundation/Foundation.h>
#import <Parse/PFConstants.h>
```

Classes

- class [PFPush](#)

7.47 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFQuery.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
#import <ParseOSX/PFGeoPoint.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFUser.h>
```

Classes

- class [PFQuery](#)

7.48 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFRelation.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFQuery.h>
```

Classes

- class [PFRelation](#)

7.49 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFRole.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFSubclassing.h>
```

Classes

- class [PFRole](#)

7.50 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFSession.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFSubclassing.h>
```

Classes

- class [PFSession](#)

Typedefs

- typedef void(^ [PFSessionResultBlock](#)) (PFSession *[PF_NULLABLE_S](#) session, NSError *[PF_NULLABLE_S](#) error)

7.50.1 Typedef Documentation

- 7.50.1.1 typedef void(^ [PFSessionResultBlock](#)) (PFSession *[PF_NULLABLE_S](#) session, NSError *[PF_NULLABLE_S](#) error)

Definition at line 21 of file PFSession.h.

7.51 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFSubclassing.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFNullability.h>
```

Classes

- protocol [PFSubclassing](#)

7.52 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFTwitterUtils.h File Reference

```
#import <Foundation/Foundation.h>
#import <Parse/PFConstants.h>
#import <Parse/PFUser.h>
#import <Parse/PF_Twitter.h>
```

Classes

- class [PFTwitterUtils](#)

7.53 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/Parse.framework/Headers/PFUser.h File Reference

```
#import <Foundation/Foundation.h>
#import <ParseOSX/PFConstants.h>
#import <ParseOSX/PFObject.h>
#import <ParseOSX/PFSubclassing.h>
```

Classes

- class [PFUser](#)

Typedefs

- typedef void(^ [PFUserLogoutResultBlock](#)) (NSError *[PF_NULLABLE_S](#) error)

Variables

- [PF_ASSUME_NONNULL_BEGIN](#) typedef void(^ [PFUserSessionUpgradeResultBlock](#)) (NSError *[PF_NULLABLE_S](#) error)

7.53.1 Typedef Documentation

7.53.1.1 typedef void(^ [PFUserLogoutResultBlock](#)) (NSError *[PF_NULLABLE_S](#) error)

Definition at line 22 of file [PFUser.h](#).

7.53.2 Variable Documentation

7.53.2.1 [PF_ASSUME_NONNULL_BEGIN](#) typedef void(^ [PFUserSessionUpgradeResultBlock](#)) (NSError *[PF_NULLABLE_S](#) error)

7.54 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseCrashReporting.framework/Headers/ParseCrashReporting.h File Reference

```
#import <Foundation/Foundation.h>
```

Classes

- class [ParseCrashReporting](#)

7.55 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseFacebookUtils.framework/Headers/PFFacebookUtils.h File Reference

```
#import <Foundation/Foundation.h>
#import <FacebookSDK/FacebookSDK.h>
#import <Parse/PFConstants.h>
#import <Parse/PFNullability.h>
#import <Parse/PFUser.h>
```

Classes

- class [PFFacebookUtils](#)

7.56 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseFacebookUtilsV4.framework/Headers/PFFacebookUtils.h File Reference

```
#import <Foundation/Foundation.h>
#import <Bolts/BFTask.h>
#import <FBSDKCoreKit/FBSDKAccessToken.h>
#import <FBSDKLoginKit/FBSDKLoginManager.h>
#import <Parse/PFConstants.h>
#import <Parse/PFNullability.h>
#import <Parse/PFUser.h>
```

Classes

- class [PFFacebookUtils](#)

7.57 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/ParseUI.h File Reference

```
#import <ParseUI/PFCollectionViewCell.h>
#import <ParseUI/PFImageView.h>
#import <ParseUI/PFLogInView.h>
#import <ParseUI/PFLogInViewController.h>
#import <ParseUI/PFProductTableViewController.h>
#import <ParseUI/PFPurchaseTableViewCell.h>
#import <ParseUI/PFQueryCollectionViewController.h>
#import <ParseUI/PFQueryTableViewController.h>
#import <ParseUI/PFSignUpView.h>
#import <ParseUI/PFSignUpViewController.h>
#import <ParseUI/PFTableViewCell.h>
#import <ParseUI/PFTextField.h>
#import <ParseUI/ParseUIConstants.h>
```

7.58 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/ParseUIConstants.h File Reference

```
#import <Availability.h>
#import <TargetConditionals.h>
```

Macros

Deprecated Macros

- #define [PARSE_UI_DEPRECATED\(_MSG\)](#)

Nullability Support

- #define [PFUI_NULLABLE](#)
- #define [PFUI_NULLABLE_S](#)
- #define [PFUI_NULL_UNSPECIFIED](#)

- `#define PFUI_NULLABLE_PROPERTY`
- `#define PFUI_ASSUME_NONNULL_BEGIN`
- `#define PFUI_ASSUME_NONNULL_END`

7.58.1 Macro Definition Documentation

7.58.1.1 `#define PARSE_UI_DEPRECATED(_MSG)`

Definition at line 39 of file ParseUIConstants.h.

7.58.1.2 `#define PFUI_ASSUME_NONNULL_BEGIN`

Definition at line 72 of file ParseUIConstants.h.

7.58.1.3 `#define PFUI_ASSUME_NONNULL_END`

Definition at line 73 of file ParseUIConstants.h.

7.58.1.4 `#define PFUI_NULL_UNSPECIFIED`

Definition at line 56 of file ParseUIConstants.h.

7.58.1.5 `#define PFUI_NULLABLE`

Definition at line 54 of file ParseUIConstants.h.

7.58.1.6 `#define PFUI_NULLABLE_PROPERTY`

Definition at line 57 of file ParseUIConstants.h.

7.58.1.7 `#define PFUI_NULLABLE_S`

Definition at line 55 of file ParseUIConstants.h.

7.59 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFCollectionViewCell.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFCollectionViewCell](#)

7.60 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFImageView.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFImageView](#)

Variables

- [PFUI_ASSUME_NONNULL_BEGIN](#) typedef void(^ [PFImageViewImageResultBlock](#))(UIImage *[PFUI_NONNULLABLE_S](#) image, NSError *[PFUI_NONNULLABLE_S](#) error)

7.60.1 Variable Documentation

7.60.1.1 [PFUI_ASSUME_NONNULL_BEGIN](#) typedef void(^ [PFImageViewImageResultBlock](#))(UIImage *[PFUI_NONNULLABLE_S](#) image, NSError *[PFUI_NONNULLABLE_S](#) error)

7.61 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFLoginView.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFLoginView](#)

Functions

- [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (NSInteger, PFLoginFields)

7.61.1 Function Documentation

7.61.1.1 [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (NSInteger , PFLoginFields)

[PFLoginFields](#) bitmask specifies the log in elements which are enabled in the view.

See also

[PFLoginViewController](#)
[PFLoginView](#)

No fields.

Username and password fields.

Forgot password button.

Login button.

Button to login with Facebook.

Button to login with Twitter.

Signup Button.

Dismiss Button.

Default value. Combines Username, Password, Login, Signup, Forgot Password and Dismiss buttons.

Definition at line 34 of file PFLoginView.h.

7.62 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFLoginViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
#import <ParseUI/PFLoginView.h>
```

Classes

- class [PFLoginViewController](#)
- protocol [<PFLoginViewControllerDelegate>](#)

Variables

Notifications

- NSString *const [PFLoginSuccessNotification](#)
- NSString *const [PFLoginFailureNotification](#)
- NSString *const [PFLoginCancelNotification](#)

7.62.1 Variable Documentation

7.62.1.1 NSString* const PFLoginCancelNotification

The notification is posted immediately after the log in is cancelled.

7.62.1.2 NSString* const PFLoginFailureNotification

The notification is posted immediately after the log in fails. If the delegate prevents the log in from starting, the notification is not sent.

7.62.1.3 NSString* const PFLoginSuccessNotification

The notification is posted immediately after the log in succeeds.

7.63 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ ParseUI.framework/Headers/PFProductTableViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
#import <ParseUI/PFQueryTableViewController.h>
```

Classes

- class [PFProductTableViewController](#)

7.64 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ ParseUI.framework/Headers/PFPurchaseTableViewCell.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
#import <ParseUI/PFTableViewCell.h>
```

Classes

- class [PFPurchaseTableViewCell](#)

Functions

- [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_ENUM](#) (uint8_t, PFPurchaseTableViewCellState)

7.64.1 Function Documentation

7.64.1.1 [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_ENUM](#) (uint8_t , PFPurchaseTableViewCellState)

An enum that represents states of the [PFPurchaseTableViewCell](#).

See also

[PFPurchaseTableViewCell](#)

Normal state of the cell.

Downloading state of the cell.

State of the cell, when the product was downloaded.

Definition at line 33 of file [PFPurchaseTableViewCell.h](#).

7.65 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/↩ ParseUI.framework/Headers/PFQueryCollectionViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFQueryCollectionViewController](#)

7.66 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFQueryTableViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFQueryTableViewController](#)

7.67 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFSignUpView.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFSignUpView](#)

Functions

- [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (NSInteger, PFSignUpFields)

7.67.1 Function Documentation

7.67.1.1 [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (NSInteger , PFSignUpFields)

[PFSignUpFields](#) bitmask specifies the sign up elements which are enabled in the view.

See also

[PFSignUpViewController](#)
[PFSignUpView](#)

Username and password fields.

Email field.

This field can be used for something else.

Sign Up Button

Dismiss Button

Default value. Combines Username, Password, Email, Sign Up and Dismiss Buttons.

Definition at line 34 of file [PFSignUpView.h](#).

7.68 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFSignUpViewController.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
#import <ParseUI/PFSignUpView.h>
```

Classes

- class [PFSignUpViewController](#)
- protocol [<PFSignUpViewControllerDelegate>](#)

Variables

Notifications

- NSString *const [PFSignUpSuccessNotification](#)
- NSString *const [PFSignUpFailureNotification](#)
- NSString *const [PFSignUpCancelNotification](#)

7.68.1 Variable Documentation

7.68.1.1 NSString* const PFSignUpCancelNotification

The notification is posted immediately after the user cancels sign up.

7.68.1.2 NSString* const PFSignUpFailureNotification

The notification is posted immediately after the sign up fails.

If the delegate prevents the sign up to start, the notification is not sent.

7.68.1.3 NSString* const PFSignUpSuccessNotification

The notification is posted immediately after the sign up succeeds.

7.69 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFTableViewCell.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
#import <ParseUI/PFImageView.h>
```

Classes

- class [PFTableViewCell](#)

7.70 /Users/jscottrichards/Documents/GitHub/BoundBreaker/Project/Bound Breaker/ParseUI.framework/Headers/PFTextField.h File Reference

```
#import <UIKit/UIKit.h>
#import <ParseUI/ParseUIConstants.h>
```

Classes

- class [PFTextField](#)

Functions

- [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (uint8_t, PFTextFieldSeparatorStyle)

7.70.1 Function Documentation

7.70.1.1 [PFUI_ASSUME_NONNULL_BEGIN](#) typedef [NS_OPTIONS](#) (uint8_t , PFTextFieldSeparatorStyle)

[PFTextFieldSeparatorStyle](#) bitmask specifies the style of the separators, that should be used for a given [PFTextField](#).

See also

[PFTextField](#)

No separators are visible.

Separator on top of the text field.

Separator at the bottom of the text field.

Definition at line 34 of file [PFTextField.h](#).

7.71 /Users/jscottrichards/Documents/GitHub/BoundBreaker/README.md File Reference

Index

- <BFAppLinkResolving>, 19
- <BFAppLinkReturnToRefererControllerDelegate>, 22
- <BFAppLinkReturnToRefererViewDelegate>, 24
- <PFLoginViewControllerDelegate>, 101
- <PFSignUpViewControllerDelegate>, 196
- <PFSubclassing>, 197
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLink.h, 219
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLinkNavigation.h, 219
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLinkResolving.h, 220
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLinkReturnTo↔
 - RefererController.h, 220
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLinkReturnTo↔
 - RefererView.h, 220
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFAppLinkTarget.h, 221
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFExecutor.h, 221
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFMeasurementEvent.h, 222
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFTask.h, 223
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFTaskCompletion↔
 - Source.h, 224
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFURL.h, 224
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BFWebViewAppLink↔
 - Resolver.h, 224
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/Bolts.h, 224
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bolts.↔
 - framework/Headers/BoltsVersion.h, 225
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/AppDelegate.h, 225
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/AppDelegate.m, 225
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/Game.h, 225
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/Game.m, 227
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/LoginViewController.h, 229
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/LoginViewController.m, 229
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/ProfileViewController.h, 230
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/ProfileViewController.m, 230
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/ViewController.h, 230
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/ViewController.m, 230
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/leaderBoardViewController.h, 229
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/leaderBoardViewController.m, 229
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - Breaker/main.m, 229
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Bound
 - BreakerTests/Bound_BreakerTests.m, 230

- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFACL.h, 231
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFAnalytics.h, 232
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFAnonymousUtils.h,
232
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFCloud.h, 232
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFConfig.h, 232
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFConstants.h, 233
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFFile.h, 239
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFGeoPoint.h, 239
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFInstallation.h, 240
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFNetworkActivity↵
IndicatorManager.h, 240
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFNullability.h, 240
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFObject+Subclass.h,
241
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFObject.h, 241
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFProduct.h, 242
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFPurchase.h, 242
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFPush.h, 242
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFQuery.h, 243
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFRelation.h, 243
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFRole.h, 243
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFSession.h, 243
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFSubclassing.h, 244
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFTwitterUtils.h, 244
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PFUser.h, 244
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/PF_Twitter.h, 231
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
framework/Headers/Parse.h, 231
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseCrash↵
Reporting.framework/Headers/ParseCrash↵
Reporting.h, 245
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
FacebookUtils.framework/Headers/PF↵
FacebookUtils.h, 245
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
FacebookUtilsV4.framework/Headers/PF↵
FacebookUtils.h, 246
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseUI.↵
framework/Headers/PFCollectionViewCell.h,
247
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseUI.↵
framework/Headers/PFImageView.h, 248
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseUI.↵
framework/Headers/PFLoginView.h, 248
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/Parse.↵
UI.framework/Headers/PFLoginView↵
Controller.h, 249
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseUI.↵
framework/Headers/PFProductTableView↵
Controller.h, 250
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseUI.↵
framework/Headers/PFPurchaseTableView↵
Cell.h, 250
- /Users/jscottrichards/Documents/GitHub/Bound↵
Breaker/Project/Bound Breaker/ParseU↵
I.framework/Headers/PFQueryCollection↵

- ViewController.h, [250](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - l.framework/Headers/PFQueryTableView↔
 - Controller.h, [251](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - framework/Headers/PFSignUpView.h, [251](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/Parse↔
 - UI.framework/Headers/PFSignUpView↔
 - Controller.h, [252](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - framework/Headers/PFTableViewCell.h, [252](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - framework/Headers/PFTextField.h, [253](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - framework/Headers/ParseUI.h, [246](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/Project/Bound Breaker/ParseUI↔
 - framework/Headers/ParseUIConstants.h, [246](#)
- /Users/jscottrichards/Documents/GitHub/Bound↔
 - Breaker/README.md, [253](#)
- __attribute__
 - PFLogInView, [97](#)
- ACL
 - PFACL, [54](#)
 - PFObjct, [138](#)
- ACLWithUser:
 - PFACL, [54](#)
- addAscendingOrder:
 - PFQuery, [159](#)
- addDescendingOrder:
 - PFQuery, [159](#)
- addObject:
 - PFRelation, [188](#)
- addObject:forKey:
 - PFObjct, [108](#)
- addObjectsFromArray:forKey:
 - PFObjct, [108](#)
- addObserverForProduct:block:
 - PFPurchase, [143](#)
- addUniqueObject:forKey:
 - PFObjct, [108](#)
- addUniqueObjectsFromArray:forKey:
 - PFObjct, [108](#)
- additionalField
 - PFSignUpView, [193](#)
- allKeys
 - PFObjct, [108](#)
- AppDelegate, [15](#)
 - window, [15](#)
- AppDelegate(), [15](#)
- appLink
 - BFApplinkNavigation, [19](#)
- appLinkData
 - BFApplinkNavigation, [19](#)
 - BFURL, [36](#)
- appLinkExtras
 - BFURL, [36](#)
- appLinkFromURLInBackground:
 - BFApplinkResolving-p, [20](#)
- appLinkReferer
 - BFURL, [36](#)
- appLinkTargetWithURL:appStoreId:appName:
 - BFApplinkTarget, [26](#)
- appLinkWithSourceURL:targets:webURL:
 - BFApplink, [16](#)
- appName
 - BFApplinkTarget, [26](#)
- appStoreId
 - BFApplinkTarget, [26](#)
- assetContentPathForProduct:
 - PFPurchase, [143](#)
- authToken
 - PF_Twitter, [52](#)
- authTokenSecret
 - PF_Twitter, [52](#)
- authorizeInBackground
 - PF_Twitter, [52](#)
- authorizeWithSuccess:failure:cancel:
 - PF_Twitter, [52](#)
- BFApplink, [16](#)
 - appLinkWithSourceURL:targets:webURL:, [16](#)
 - sourceURL, [16](#)
 - targets, [16](#)
 - webURL, [17](#)
- BFApplink.h
 - BFApplinkVersion, [219](#)
- BFApplinkNavigateBackToReferrerEventName
 - BFMeasurementEvent.h, [222](#)
- BFApplinkNavigateInEventName
 - BFMeasurementEvent.h, [222](#)
- BFApplinkNavigateOutEventName
 - BFMeasurementEvent.h, [222](#)
- BFApplinkNavigation, [17](#)
 - appLink, [19](#)
 - appLinkData, [19](#)
 - defaultResolver, [18](#)
 - extras, [19](#)
 - navigate:, [18](#)
 - navigateToAppLink:error:, [18](#)
 - navigateToURLInBackground:, [18](#)
 - navigateToURLInBackground:resolver:, [18](#)
 - navigationWithAppLink:extras:appLinkData:, [18](#)
 - resolveAppLinkInBackground:, [18](#)
 - resolveAppLinkInBackground:resolver:, [18](#)
 - setDefaultResolver:, [18](#)
- BFApplinkNavigation.h
 - NS_ENUM, [220](#)
- BFApplinkParseEventName
 - BFMeasurementEvent.h, [222](#)
- BFApplinkResolving-p

- appLinkFromURLInBackground:, 20
- BFAppLinkReturnToRefererController, 21
 - closeViewAnimated:, 21
 - delegate, 22
 - init, 21
 - initWithDisplayAboveNavController:, 22
 - removeFromNavController, 22
 - showViewForRefererAppLink:, 22
 - showViewForRefererURL:, 22
 - view, 22
- BFAppLinkReturnToRefererControllerDelegate-p
 - returnToRefererController:didNavigateToApp↔
 - Link:type:, 23
 - returnToRefererController:willNavigateToApp↔
 - Link:, 23
- BFAppLinkReturnToRefererView, 23
 - closed, 24
 - delegate, 24
 - includeStatusBarInSize, 24
 - refererAppLink, 24
 - textColor, 24
- BFAppLinkReturnToRefererView.h
 - BFIncludeStatusBarInSize, 221
 - BFIncludeStatusBarInSizeAlways, 221
 - BFIncludeStatusBarInSizeIOS7AndLater, 221
 - BFIncludeStatusBarInSizeNever, 221
- BFAppLinkReturnToRefererViewDelegate-p
 - returnToRefererViewDidTapInsideCloseButton:, 25
 - returnToRefererViewDidTapInsideLink:link:, 25
- BFAppLinkTarget, 25
 - appLinkTargetWithURL:appStoreId:appName:, 26
 - appName, 26
 - appStoreId, 26
 - URL, 26
- BFAppLinkVersion
 - BFAppLink.h, 219
- BFContinuationBlock
 - BFTask.h, 223
- BFExecutor, 26
 - defaultExecutor, 27
 - execute:, 27
 - executorWithBlock:, 27
 - executorWithDispatchQueue:, 27
 - executorWithOperationQueue:, 28
 - immediateExecutor, 28
 - mainThreadExecutor, 28
- BFIncludeStatusBarInSize
 - BFAppLinkReturnToRefererView.h, 221
- BFIncludeStatusBarInSizeAlways
 - BFAppLinkReturnToRefererView.h, 221
- BFIncludeStatusBarInSizeIOS7AndLater
 - BFAppLinkReturnToRefererView.h, 221
- BFIncludeStatusBarInSizeNever
 - BFAppLinkReturnToRefererView.h, 221
- BFMeasurementEvent, 28
- BFMeasurementEvent.h
 - BFAppLinkNavigateBackToReferrerEventName, 222
 - BFAppLinkNavigateInEventName, 222
 - BFAppLinkNavigateOutEventName, 222
 - BFAppLinkParseEventName, 222
 - BFMeasurementEventArgsKey, 222
 - BFMeasurementEventNameKey, 223
 - BFMeasurementEventNotificationName, 223
- BFMeasurementEventArgsKey
 - BFMeasurementEvent.h, 222
- BFMeasurementEventNameKey
 - BFMeasurementEvent.h, 223
- BFMeasurementEventNotificationName
 - BFMeasurementEvent.h, 223
- BFTask, 28
 - cancelled, 32
 - cancelledTask, 29
 - completed, 32
 - continueWithBlock:, 29
 - continueWithExecutor:withBlock:, 30
 - continueWithExecutor:withSuccessBlock:, 30
 - continueWithSuccessBlock:, 30
 - error, 32
 - exception, 32
 - faulted, 32
 - result, 32
 - taskForCompletionOfAllTasks:, 30
 - taskForCompletionOfAllTasksWithResults:, 31
 - taskFromExecutor:withBlock:, 31
 - taskWithDelay:, 31
 - taskWithError:, 31
 - taskWithException:, 31
 - taskWithResult:, 31
 - waitUntilFinished, 32
- BFTask.h
 - BFContinuationBlock, 223
 - BFTaskErrorDomain, 223
 - BFTaskMultipleExceptionsException, 223
- BFTaskCompletionSource, 33
 - cancel, 33
 - setError:, 33
 - setException:, 34
 - setResult:, 34
 - task, 35
 - taskCompletionSource, 34
 - trySetCancelled, 34
 - trySetError:, 34
 - trySetException:, 34
 - trySetResult:, 35
- BFTaskErrorDomain
 - BFTask.h, 223
- BFTaskMultipleExceptionsException
 - BFTask.h, 223
- BFURL, 35
 - appLinkData, 36
 - appLinkExtras, 36
 - appLinkReferer, 36
 - inputQueryParameters, 36
 - inputURL, 36
 - targetQueryParameters, 37

- targetURL, 37
- URLWithInboundURL:sourceApplication:, 36
- URLWithURL:, 36
- BFWebViewAppLinkResolver, 37
 - sharedInstance, 37
- BOLTS_VERSION
 - BoltsVersion.h, 225
- badge
 - PFInstallation, 96
- Ball
 - Game, 42
- ballNotTouchingObstacle
 - Game, 40
- ballSpeed
 - Game.m, 227
- ballTouchingObstacle
 - Game, 40
- become:
 - PFUser, 208
- become:error:
 - PFUser, 208
- becomeInBackground:
 - PFUser, 209
- becomeInBackground:block:
 - PFUser, 209
- becomeInBackground:target:selector:
 - PFUser, 209
- Bolts, 38
 - version, 38
- Bolts.h
 - kBFMultipleErrorsError, 225
- BoltsVersion.h
 - BOLTS_VERSION, 225
- Bound_BreakerTests, 38
- buyProduct:block:
 - PFPurchase, 143
- cachePolicy
 - PFQuery, 177
- callFunction:withParameters:
 - PFCLOUD, 64
- callFunction:withParameters:error:
 - PFCLOUD, 64
- callFunctionInBackground:withParameters:
 - PFCLOUD, 66
- callFunctionInBackground:withParameters:block:
 - PFCLOUD, 66
- callFunctionInBackground:withParameters:target↵
 - :selector:
 - PFCLOUD, 66
- cancel
 - BFTaskCompletionSource, 33
 - PFFile, 83
 - PFQuery, 160
- cancelled
 - BFTask, 32
- cancelledTask
 - BFTask, 29
- channels
 - PFInstallation, 96
- clear
 - PFQueryCollectionViewController, 179
 - PFQueryTableViewController, 184
- clearAllCachedResults
 - PFQuery, 160
- clearCachedResult
 - PFQuery, 160
- clearExpiration
 - PFPush, 147
- closeViewAnimated:
 - BFAppLinkReturnToRefererController, 21
- closed
 - BFAppLinkReturnToRefererView, 24
- collectionView:cellForItemAtIndexPath:object:
 - PFQueryCollectionViewController, 179
- collectionViewReusableViewForNextPageAction:
 - PFQueryCollectionViewController, 180
- completed
 - BFTask, 32
- consumerKey
 - PF_Twitter, 52
- consumerSecret
 - PF_Twitter, 53
- continueWithBlock:
 - BFTask, 29
- continueWithExecutor:withBlock:
 - BFTask, 30
- continueWithExecutor:withSuccessBlock:
 - BFTask, 30
- continueWithSuccessBlock:
 - BFTask, 30
- countObjects
 - PFQuery, 160
- countObjects:
 - PFQuery, 160
- countObjectsInBackground
 - PFQuery, 160
- countObjectsInBackgroundWithBlock:
 - PFQuery, 160
- countObjectsInBackgroundWithTarget:selector:
 - PFQuery, 161
- createdAt
 - PFObject, 138
- currentConfig
 - PFConfig, 68
- currentInstallation
 - PFInstallation, 95
- currentUser
 - PFUser, 209
- decrementActivityCount
 - PFNetworkActivityIndicatorManager, 103
- defaultBallYCoord
 - Game.m, 227
- defaultExecutor
 - BFExecutor, 27
- defaultResolver
 - BFAppLinkNavigation, 18

- delegate
 - BFApplinkReturnToRefererController, 22
 - BFApplinkReturnToRefererView, 24
 - PFLogInViewController, 100
 - PFSignUpViewController, 195
- delete
 - PFObjct, 108
- delete:
 - PFObjct, 109
- deleteAll:
 - PFObjct, 109
- deleteAll:error:
 - PFObjct, 109
- deleteAllInBackground:
 - PFObjct, 109
- deleteAllInBackground:block:
 - PFObjct, 109
- deleteAllInBackground:target:selector:
 - PFObjct, 111
- deleteButton:
 - ProfileViewController, 216
- deleteEventually
 - PFObjct, 111
- deleteInBackground
 - PFObjct, 111
- deleteInBackgroundWithBlock:
 - PFObjct, 111
- deleteInBackgroundWithTarget:selector:
 - PFObjct, 111
- deviceToken
 - PFIInstallation, 96
- deviceType
 - PFIInstallation, 96
- dirty
 - PFObjct, 138
- dismissButton
 - PFLogInView, 98
 - PFSignUpView, 193
- distanceInKilometersTo:
 - PFGeoPoint, 90
- distanceInMilesTo:
 - PFGeoPoint, 90
- distanceInRadiansTo:
 - PFGeoPoint, 90
- downloadAssetForTransaction:completion:
 - PFPurchase, 144
- downloadAssetForTransaction:completion:progress:
 - PFPurchase, 144
- downloadName
 - PFPProduct, 141
- EasyDifficulty:
 - Game, 40
- EasyDifficultyButton
 - Game, 42
- email
 - PFUser, 215
- emailAsUsername
 - PFLogInView, 98
- PFLogInViewController, 100
- PFSignUpView, 193
- PFSignUpViewController, 195
- emailField
 - LoginViewController, 47
 - PFSignUpView, 193
- emailLabel
 - ProfileViewController, 216
- enable
 - ParseCrashReporting, 51
- enableAutomaticUser
 - PFUser, 209
- enableLocalDatastore
 - Parse, 49
- enableRevocableSessionInBackground
 - PFUser, 210
- enableRevocableSessionInBackgroundWithBlock:
 - PFUser, 210
- enabled
 - PFNetworkActivityIndicatorManager, 104
- error
 - BFTask, 32
- estimatedData
 - PFObjct, 138
- exception
 - BFTask, 32
- execute:
 - BFExecutor, 27
- executorWithBlock:
 - BFExecutor, 27
- executorWithDispatchQueue:
 - BFExecutor, 27
- executorWithOperationQueue:
 - BFExecutor, 28
- expireAfterTimeInterval:
 - PFPush, 147
- expireAtDate:
 - PFPush, 147
- extras
 - BFApplinkNavigation, 19
- facebookButton
 - PFLogInView, 98
- facebookLoginBehavior
 - PFFacebookUtils, 72
- facebookLoginManager
 - PFFacebookUtils, 72
- facebookPermissions
 - PFLogInViewController, 100
- faulted
 - BFTask, 32
- fetch
 - PFObjct, 111
- fetch:
 - PFObjct, 111
- fetchAll:
 - PFObjct, 112
- fetchAll:error:
 - PFObjct, 112

- fetchAllIfNeeded:
 - PFObjct, [112](#)
- fetchAllIfNeeded:error:
 - PFObjct, [112](#)
- fetchAllIfNeededInBackground:
 - PFObjct, [112](#)
- fetchAllIfNeededInBackground:block:
 - PFObjct, [112](#)
- fetchAllIfNeededInBackground:target:selector:
 - PFObjct, [114](#)
- fetchAllInBackground:
 - PFObjct, [114](#)
- fetchAllInBackground:block:
 - PFObjct, [114](#)
- fetchAllInBackground:target:selector:
 - PFObjct, [114](#)
- fetchFromLocalDatastore
 - PFObjct, [114](#)
- fetchFromLocalDatastore:
 - PFObjct, [114](#)
- fetchFromLocalDatastoreInBackground
 - PFObjct, [114](#)
- fetchFromLocalDatastoreInBackgroundWithBlock:
 - PFObjct, [115](#)
- fetchIfNeeded
 - PFObjct, [115](#)
- fetchIfNeeded:
 - PFObjct, [115](#)
- fetchIfNeededInBackground
 - PFObjct, [115](#)
- fetchIfNeededInBackgroundWithBlock:
 - PFObjct, [115](#)
- fetchIfNeededInBackgroundWithTarget:selector:
 - PFObjct, [115](#)
- fetchInBackground
 - PFObjct, [115](#)
- fetchInBackgroundWithBlock:
 - PFObjct, [115](#)
- fetchInBackgroundWithTarget:selector:
 - PFObjct, [117](#)
- fields
 - PFLogInView, [98](#)
 - PFLogInViewController, [100](#)
 - PFSignUpView, [193](#)
 - PFSignUpViewController, [195](#)
- file
 - PFIimageView, [94](#)
- fileWithData:
 - PFFile, [83](#)
- fileWithData:contentType:
 - PFFile, [84](#)
- fileWithName:contentsAtPath:
 - PFFile, [84](#)
- fileWithName:data:
 - PFFile, [84](#)
- fileWithName:data:contentType:
 - PFFile, [84](#)
- findObjects
 - PFQuery, [161](#)
- findObjects:
 - PFQuery, [161](#)
- findObjectsInBackground
 - PFQuery, [161](#)
- findObjectsInBackgroundWithBlock:
 - PFQuery, [161](#)
- findObjectsInBackgroundWithTarget:selector:
 - PFQuery, [161](#)
- fromLocalDatastore
 - PFQuery, [161](#)
- fromPin
 - PFQuery, [162](#)
- fromPinWithName:
 - PFQuery, [162](#)
- Game, [39](#)
 - Ball, [42](#)
 - ballNotTouchingObstacle, [40](#)
 - ballTouchingObstacle, [40](#)
 - EasyDifficulty:, [40](#)
 - EasyDifficultyButton, [42](#)
 - GameOver, [40](#), [42](#)
 - HardDifficulty:, [40](#)
 - HardDifficultyButton, [42](#)
 - LeftLiftInside:, [40](#)
 - LeftLiftOutside:, [41](#)
 - LeftPress:, [41](#)
 - longObstacle1, [42](#)
 - longObstacle2, [42](#)
 - longObstacle3, [42](#)
 - longObstacle4, [43](#)
 - MediumDifficulty:, [41](#)
 - MediumDifficultyButton, [43](#)
 - ObstacleMovement, [43](#)
 - ObstaclesMoving, [41](#)
 - PlaceObstacles1, [41](#)
 - PlaceObstacles2, [41](#)
 - RightLiftInside:, [41](#)
 - RightLiftOutside:, [41](#)
 - RightPress:, [42](#)
 - Score, [42](#)
 - ScoreLabel, [43](#)
 - StartGame:, [42](#)
 - TapLeft, [43](#)
 - TapRight, [43](#)
 - WelcomeLabel, [43](#)
- Game(), [43](#)
- Game.h
 - HighScoreNumber, [226](#)
 - Left, [226](#)
 - randomLongObstacle1Placement, [226](#)
 - randomLongObstacle2Placement, [226](#)
 - randomLongObstacle3Placement, [226](#)
 - randomLongObstacle4Placement, [226](#)
 - Right, [226](#)
 - ScoreNumber, [227](#)
- Game.m
 - ballSpeed, [227](#)

- defaultBallYCoord, [227](#)
- longObstacleWidth, [227](#)
- numberOfObstaclesOnScreen, [227](#)
- obstacleDistance, [228](#)
- obstacleGap, [228](#)
- obstacleGeneratorBuffer, [228](#)
- obstacleSpeed, [228](#)
- recoverySpeed, [228](#)
- screenBottom, [228](#)
- screenLeft, [228](#)
- screenRight, [228](#)
- screenTop, [228](#)
- GameOver
 - Game, [40](#), [42](#)
- geoPoint
 - PFGeoPoint, [92](#)
- geoPointForCurrentLocationInBackground:
 - PFGeoPoint, [92](#)
- geoPointWithLatitude:longitude:
 - PFGeoPoint, [92](#)
- geoPointWithLocation:
 - PFGeoPoint, [92](#)
- getApplicationId
 - Parse, [49](#)
- getClientKey
 - Parse, [49](#)
- getConfig
 - PFCConfig, [68](#)
- getConfig:
 - PFCConfig, [69](#)
- getConfigInBackground
 - PFCConfig, [69](#)
- getConfigInBackgroundWithBlock:
 - PFCConfig, [69](#)
- getCurrentSessionInBackground
 - PFSession, [191](#)
- getCurrentSessionInBackgroundWithBlock:
 - PFSession, [191](#)
- getData
 - PFFFile, [85](#)
- getData:
 - PFFFile, [85](#)
- getDataInBackground
 - PFFFile, [85](#)
- getDataInBackgroundWithBlock:
 - PFFFile, [85](#)
- getDataInBackgroundWithBlock:progressBlock:
 - PFFFile, [85](#)
- getDataInBackgroundWithTarget:selector:
 - PFFFile, [86](#)
- getDataStream
 - PFFFile, [86](#)
- getDataStream:
 - PFFFile, [86](#)
- getDataStreamInBackground
 - PFFFile, [86](#)
- getDataStreamInBackgroundWithBlock:
 - PFFFile, [86](#)
- getDataStreamInBackgroundWithBlock:progress↔
 - Block:
 - PFFFile, [86](#)
- getFirstObject
 - PFQuery, [162](#)
- getFirstObject:
 - PFQuery, [163](#)
- getFirstObjectInBackground
 - PFQuery, [163](#)
- getFirstObjectInBackgroundWithBlock:
 - PFQuery, [163](#)
- getFirstObjectInBackgroundWithTarget:selector:
 - PFQuery, [164](#)
- getObjectInBackgroundWithId:
 - PFQuery, [164](#)
- getObjectInBackgroundWithId:block:
 - PFQuery, [164](#)
- getObjectInBackgroundWithId:target:selector:
 - PFQuery, [164](#)
- getObjectOfClass:objectId:
 - PFQuery, [164](#)
- getObjectOfClass:objectId:error:
 - PFQuery, [165](#)
- getObjectWithId:
 - PFQuery, [165](#)
- getObjectWithId:error:
 - PFQuery, [165](#)
- getPublicReadAccess
 - PFAcl, [55](#)
- getPublicWriteAccess
 - PFAcl, [55](#)
- getReadAccessForRole:
 - PFAcl, [55](#)
- getReadAccessForRoleWithName:
 - PFAcl, [55](#)
- getReadAccessForUser:
 - PFAcl, [55](#)
- getReadAccessForUserId:
 - PFAcl, [56](#)
- getSubscribedChannels:
 - PFPush, [148](#)
- getSubscribedChannelsInBackground
 - PFPush, [148](#)
- getSubscribedChannelsInBackgroundWithBlock:
 - PFPush, [148](#)
- getSubscribedChannelsInBackgroundWithTarget↔
 - :selector:
 - PFPush, [148](#)
- getUserObjectWithId:
 - PFQuery, [166](#)
- getUserObjectWithId:error:
 - PFQuery, [166](#)
- getWriteAccessForRole:
 - PFAcl, [56](#)
- getWriteAccessForRoleWithName:
 - PFAcl, [56](#)
- getWriteAccessForUser:
 - PFAcl, [56](#)

- getWriteAccessForUserId:
 - PFACL, 57
- handleOpenURL:
 - PFFacebookUtils, 72
- handlePush:
 - PFPush, 148
- HardDifficulty:
 - Game, 40
- HardDifficultyButton
 - Game, 42
- hasCachedResult
 - PFQuery, 166
- highScoreLabel
 - ProfileViewController, 216
- HighScoreNumber
 - Game.h, 226
- icon
 - PFProduct, 141
- ignoreACLs
 - PFQuery, 166
- imageKey
 - PFQueryTableViewController, 186
- imageView
 - PFCollectionViewCell, 67
 - PFTableViewCell, 200
- immediateExecutor
 - BFExecutor, 28
- includeKey:
 - PFQuery, 166
- includeStatusBarInSize
 - BFAppLinkReturnToRefererView, 24
- incrementActivityCount
 - PFNetworkActivityIndicatorManager, 103
- incrementKey:
 - PFObjct, 117
- incrementKey:byAmount:
 - PFObjct, 117
- init
 - BFAppLinkReturnToRefererController, 21
 - PFObjct, 117
 - PFObjct(Subclass), 139
- initWithDisplayAboveNavController:
 - BFAppLinkReturnToRefererController, 22
- initWithClassName:
 - PFObjct, 117
 - PFQuery, 168
 - PFQueryCollectionViewController, 180
 - PFQueryTableViewController, 184
- initWithCollectionViewLayout:className:
 - PFQueryCollectionViewController, 180
- initWithFields:
 - PFLoginView, 98
 - PFSignUpView, 193
- initWithFrame:separatorStyle:
 - PFTextField, 200
- initWithName:
 - PFRole, 189
- initWithName:acl:
 - PFRole, 189
- initWithStyle:
 - PFProductTableViewController, 142
- initWithStyle:className:
 - PFQueryTableViewController, 184
- initializeFacebook
 - PFFacebookUtils, 72
- initializeFacebookWithApplicationLaunchOptions:
 - PFFacebookUtils, 72
- initializeFacebookWithURLSchemeSuffix:
 - PFFacebookUtils, 73
- initializeWithApplicationId:
 - PFFacebookUtils, 73
- initializeWithApplicationId:urlSchemeSuffix:
 - PFFacebookUtils, 73
- initializeWithConsumerKey:consumerSecret:
 - PFTwitterUtils, 202
- inputQueryParameters
 - BFURL, 36
- inputURL
 - BFURL, 36
- installationId
 - PFInstallation, 96
- isAuthenticated
 - PFUser, 210
- isCrashReportingEnabled
 - ParseCrashReporting, 51
- isDataAvailable
 - PFFile, 89
 - PFObjct, 117
- isDirty
 - PFFile, 89
 - PFObjct, 117
- isDirtyForKey:
 - PFObjct, 118
- isLinkedWithUser:
 - PFAnonymousUtils, 63
 - PFFacebookUtils, 73, 74
 - PFTwitterUtils, 202
- isLocalDatastoreEnabled
 - Parse, 49
- isNew
 - PFUser, 215
- kBFMultipleErrorsError
 - Bolts.h, 225
- kPFDeviceType
 - PFConstants.h, 238
- kPFParseServer
 - PFConstants.h, 238
- latitude
 - PFGeoPoint, 92
- leaderBoardViewController, 44
 - score1, 44
 - score10, 44
 - score2, 45
 - score3, 45

- score4, [45](#)
- score5, [45](#)
- score6, [45](#)
- score7, [45](#)
- score8, [45](#)
- score9, [45](#)
- username1, [45](#)
- username10, [45](#)
- username2, [45](#)
- username3, [45](#)
- username4, [46](#)
- username5, [46](#)
- username6, [46](#)
- username7, [46](#)
- username8, [46](#)
- username9, [46](#)
- leaderBoardViewController(), [46](#)
- Left
 - Game.h, [226](#)
- LeftLiftInside:
 - Game, [40](#)
- LeftLiftOutside:
 - Game, [41](#)
- LeftPress:
 - Game, [41](#)
- limit
 - PFQuery, [177](#)
- linkUser:
 - PFTwitterUtils, [203](#)
- linkUser:block:
 - PFTwitterUtils, [203](#)
- linkUser:facebookId:accessToken:expirationDate↔
 - :block:
 - PFFacebookUtils, [74](#)
- linkUser:facebookId:accessToken:expirationDate↔
 - :target:selector:
 - PFFacebookUtils, [74](#)
- linkUser:permissions:
 - PFFacebookUtils, [74](#)
- linkUser:permissions:block:
 - PFFacebookUtils, [74](#)
- linkUser:permissions:target:selector:
 - PFFacebookUtils, [75](#)
- linkUser:target:selector:
 - PFTwitterUtils, [203](#)
- linkUser:twitterId:screenName:authToken:authToken↔
 - Secret:block:
 - PFTwitterUtils, [203](#)
- linkUser:twitterId:screenName:authToken:authToken↔
 - Secret:target:selector:
 - PFTwitterUtils, [203](#)
- linkUserInBackground:
 - PFTwitterUtils, [203](#)
- linkUserInBackground:facebookId:accessToken↔
 - :expirationDate:
 - PFFacebookUtils, [75](#)
- linkUserInBackground:permissions:
 - PFFacebookUtils, [75](#)
- linkUserInBackground:twitterId:screenName:authToken:authTokenSecret:
 - PFTwitterUtils, [204](#)
- linkUserInBackground:withAccessToken:
 - PFFacebookUtils, [75](#)
- linkUserInBackground:withAccessToken:block:
 - PFFacebookUtils, [76](#)
- linkUserInBackground:withPublishPermissions:
 - PFFacebookUtils, [76](#)
- linkUserInBackground:withPublishPermissions:block:
 - PFFacebookUtils, [76](#)
- linkUserInBackground:withReadPermissions:
 - PFFacebookUtils, [76](#)
- linkUserInBackground:withReadPermissions:block:
 - PFFacebookUtils, [77](#)
- loadInBackground
 - PFImageView, [93](#)
- loadInBackground:
 - PFImageView, [93](#)
- loadInBackground:progressBlock:
 - PFImageView, [94](#)
- loadNextPage
 - PFQueryCollectionViewController, [180](#)
 - PFQueryTableViewController, [184](#)
- loadObjects
 - PFQueryCollectionViewController, [180](#)
 - PFQueryTableViewController, [184](#)
- loadObjects:clear:
 - PFQueryCollectionViewController, [181](#)
 - PFQueryTableViewController, [184](#)
- loading
 - PFQueryCollectionViewController, [181](#)
 - PFQueryTableViewController, [186](#)
- loadingViewEnabled
 - PFQueryCollectionViewController, [182](#)
 - PFQueryTableViewController, [186](#)
- loginButton
 - PFLogInView, [98](#)
- loginInBackground
 - PFAnonymousUtils, [63](#)
 - PFTwitterUtils, [204](#)
- loginInBackgroundWithAccessToken:
 - PFFacebookUtils, [77](#)
- loginInBackgroundWithAccessToken:block:
 - PFFacebookUtils, [77](#)
- loginInBackgroundWithPublishPermissions:
 - PFFacebookUtils, [77](#)
- loginInBackgroundWithPublishPermissions:block:
 - PFFacebookUtils, [78](#)
- loginInBackgroundWithReadPermissions:
 - PFFacebookUtils, [78](#)
- loginInBackgroundWithReadPermissions:block:
 - PFFacebookUtils, [78](#)
- loginView
 - PFLogInViewController, [101](#)
- loginViewController:didFailToLoginWithError:
 - PFLogInViewControllerDelegate-p, [102](#)
- loginViewController:didLoginUser:

- PFLoginViewControllerDelegate-p, [102](#)
- loginViewController:shouldBeginLoginWithUsername↔
 - :password:
 - PFLoginViewControllerDelegate-p, [102](#)
- loginViewControllerDidCancelLogin:
 - PFLoginViewControllerDelegate-p, [102](#)
- loginWithBlock:
 - PFAnonymousUtils, [63](#)
 - PFTwitterUtils, [204](#)
- loginWithFacebookId:accessToken:expirationDate↔
 - :block:
 - PFFacebookUtils, [78](#)
- loginWithFacebookId:accessToken:expirationDate↔
 - :target:selector:
 - PFFacebookUtils, [79](#)
- loginWithFacebookIdInBackground:accessToken↔
 - :expirationDate:
 - PFFacebookUtils, [79](#)
- loginWithPermissions:block:
 - PFFacebookUtils, [79](#)
- loginWithPermissions:target:selector:
 - PFFacebookUtils, [79](#)
- loginWithPermissionsInBackground:
 - PFFacebookUtils, [79](#)
- loginWithTarget:selector:
 - PFAnonymousUtils, [64](#)
 - PFTwitterUtils, [204](#)
- loginWithTwitterId:screenName:authToken:authToken↔
 - Secret:block:
 - PFTwitterUtils, [205](#)
- loginWithTwitterId:screenName:authToken:authToken↔
 - Secret:target:selector:
 - PFTwitterUtils, [205](#)
- loginWithTwitterIdInBackground:screenName:auth↔
 - Token:authTokenSecret:
 - PFTwitterUtils, [205](#)
- loginWithUsername:password:
 - PFUser, [210](#)
- loginWithUsername:password:error:
 - PFUser, [211](#)
- loginWithUsernameInBackground:password:
 - PFUser, [211](#)
- loginWithUsernameInBackground:password:block:
 - PFUser, [211](#)
- loginWithUsernameInBackground:password:target↔
 - :selector:
 - PFUser, [212](#)
- logLevel
 - Parse, [49](#)
- logout
 - PFUser, [212](#)
- logoutInBackground
 - PFUser, [212](#)
- logoutInBackgroundWithBlock:
 - PFUser, [212](#)
- loginButton:
 - LoginViewController, [47](#)
- loginPasswordField
- LoginViewController, [47](#)
- loginUsernameField
 - LoginViewController, [47](#)
- LoginViewController, [46](#)
 - emailField, [47](#)
 - loginButton:, [47](#)
 - loginPasswordField, [47](#)
 - loginUsernameField, [47](#)
 - passwordField, [47](#)
 - reEnterPasswordField, [48](#)
 - registerAction:, [47](#)
 - userNameField, [48](#)
- LoginViewController(), [48](#)
- logo
 - PFLoginView, [98](#)
 - PFSignUpView, [193](#)
- logoutButton:
 - ProfileViewController, [216](#)
- longObstacle1
 - Game, [42](#)
- longObstacle2
 - Game, [42](#)
- longObstacle3
 - Game, [42](#)
- longObstacle4
 - Game, [43](#)
- longObstacleWidth
 - Game.m, [227](#)
- longitude
 - PFGeoPoint, [92](#)
- main
 - main.m, [230](#)
- main.m
 - main, [230](#)
- mainThreadExecutor
 - BFExecutor, [28](#)
- maxCacheAge
 - PFQuery, [177](#)
- MediumDifficulty:
 - Game, [41](#)
- MediumDifficultyButton
 - Game, [43](#)
- minPasswordLength
 - PFSignUpViewController, [195](#)
- NS_ENUM
 - BFAppLinkNavigation.h, [220](#)
 - PFConstants.h, [236](#), [237](#)
 - PFPurchaseTableViewCell.h, [250](#)
- NS_OPTIONS
 - PFLoginView.h, [248](#)
 - PFSignUpView.h, [251](#)
 - PFTextField.h, [253](#)
- NS_REQUIRES_SUPER
 - PFQueryCollectionViewController, [181](#)
- NSColor
 - PFConstants.h, [238](#)
- UIImage

- PFConstants.h, 238
- NSView
 - PFConstants.h, 239
- name
 - PFFile, 89
 - PFRole, 190
- navigate:
 - BFAppLinkNavigation, 18
- navigateToAppLink:error:
 - BFAppLinkNavigation, 18
- navigateToURLInBackground:
 - BFAppLinkNavigation, 18
- navigateToURLInBackground:resolver:
 - BFAppLinkNavigation, 18
- navigationWithAppLink:extras:appLinkData:
 - BFAppLinkNavigation, 18
- networkActivityIndicatorCount
 - PFNetworkActivityIndicatorManager, 104
- networkActivityIndicatorVisible
 - PFNetworkActivityIndicatorManager, 104
- numberOfObstaclesOnScreen
 - Game.m, 227
- object
 - PFObject, 118
 - PFObject(Subclass), 139
 - PFSubclassing-p, 198
- objectAtIndexPath:
 - PFQueryCollectionViewController, 181
 - PFQueryTableViewController, 185
- objectForKey:
 - PFFConfig, 69
 - PFObject, 118
- objectForKeyedSubscript:
 - PFFConfig, 69
 - PFObject, 118
- objectId
 - PFObject, 138
- objectWithClassName:
 - PFObject, 118
- objectWithClassName:dictionary:
 - PFObject, 119
- objectWithoutDataWithClassName:objectId:
 - PFObject, 119
- objectWithoutDataWithObjectId:
 - PFObject, 119
 - PFObject(Subclass), 139
 - PFSubclassing-p, 198
- objects
 - PFQueryCollectionViewController, 182
 - PFQueryTableViewController, 186
- objectsDidLoad:
 - PFQueryCollectionViewController, 181
 - PFQueryTableViewController, 185
- objectsPerPage
 - PFQueryCollectionViewController, 182
 - PFQueryTableViewController, 186
- objectsWillLoad
 - PFQueryTableViewController, 185
- obstacleDistance
 - Game.m, 228
- obstacleGap
 - Game.m, 228
- obstacleGeneratorBuffer
 - Game.m, 228
- ObstacleMovement
 - Game, 43
- obstacleSpeed
 - Game.m, 228
- ObstaclesMoving
 - Game, 41
- operationSetQueue
 - PFObject, 138
- orQueryWithSubqueries:
 - PFQuery, 169
- order
 - PFProduct, 141
- orderByAscending:
 - PFQuery, 168
- orderByDescending:
 - PFQuery, 168
- orderBySortDescriptor:
 - PFQuery, 168
- orderBySortDescriptors:
 - PFQuery, 168
- PARSE_API_VERSION
 - PFConstants.h, 239
- PARSE_DEPRECATED
 - PFConstants.h, 234
 - PFQuery, 169
- PARSE_IOS_ONLY
 - PFConstants.h, 234
- PARSE_OSX_ONLY
 - PFConstants.h, 234
- PARSE_UI_DEPRECATED
 - ParseUIConstants.h, 247
- PARSE_VERSION
 - PFConstants.h, 234
- PF_ASSUME_NONNULL_BEGIN
 - PFNullability.h, 240
- PF_ASSUME_NONNULL_END
 - PFNullability.h, 240
- PF_NONNULL
 - PFNullability.h, 240
- PF_NONNULL_S
 - PFNullability.h, 241
- PF_NULLABLE
 - PFNullability.h, 241
- PF_NULLABLE_PROPERTY
 - PFNullability.h, 241
- PF_NULLABLE_S
 - PFNullability.h, 241
- PF_Twitter, 51
 - authToken, 52
 - authTokenSecret, 52
 - authorizeInBackground, 52
 - authorizeWithSuccess:failure:cancel:, 52

- consumerKey, 52
- consumerSecret, 53
- screenName, 53
- signRequest:, 52
- userId, 53
- PFACL, 53
 - ACL, 54
 - ACLWithUser:, 54
 - getPublicReadAccess, 55
 - getPublicWriteAccess, 55
 - getReadAccessForRole:, 55
 - getReadAccessForRoleWithName:, 55
 - getReadAccessForUser:, 55
 - getReadAccessForUserId:, 56
 - getWriteAccessForRole:, 56
 - getWriteAccessForRoleWithName:, 56
 - getWriteAccessForUser:, 56
 - getWriteAccessForUserId:, 57
 - setDefaultACL:withAccessForCurrentUser:, 57
 - setPublicReadAccess:, 57
 - setPublicWriteAccess:, 57
 - setReadAccess:forRole:, 58
 - setReadAccess:forRoleWithName:, 58
 - setReadAccess:forUser:, 58
 - setReadAccess:forUserId:, 58
 - setWriteAccess:forRole:, 58
 - setWriteAccess:forRoleWithName:, 58
 - setWriteAccess:forUser:, 59
 - setWriteAccess:forUserId:, 59
- PFAalytics, 59
 - trackAppOpenedWithLaunchOptions:, 60
 - trackAppOpenedWithLaunchOptionsInBackground↵:block:, 60
 - trackAppOpenedWithRemoteNotificationPayload:, 60
 - trackAppOpenedWithRemoteNotificationPayload↵InBackground:block:, 60
 - trackEvent:, 61
 - trackEvent:dimensions:, 61
 - trackEventInBackground:block:, 61
 - trackEventInBackground:dimensions:block:, 62
- PFAnonymousUtils, 62
 - isLinkedWithUser:, 63
 - logInInBackground, 63
 - logInWithBlock:, 63
 - logInWithTarget:selector:, 64
- PFArrayResultBlock
 - PFConstants.h, 235
- PFBooleanResultBlock
 - PFConstants.h, 235
- PFCloud, 64
 - callFunction:withParameters:, 64
 - callFunction:withParameters:error:, 64
 - callFunctionInBackground:withParameters:, 66
 - callFunctionInBackground:withParameters:block:, 66
 - callFunctionInBackground:withParameters:target↵:selector:, 66
- PFCollectionViewCell, 66
 - imageView, 67
 - textLabel, 67
 - updateFromObject:, 67
- PFConfig, 68
 - currentConfig, 68
 - getConfig, 68
 - getConfig:, 69
 - getConfigInBackground, 69
 - getConfigInBackgroundWithBlock:, 69
 - objectForKey:, 69
 - objectForKeyedSubscript:, 69
- PFConfig.h
 - PFConfigResultBlock, 233
- PFConfigResultBlock
 - PFConfig.h, 233
- PFConstants.h
 - kPFDeviceType, 238
 - kPFParseServer, 238
 - NS_ENUM, 236, 237
 - NSColor, 238
 - NSImage, 238
 - NSView, 239
 - PARSE_API_VERSION, 239
 - PARSE_DEPRECATED, 234
 - PARSE_IOS_ONLY, 234
 - PARSE_OSX_ONLY, 234
 - PARSE_VERSION, 234
 - PFArrayResultBlock, 235
 - PFBooleanResultBlock, 235
 - PFDataResultBlock, 235
 - PFDataStreamResultBlock, 235
 - PFIdResultBlock, 235
 - PFIntegerResultBlock, 235
 - PFOBJECTResultBlock, 235
 - PFParseErrorDomain, 239
 - PFProgressBlock, 235
 - PFSetResultBlock, 235
 - PFStringResultBlock, 235
 - PFUserResultBlock, 235
- PFDataResultBlock
 - PFConstants.h, 235
- PFDataStreamResultBlock
 - PFConstants.h, 235
- PFFacebookUtils, 70
 - facebookLoginBehavior, 72
 - facebookLoginManager, 72
 - handleOpenURL:, 72
 - initializeFacebook, 72
 - initializeFacebookWithApplicationLaunchOptions:, 72
 - initializeFacebookWithUrlSchemeSuffix:, 73
 - initializeWithApplicationId:, 73
 - initializeWithApplicationId:urlSchemeSuffix:, 73
 - isLinkedWithUser:, 73, 74
 - linkUser:facebookId:accessToken:expirationDate↵:block:, 74

- linkUser:facebookId:accessToken:expirationDate↔
:target:selector:, 74
- linkUser:permissions:, 74
- linkUser:permissions:block:, 74
- linkUser:permissions:target:selector:, 75
- linkUserInBackground:facebookId:accessToken↔
:expirationDate:, 75
- linkUserInBackground:permissions:, 75
- linkUserInBackground:withAccessToken:, 75
- linkUserInBackground:withAccessToken:block:, 76
- linkUserInBackground:withPublishPermissions:, 76
- linkUserInBackground:withPublishPermissions↔
:block:, 76
- linkUserInBackground:withReadPermissions:, 76
- linkUserInBackground:withReadPermissions↔
:block:, 77
- logInInBackgroundWithAccessToken:, 77
- logInInBackgroundWithAccessToken:block:, 77
- logInInBackgroundWithPublishPermissions:, 77
- logInInBackgroundWithPublishPermissions↔
:block:, 78
- logInInBackgroundWithReadPermissions:, 78
- logInInBackgroundWithReadPermissions:block:, 78
- logInWithFacebookId:accessToken:expiration↔
Date:block:, 78
- logInWithFacebookId:accessToken:expiration↔
Date:target:selector:, 79
- logInWithFacebookIdInBackground:accessToken↔
:expirationDate:, 79
- logInWithPermissions:block:, 79
- logInWithPermissions:target:selector:, 79
- logInWithPermissionsInBackground:, 79
- reauthorizeUser:withPublishPermissions:audience↔
:block:, 80
- reauthorizeUser:withPublishPermissions:audience↔
:target:selector:, 80
- reauthorizeUserInBackground:withPublish↔
Permissions:audience:, 80
- session, 80
- setFacebookLoginBehavior:, 81
- unlinkUser:, 81
- unlinkUser:error:, 81
- unlinkUserInBackground:, 81
- unlinkUserInBackground:block:, 82
- unlinkUserInBackground:target:selector:, 82
- PFFile, 82
 - cancel, 83
 - fileWithData:, 83
 - fileWithData:contentType:, 84
 - fileWithName:contentsAtPath:, 84
 - fileWithName:data:, 84
 - fileWithName:data:contentType:, 84
 - getData, 85
 - getData:, 85
 - getDataInBackground, 85
 - getDataInBackgroundWithBlock:, 85
 - getDataInBackgroundWithBlock:progressBlock:, 85
 - getDataInBackgroundWithTarget:selector:, 86
 - getDataStream, 86
 - getDataStream:, 86
 - getDataStreamInBackground, 86
 - getDataStreamInBackgroundWithBlock:, 86
 - getDataStreamInBackgroundWithBlock:progress↔
Block:, 86
 - isDataAvailable, 89
 - isDirty, 89
 - name, 89
 - save, 88
 - save:, 88
 - saveInBackground, 88
 - saveInBackgroundWithBlock:, 88
 - saveInBackgroundWithBlock:progressBlock:, 88
 - saveInBackgroundWithTarget:selector:, 89
 - url, 89
- PFGeoPoint, 89
 - distanceInKilometersTo:, 90
 - distanceInMilesTo:, 90
 - distanceInRadiansTo:, 90
 - geoPoint, 92
 - geoPointForCurrentLocationInBackground:, 92
 - geoPointWithLatitude:longitude:, 92
 - geoPointWithLocation:, 92
 - latitude, 92
 - longitude, 92
- PFGeoPoint.h
 - PFGeoPointResultBlock, 239
- PFGeoPointResultBlock
 - PFGeoPoint.h, 239
- PFIidResultBlock
 - PConstants.h, 235
- PFIImageView, 93
 - file, 94
 - loadInBackground, 93
 - loadInBackground:, 93
 - loadInBackground:progressBlock:, 94
- PFIImageView.h
 - PFIImageViewImageResultBlock, 248
- PFIImageViewImageResultBlock
 - PFIImageView.h, 248
- PFIInstallation, 94
 - badge, 96
 - channels, 96
 - currentInstallation, 95
 - deviceToken, 96
 - deviceType, 96
 - installationId, 96
 - query, 95
 - setDeviceTokenFromData:, 95
 - timeZone, 96
- PFIIntegerResultBlock
 - PConstants.h, 235
- PFILogInCancelNotification
 - PFILogInViewController.h, 249

- PFLogInFailureNotification
 - PFLogInViewController.h, 249
- PFLogInSuccessNotification
 - PFLogInViewController.h, 249
- PFLogInView, 96
 - __attribute__, 97
 - dismissButton, 98
 - emailAsUsername, 98
 - facebookButton, 98
 - fields, 98
 - initWithFields:, 98
 - loginButton, 98
 - logo, 98
 - passwordField, 99
 - passwordForgottenButton, 99
 - presentingViewController, 99
 - signupButton, 99
 - twitterButton, 99
 - usernameField, 99
- PFLogInView.h
 - NS_OPTIONS, 248
- PFLogInViewController, 99
 - delegate, 100
 - emailAsUsername, 100
 - facebookPermissions, 100
 - fields, 100
 - loginView, 101
 - signupController, 101
- PFLogInViewController.h
 - PFLogInCancelNotification, 249
 - PFLogInFailureNotification, 249
 - PFLogInSuccessNotification, 249
- PFLogInViewControllerDelegate-p
 - loginViewController:didFailToLoginWithError:, 102
 - loginViewController:didLogInUser:, 102
 - loginViewController:shouldBeginLoginWith←
 - Username:password:, 102
 - loginViewControllerDidCancelLogin:, 102
- PFNetworkActivityIndicatorManager, 103
 - decrementActivityCount, 103
 - enabled, 104
 - incrementActivityCount, 103
 - networkActivityCount, 104
 - networkActivityIndicatorVisible, 104
 - sharedManager, 103
- PFNullability.h
 - PF_ASSUME_NONNULL_BEGIN, 240
 - PF_ASSUME_NONNULL_END, 240
 - PF_NONNULL, 240
 - PF_NONNULL_S, 241
 - PF_NULLABLE, 241
 - PF_NULLABLE_PROPERTY, 241
 - PF_NULLABLE_S, 241
- PFObject, 104
 - ACL, 138
 - addObject:forKey:, 108
 - addObjectsFromArray:forKey:, 108
 - addUniqueObject:forKey:, 108
 - addUniqueObjectsFromArray:forKey:, 108
 - allKeys, 108
 - createdAt, 138
 - delete, 108
 - delete:, 109
 - deleteAll:, 109
 - deleteAll:error:, 109
 - deleteAllInBackground:, 109
 - deleteAllInBackground:block:, 109
 - deleteAllInBackground:target:selector:, 111
 - deleteEventually, 111
 - deleteInBackground, 111
 - deleteInBackgroundWithBlock:, 111
 - deleteInBackgroundWithTarget:selector:, 111
 - dirty, 138
 - estimatedData, 138
 - fetch, 111
 - fetch:, 111
 - fetchAll:, 112
 - fetchAll:error:, 112
 - fetchAllIfNeeded:, 112
 - fetchAllIfNeeded:error:, 112
 - fetchAllIfNeededInBackground:, 112
 - fetchAllIfNeededInBackground:block:, 112
 - fetchAllIfNeededInBackground:target:selector:, 114
 - fetchAllInBackground:, 114
 - fetchAllInBackground:block:, 114
 - fetchAllInBackground:target:selector:, 114
 - fetchFromLocalDatastore, 114
 - fetchFromLocalDatastore:, 114
 - fetchFromLocalDatastoreInBackground, 114
 - fetchFromLocalDatastoreInBackgroundWith←
 - Block:, 115
 - fetchIfNeeded, 115
 - fetchIfNeeded:, 115
 - fetchIfNeededInBackground, 115
 - fetchIfNeededInBackgroundWithBlock:, 115
 - fetchIfNeededInBackgroundWithTarget:selector:, 115
 - fetchInBackground, 115
 - fetchInBackgroundWithBlock:, 115
 - fetchInBackgroundWithTarget:selector:, 117
 - incrementKey:, 117
 - incrementKey:byAmount:, 117
 - init, 117
 - initWithClassName:, 117
 - isDataAvailable, 117
 - isDirty, 117
 - isDirtyForKey:, 118
 - object, 118
 - objectForKey:, 118
 - objectForKeyedSubscript:, 118
 - objectId, 138
 - objectWithClassName:, 118
 - objectWithClassName:dictionary:, 119
 - objectWithoutDataWithClassName:objectId:, 119
 - objectWithoutDataWithObjectId:, 119
 - operationSetQueue, 138

- parseClassName, 138
- pin, 119
- pin:, 120
- pinAll:, 120
- pinAll:error:, 121
- pinAll:withName:, 121
- pinAll:withName:error:, 121
- pinAllInBackground:, 122
- pinAllInBackground:block:, 122
- pinAllInBackground:withName:, 123
- pinAllInBackground:withName:block:, 123
- pinInBackground, 123
- pinInBackgroundWithBlock:, 124
- pinInBackgroundWithName:, 124
- pinInBackgroundWithName:block:, 124
- pinWithName:, 125
- pinWithName:error:, 125
- query, 126
- queryWithPredicate:, 126
- registerSubclass, 126
- relationForKey:, 126
- relationforKey:, 127
- removeObject:forKey:, 127
- removeObjectForKey:, 127
- removeObjectsInArray:forKey:, 127
- save, 127
- save:, 127
- saveAll:, 128
- saveAll:error:, 128
- saveAllInBackground:, 128
- saveAllInBackground:block:, 128
- saveAllInBackground:target:selector:, 128
- saveEventually, 129
- saveEventually:, 129
- saveInBackground, 129
- saveInBackgroundWithBlock:, 129
- saveInBackgroundWithTarget:selector:, 129
- setObject:forKey:, 130
- setObject:forKeyedSubscript:, 131
- unpin, 131
- unpin:, 131
- unpinAll:, 131
- unpinAll:error:, 132
- unpinAll:withName:, 132
- unpinAll:withName:error:, 132
- unpinAllInBackground:, 133
- unpinAllInBackground:block:, 133
- unpinAllInBackground:withName:, 133
- unpinAllInBackground:withName:block:, 134
- unpinAllObjects, 134
- unpinAllObjects:, 134
- unpinAllObjectsInBackground, 134
- unpinAllObjectsInBackgroundWithBlock:, 135
- unpinAllObjectsInBackgroundWithName:, 135
- unpinAllObjectsInBackgroundWithName:block↵
:, 135
- unpinAllObjectsWithName:, 135
- unpinAllObjectsWithName:error:, 136
- unpinInBackground, 136
- unpinInBackgroundWithBlock:, 136
- unpinInBackgroundWithName:, 136
- unpinInBackgroundWithName:block:, 137
- unpinWithName:, 137
- unpinWithName:error:, 137
- updatedAt, 138
- PFObjec(Subclass), 138
 - init, 139
 - object, 139
 - objectWithoutDataWithObjectId:, 139
 - query, 139
 - queryWithPredicate:, 139
 - registerSubclass, 140
- PFObjec.h
 - PFObjecDefaultPin, 242
- PFObjecDefaultPin
 - PFObjec.h, 242
- PFObjecResultBlock
 - PConstants.h, 235
- PFParseErrorDomain
 - PConstants.h, 239
- PFPProduct, 140
 - downloadName, 141
 - icon, 141
 - order, 141
 - productIdentifier, 141
 - subtitle, 141
 - title, 141
- PFPProductTableViewCell, 142
 - initWithStyle:, 142
- PFPProgressBlock
 - PConstants.h, 235
- PFPurchase, 143
 - addObserverForProduct:block:, 143
 - assetContentPathForProduct:, 143
 - buyProduct:block:, 143
 - downloadAssetForTransaction:completion:, 144
 - downloadAssetForTransaction:completion↵
:progress:, 144
 - restore, 144
- PFPurchaseTableViewCell, 144
 - priceLabel, 145
 - progressView, 145
 - state, 145
- PFPurchaseTableViewCell.h
 - NS_ENUM, 250
- PFPush, 146
 - clearExpiration, 147
 - expireAfterTimeInterval:, 147
 - expireAtDate:, 147
 - getSubscribedChannels:, 148
 - getSubscribedChannelsInBackground, 148
 - getSubscribedChannelsInBackgroundWithBlock:, 148
 - getSubscribedChannelsInBackgroundWith↵
Target:selector:, 148
 - handlePush:, 148

- push, 149
- sendPush:, 149
- sendPushDataToChannel:withData:error:, 149
- sendPushDataToChannelInBackground:with↵
Data:, 149
- sendPushDataToChannelInBackground:with↵
Data:block:, 149
- sendPushDataToChannelInBackground:with↵
Data:target:selector:, 150
- sendPushDataToQuery:withData:error:, 150
- sendPushDataToQueryInBackground:withData:, 150
- sendPushDataToQueryInBackground:withData↵
:block:, 150
- sendPushInBackground, 151
- sendPushInBackgroundWithBlock:, 151
- sendPushInBackgroundWithTarget:selector:, 151
- sendPushMessageToChannel:withMessage↵
:error:, 151
- sendPushMessageToChannelInBackground:with↵
Message:, 151
- sendPushMessageToChannelInBackground:with↵
Message:block:, 151
- sendPushMessageToChannelInBackground:with↵
Message:target:selector:, 153
- sendPushMessageToQuery:withMessage:error:, 153
- sendPushMessageToQueryInBackground:with↵
Message:, 153
- sendPushMessageToQueryInBackground:with↵
Message:block:, 153
- setChannel:, 153
- setChannels:, 154
- setData:, 154
- setMessage:, 154
- setPushToAndroid:, 154
- setPushToIOS:, 154
- setQuery:, 155
- storeDeviceToken:, 155
- subscribeToChannel:error:, 155
- subscribeToChannelInBackground:, 155
- subscribeToChannelInBackground:block:, 155
- subscribeToChannelInBackground:target:selector↵
:, 156
- unsubscribeFromChannel:error:, 156
- unsubscribeFromChannelInBackground:, 156
- unsubscribeFromChannelInBackground:block↵
:, 156
- unsubscribeFromChannelInBackground:target↵
:selector:, 156
- PFQuery, 157
 - addAscendingOrder:, 159
 - addDescendingOrder:, 159
 - cachePolicy, 177
 - cancel, 160
 - clearAllCachedResults, 160
 - clearCachedResult, 160
 - countObjects, 160
 - countObjects:, 160
 - countObjectsInBackground, 160
 - countObjectsInBackgroundWithBlock:, 160
 - countObjectsInBackgroundWithTarget:selector:, 161
 - findObjects, 161
 - findObjects:, 161
 - findObjectsInBackground, 161
 - findObjectsInBackgroundWithBlock:, 161
 - findObjectsInBackgroundWithTarget:selector:, 161
 - fromLocalDatastore, 161
 - fromPin, 162
 - fromPinWithName:, 162
 - getFirstObject, 162
 - getFirstObject:, 163
 - getFirstObjectInBackground, 163
 - getFirstObjectInBackgroundWithBlock:, 163
 - getFirstObjectInBackgroundWithTarget:selector:, 164
 - getObjectInBackgroundWithId:, 164
 - getObjectInBackgroundWithId:block:, 164
 - getObjectInBackgroundWithId:target:selector:, 164
 - getObjectOfClass:objectId:, 164
 - getObjectOfClass:objectId:error:, 165
 - getObjectWithId:, 165
 - getObjectWithId:error:, 165
 - getUserObjectWithId:, 166
 - getUserObjectWithId:error:, 166
 - hasCachedResult, 166
 - ignoreACLs, 166
 - includeKey:, 166
 - initWithClassName:, 168
 - limit, 177
 - maxCacheAge, 177
 - orQueryWithSubqueries:, 169
 - orderByAscending:, 168
 - orderByDescending:, 168
 - orderBySortDescriptor:, 168
 - orderBySortDescriptors:, 168
 - PARSE_DEPRECATED, 169
 - parseClassName, 177
 - queryWithClassName:, 169
 - queryWithClassName:predicate:, 169
 - selectKeys:, 170
 - skip, 177
 - trace, 178
 - whereKey:containedIn:, 170
 - whereKey:containsAllObjectsInArray:, 170
 - whereKey:containsString:, 170
 - whereKey:doesNotMatchKey:inQuery:, 171
 - whereKey:doesNotMatchQuery:, 171
 - whereKey:equalTo:, 171
 - whereKey:greaterThan:, 172
 - whereKey:greaterThanOrEqualTo:, 172
 - whereKey:hasPrefix:, 172
 - whereKey:hasSuffix:, 172
 - whereKey:lessThan:, 173
 - whereKey:lessThanOrEqualTo:, 173

- whereKey:matchesKey:inQuery:, 173
- whereKey:matchesQuery:, 173
- whereKey:matchesRegex:, 174
- whereKey:matchesRegex:modifiers:, 174
- whereKey:nearGeoPoint:, 174
- whereKey:nearGeoPoint:withinKilometers:, 175
- whereKey:nearGeoPoint:withinMiles:, 175
- whereKey:nearGeoPoint:withinRadians:, 175
- whereKey:notContainedIn:, 176
- whereKey:notEqualTo:, 176
- whereKey:withinGeoBoxFromSouthwest:to↔Northeast:, 176
- whereKeyDoesNotExist:, 176
- whereKeyExists:, 177
- PFQueryCollectionView, 178
 - clear, 179
 - collectionView:cellForItemAtIndexPath:object:, 179
 - collectionViewReusableViewForNextPageAction:, 180
 - initWithClassName:, 180
 - initWithCollectionViewLayout:className:, 180
 - loadNextPage, 180
 - loadObjects, 180
 - loadObjects:clear:, 181
 - loading, 181
 - loadingViewEnabled, 182
 - NS_REQUIRES_SUPER, 181
 - objectAtIndexPath:, 181
 - objects, 182
 - objectsDidLoad:, 181
 - objectsPerPage, 182
 - paginationEnabled, 182
 - parseClassName, 182
 - pullToRefreshEnabled, 182
 - queryForCollection, 181
- PFQueryTableViewController, 182
 - clear, 184
 - imageKey, 186
 - initWithClassName:, 184
 - initWithStyle:className:, 184
 - loadNextPage, 184
 - loadObjects, 184
 - loadObjects:clear:, 184
 - loading, 186
 - loadingViewEnabled, 186
 - objectAtIndexPath:, 185
 - objects, 186
 - objectsDidLoad:, 185
 - objectsPerPage, 186
 - objectsWillLoad, 185
 - paginationEnabled, 186
 - parseClassName, 187
 - placeholderImage, 187
 - pullToRefreshEnabled, 187
 - queryForTable, 185
 - tableView:cellForNextPageAtIndexPath:, 185
 - tableView:cellForRowAtIndexPath:object:, 185
 - textKey, 187
- PFRelation, 187
 - addObject:, 188
 - query, 188
 - removeObject:, 188
 - targetClass, 188
- PFRole, 188
 - initWithName:, 189
 - initWithName:acl:, 189
 - name, 190
 - roleWithName:, 190
 - roleWithName:acl:, 190
 - roles, 190
 - users, 190
- PFSession, 191
 - getCurrentSessionInBackground, 191
 - getCurrentSessionInBackgroundWithBlock:, 191
 - sessionToken, 192
- PFSession.h
 - PFSessionResultBlock, 244
- PFSessionResultBlock
 - PFSession.h, 244
- PFSetResultBlock
 - PFConstants.h, 235
- PFSignUpCancelNotification
 - PFSignUpViewController.h, 252
- PFSignUpFailureNotification
 - PFSignUpViewController.h, 252
- PFSignUpSuccessNotification
 - PFSignUpViewController.h, 252
- PFSignUpView, 192
 - additionalField, 193
 - dismissButton, 193
 - emailAsUsername, 193
 - emailField, 193
 - fields, 193
 - initWithFields:, 193
 - logo, 193
 - passwordField, 194
 - presentingViewController, 194
 - signUpButton, 194
 - usernameField, 194
- PFSignUpView.h
 - NS_OPTIONS, 251
- PFSignUpViewController, 194
 - delegate, 195
 - emailAsUsername, 195
 - fields, 195
 - minPasswordLength, 195
 - signUpView, 195
- PFSignUpViewController.h
 - PFSignUpCancelNotification, 252
 - PFSignUpFailureNotification, 252
 - PFSignUpSuccessNotification, 252
- PFSignUpViewControllerDelegate-p
 - signUpViewController:didFailToSignUpWithError:, 196
 - signUpViewController:didSignUpUser:, 197
 - signUpViewController:shouldBeginSignUp:, 197

- signUpViewControllerDidCancelSignUp:, 197
- PFStringResultBlock
 - PFCConstants.h, 235
- PFSubclassing-p
 - object, 198
 - objectWithoutDataWithObjectId:, 198
 - parseClassName, 198
 - query, 198
 - queryWithPredicate:, 198
 - registerSubclass, 199
- PFTableViewCell, 199
 - imageView, 200
- PFTextField, 200
 - initWithFrame:separatorStyle:, 200
 - separatorStyle, 201
 - UI_APPEARANCE_SELECTOR, 201
- PFTextField.h
 - NS_OPTIONS, 253
- PFTwitterUtils, 201
 - initWithConsumerKey:consumerSecret:, 202
 - isLinkedWithUser:, 202
 - linkUser:, 203
 - linkUser:block:, 203
 - linkUser:target:selector:, 203
 - linkUser:twitterId:screenName:authToken:authTokenSecret:block:, 203
 - linkUser:twitterId:screenName:authToken:authTokenSecret:target:selector:, 203
 - linkUserInBackground:, 203
 - linkUserInBackground:twitterId:screenName:authToken:authTokenSecret:, 204
 - logInInBackground, 204
 - logInWithBlock:, 204
 - logInWithTarget:selector:, 204
 - logInWithTwitterId:screenName:authToken:authTokenSecret:block:, 205
 - logInWithTwitterId:screenName:authToken:authTokenSecret:target:selector:, 205
 - logInWithTwitterIdInBackground:screenName:authToken:authTokenSecret:, 205
 - twitter, 205
 - unlinkUser:, 205
 - unlinkUser:error:, 206
 - unlinkUserInBackground:, 206
 - unlinkUserInBackground:block:, 206
 - unlinkUserInBackground:target:selector:, 206
- PFUI_ASSUME_NONNULL_BEGIN
 - ParseUIConstants.h, 247
- PFUI_ASSUME_NONNULL_END
 - ParseUIConstants.h, 247
- PFUI_NULL_UNSPECIFIED
 - ParseUIConstants.h, 247
- PFUI_NULLABLE
 - ParseUIConstants.h, 247
- PFUI_NULLABLE_PROPERTY
 - ParseUIConstants.h, 247
- PFUI_NULLABLE_S
 - ParseUIConstants.h, 247
- PFUser, 207
 - become:, 208
 - become:error:, 208
 - becomeInBackground:, 209
 - becomeInBackground:block:, 209
 - becomeInBackground:target:selector:, 209
 - currentUser, 209
 - email, 215
 - enableAutomaticUser, 209
 - enableRevocableSessionInBackground, 210
 - enableRevocableSessionInBackgroundWithBlock:, 210
 - isAuthenticated, 210
 - isNew, 215
 - logInWithUsername:password:, 210
 - logInWithUsername:password:error:, 211
 - logInWithUsernameInBackground:password:, 211
 - logInWithUsernameInBackground:password:block:, 211
 - logInWithUsernameInBackground:password:target:selector:, 212
 - logOut, 212
 - logOutInBackground, 212
 - logOutInBackgroundWithBlock:, 212
 - password, 215
 - requestPasswordResetForEmail:, 212
 - requestPasswordResetForEmail:error:, 212
 - requestPasswordResetForEmailInBackground:, 213
 - requestPasswordResetForEmailInBackground:block:, 213
 - requestPasswordResetForEmailInBackground:target:selector:, 213
 - sessionToken, 215
 - signUp, 213
 - signUp:, 213
 - signUpInBackground, 214
 - signUpInBackgroundWithBlock:, 214
 - signUpInBackgroundWithTarget:selector:, 214
 - user, 214
 - username, 215
- PFUser.h
 - PFUserLogoutResultBlock, 245
 - PFUserSessionUpgradeResultBlock, 245
- PFUserLogoutResultBlock
 - PFUser.h, 245
- PFUserResultBlock
 - PFCConstants.h, 235
- PFUserSessionUpgradeResultBlock
 - PFUser.h, 245
- paginationEnabled
 - PFQueryCollectionViewController, 182
 - PFQueryTableViewController, 186
- Parse, 48
 - enableLocalDatastore, 49
 - getApplicationId, 49
 - getClientKey, 49
 - isLocalDatastoreEnabled, 49

- logLevel, [49](#)
 - setApplicationId:clientKey:, [50](#)
 - setLogLevel:, [50](#)
- parseClassName
 - PFObjct, [138](#)
 - PFQuery, [177](#)
 - PFQueryCollectionViewController, [182](#)
 - PFQueryTableViewController, [187](#)
 - PFSubclassing-p, [198](#)
- ParseCrashReporting, [50](#)
 - enable, [51](#)
 - isCrashReportingEnabled, [51](#)
- ParseUIConstants.h
 - PARSE_UI_DEPRECATED, [247](#)
 - PFUI_ASSUME_NONNULL_BEGIN, [247](#)
 - PFUI_ASSUME_NONNULL_END, [247](#)
 - PFUI_NULL_UNSPECIFIED, [247](#)
 - PFUI_NULLABLE, [247](#)
 - PFUI_NULLABLE_PROPERTY, [247](#)
 - PFUI_NULLABLE_S, [247](#)
- password
 - PFUser, [215](#)
- passwordField
 - LoginViewController, [47](#)
 - PFLogInView, [99](#)
 - PFSignUpView, [194](#)
- passwordForgottenButton
 - PFLogInView, [99](#)
- pin
 - PFObjct, [119](#)
- pin:
 - PFObjct, [120](#)
- pinAll:
 - PFObjct, [120](#)
- pinAll:error:
 - PFObjct, [121](#)
- pinAll:withName:
 - PFObjct, [121](#)
- pinAll:withName:error:
 - PFObjct, [121](#)
- pinAllInBackground:
 - PFObjct, [122](#)
- pinAllInBackground:block:
 - PFObjct, [122](#)
- pinAllInBackground:withName:
 - PFObjct, [123](#)
- pinAllInBackground:withName:block:
 - PFObjct, [123](#)
- pinInBackground
 - PFObjct, [123](#)
- pinInBackgroundWithBlock:
 - PFObjct, [124](#)
- pinInBackgroundWithName:
 - PFObjct, [124](#)
- pinInBackgroundWithName:block:
 - PFObjct, [124](#)
- pinWithName:
 - PFObjct, [125](#)
- pinWithName:error:
 - PFObjct, [125](#)
- PlaceObstacles1
 - Game, [41](#)
- PlaceObstacles2
 - Game, [41](#)
- placeholderImage
 - PFQueryTableViewController, [187](#)
- presentingViewController
 - PFLogInView, [99](#)
 - PFSignUpView, [194](#)
- priceLabel
 - PFPurchaseTableViewCell, [145](#)
- productIdentifier
 - PFProduct, [141](#)
- ProfileViewController, [215](#)
 - deleteButton:, [216](#)
 - emailLabel, [216](#)
 - highScoreLabel, [216](#)
 - logoutButton:, [216](#)
 - saveChanges:, [216](#)
 - userNameField, [216](#)
 - userNameLabel, [216](#)
- ProfileViewController(), [217](#)
- progressView
 - PFPurchaseTableViewCell, [145](#)
- pullToRefreshEnabled
 - PFQueryCollectionViewController, [182](#)
 - PFQueryTableViewController, [187](#)
- push
 - PFPush, [149](#)
- query
 - PFInstallation, [95](#)
 - PFObjct, [126](#)
 - PFObjct(Subclass), [139](#)
 - PFRelation, [188](#)
 - PFSubclassing-p, [198](#)
- queryForCollection
 - PFQueryCollectionViewController, [181](#)
- queryForTable
 - PFQueryTableViewController, [185](#)
- queryWithClassName:
 - PFQuery, [169](#)
- queryWithClassName:predicate:
 - PFQuery, [169](#)
- queryWithPredicate:
 - PFObjct, [126](#)
 - PFObjct(Subclass), [139](#)
 - PFSubclassing-p, [198](#)
- randomLongObstacle1Placement
 - Game.h, [226](#)
- randomLongObstacle2Placement
 - Game.h, [226](#)
- randomLongObstacle3Placement
 - Game.h, [226](#)
- randomLongObstacle4Placement
 - Game.h, [226](#)

- reEnterPasswordField
 - LoginViewController, [48](#)
- reauthorizeUser:withPublishPermissions:audience↔
 - :block:
 - PFFacebookUtils, [80](#)
- reauthorizeUser:withPublishPermissions:audience↔
 - :target:selector:
 - PFFacebookUtils, [80](#)
- reauthorizeUserInBackground:withPublishPermissions↔
 - :audience:
 - PFFacebookUtils, [80](#)
- recoverySpeed
 - Game.m, [228](#)
- refererAppLink
 - BFApplinkReturnToRefererView, [24](#)
- registerAction:
 - LoginViewController, [47](#)
- registerSubclass
 - PFObjct, [126](#)
 - PFObjct(Subclass), [140](#)
 - PFSubclassing-p, [199](#)
- relationForKey:
 - PFObjct, [126](#)
- relationforKey:
 - PFObjct, [127](#)
- removeFromNavController
 - BFApplinkReturnToRefererController, [22](#)
- removeObject:
 - PFRelation, [188](#)
- removeObject:forKey:
 - PFObjct, [127](#)
- removeObjectForKey:
 - PFObjct, [127](#)
- removeObjectsInArray:forKey:
 - PFObjct, [127](#)
- requestPasswordResetForEmail:
 - PFUser, [212](#)
- requestPasswordResetForEmail:error:
 - PFUser, [212](#)
- requestPasswordResetForEmailInBackground:
 - PFUser, [213](#)
- requestPasswordResetForEmailInBackground:block:
 - PFUser, [213](#)
- requestPasswordResetForEmailInBackground:target↔
 - :selector:
 - PFUser, [213](#)
- resolveAppLinkInBackground:
 - BFApplinkNavigation, [18](#)
- resolveAppLinkInBackground:resolver:
 - BFApplinkNavigation, [18](#)
- restore
 - PFPurchase, [144](#)
- result
 - BFTask, [32](#)
- returnToRefererController:didNavigateToAppLink:type:
 - BFApplinkReturnToRefererControllerDelegate-p, [23](#)
- returnToRefererController:willNavigateToAppLink:
 - BFApplinkReturnToRefererControllerDelegate-p, [23](#)
- returnToRefererViewDidTapInsideCloseButton:
 - BFApplinkReturnToRefererViewDelegate-p, [25](#)
- returnToRefererViewDidTapInsideLink:link:
 - BFApplinkReturnToRefererViewDelegate-p, [25](#)
- Right
 - Game.h, [226](#)
- RightLiftInside:
 - Game, [41](#)
- RightLiftOutside:
 - Game, [41](#)
- RightPress:
 - Game, [42](#)
- roleWithName:
 - PFRole, [190](#)
- roleWithName:acl:
 - PFRole, [190](#)
- roles
 - PFRole, [190](#)
- save
 - PFFile, [88](#)
 - PFObjct, [127](#)
- save:
 - PFFile, [88](#)
 - PFObjct, [127](#)
- saveAll:
 - PFObjct, [128](#)
- saveAll:error:
 - PFObjct, [128](#)
- saveAllInBackground:
 - PFObjct, [128](#)
- saveAllInBackground:block:
 - PFObjct, [128](#)
- saveAllInBackground:target:selector:
 - PFObjct, [128](#)
- saveChanges:
 - ProfileViewController, [216](#)
- saveEventually
 - PFObjct, [129](#)
- saveEventually:
 - PFObjct, [129](#)
- saveInBackground
 - PFFile, [88](#)
 - PFObjct, [129](#)
- saveInBackgroundWithBlock:
 - PFFile, [88](#)
 - PFObjct, [129](#)
- saveInBackgroundWithBlock:progressBlock:
 - PFFile, [88](#)
- saveInBackgroundWithTarget:selector:
 - PFFile, [89](#)
 - PFObjct, [129](#)
- Score
 - Game, [42](#)
- score1
 - leaderBoardViewController, [44](#)
- score10

- leaderBoardViewController, [44](#)
- score2
 - leaderBoardViewController, [45](#)
- score3
 - leaderBoardViewController, [45](#)
- score4
 - leaderBoardViewController, [45](#)
- score5
 - leaderBoardViewController, [45](#)
- score6
 - leaderBoardViewController, [45](#)
- score7
 - leaderBoardViewController, [45](#)
- score8
 - leaderBoardViewController, [45](#)
- score9
 - leaderBoardViewController, [45](#)
- ScoreLabel
 - Game, [43](#)
- ScoreNumber
 - Game.h, [227](#)
- screenBottom
 - Game.m, [228](#)
- screenLeft
 - Game.m, [228](#)
- screenName
 - PF_Twitter, [53](#)
- screenRight
 - Game.m, [228](#)
- screenTop
 - Game.m, [228](#)
- selectKeys:
 - PFQuery, [170](#)
- sendPush:
 - PFPush, [149](#)
- sendPushDataToChannel:withData:error:
 - PFPush, [149](#)
- sendPushDataToChannelInBackground:withData:
 - PFPush, [149](#)
- sendPushDataToChannelInBackground:withData↔
 - :block:
 - PFPush, [149](#)
- sendPushDataToChannelInBackground:withData↔
 - :target:selector:
 - PFPush, [150](#)
- sendPushDataToQuery:withData:error:
 - PFPush, [150](#)
- sendPushDataToQueryInBackground:withData:
 - PFPush, [150](#)
- sendPushDataToQueryInBackground:withData:block:
 - PFPush, [150](#)
- sendPushInBackground
 - PFPush, [151](#)
- sendPushInBackgroundWithBlock:
 - PFPush, [151](#)
- sendPushInBackgroundWithTarget:selector:
 - PFPush, [151](#)
- sendPushMessageToChannel:withMessage:error:
 - PFPush, [151](#)
- sendPushMessageToChannelInBackground:with↔
 - Message:
 - PFPush, [151](#)
- sendPushMessageToChannelInBackground:with↔
 - Message:block:
 - PFPush, [151](#)
- sendPushMessageToChannelInBackground:with↔
 - Message:target:selector:
 - PFPush, [153](#)
- sendPushMessageToQuery:withMessage:error:
 - PFPush, [153](#)
- sendPushMessageToQueryInBackground:with↔
 - Message:
 - PFPush, [153](#)
- sendPushMessageToQueryInBackground:with↔
 - Message:block:
 - PFPush, [153](#)
- separatorStyle
 - PFTextField, [201](#)
- session
 - PFFacebookUtils, [80](#)
- sessionToken
 - PFSession, [192](#)
 - PFUser, [215](#)
- setApplicationId:clientKey:
 - Parse, [50](#)
- setChannel:
 - PFPush, [153](#)
- setChannels:
 - PFPush, [154](#)
- setData:
 - PFPush, [154](#)
- setDefaultACL:withAccessForCurrentUser:
 - PFACL, [57](#)
- setDefaultResolver:
 - BFAAppLinkNavigation, [18](#)
- setDeviceTokenFromData:
 - PFInstallation, [95](#)
- setError:
 - BFTaskCompletionSource, [33](#)
- setException:
 - BFTaskCompletionSource, [34](#)
- setFacebookLoginBehavior:
 - PFFacebookUtils, [81](#)
- setLogLevel:
 - Parse, [50](#)
- setMessage:
 - PFPush, [154](#)
- setObject:forKey:
 - PFObject, [130](#)
- setObject:forKeyedSubscript:
 - PFObject, [131](#)
- setPublicReadAccess:
 - PFACL, [57](#)
- setPublicWriteAccess:
 - PFACL, [57](#)
- setPushToAndroid:

- PFPush, 154
- setPushToIOS:
 - PFPush, 154
- setQuery:
 - PFPush, 155
- setReadAccess:forRole:
 - PFACL, 58
- setReadAccess:forRoleWithName:
 - PFACL, 58
- setReadAccess:forUser:
 - PFACL, 58
- setReadAccess:forUserId:
 - PFACL, 58
- setResult:
 - BFTaskCompletionSource, 34
- setWriteAccess:forRole:
 - PFACL, 58
- setWriteAccess:forRoleWithName:
 - PFACL, 58
- setWriteAccess:forUser:
 - PFACL, 59
- setWriteAccess:forUserId:
 - PFACL, 59
- sharedInstance
 - BFWebViewAppLinkResolver, 37
- sharedManager
 - PFNetworkActivityIndicatorManager, 103
- showViewForRefererAppLink:
 - BFAppLinkReturnToRefererController, 22
- showViewForRefererURL:
 - BFAppLinkReturnToRefererController, 22
- signRequest:
 - PF_Twitter, 52
- signUp
 - PFUser, 213
- signUp:
 - PFUser, 213
- signUpButton
 - PFLogInView, 99
 - PFSignUpView, 194
- signUpController
 - PFLogInViewController, 101
- signUpInBackground
 - PFUser, 214
- signUpInBackgroundWithBlock:
 - PFUser, 214
- signUpInBackgroundWithTarget:selector:
 - PFUser, 214
- signUpView
 - PFSignUpViewController, 195
- signUpViewController:didFailToSignUpWithError:
 - PFSignUpViewControllerDelegate-p, 196
- signUpViewController:didSignUpUser:
 - PFSignUpViewControllerDelegate-p, 197
- signUpViewController:shouldBeginSignUp:
 - PFSignUpViewControllerDelegate-p, 197
- signUpViewControllerDidCancelSignUp:
 - PFSignUpViewControllerDelegate-p, 197
- skip
 - PFQuery, 177
- sourceURL
 - BFAppLink, 16
- StartGame:
 - Game, 42
- state
 - PFPurchaseTableViewCell, 145
- storeDeviceToken:
 - PFPush, 155
- subscribeToChannel:error:
 - PFPush, 155
- subscribeToChannelInBackground:
 - PFPush, 155
- subscribeToChannelInBackground:block:
 - PFPush, 155
- subscribeToChannelInBackground:target:selector:
 - PFPush, 156
- subtitle
 - PFProduct, 141
- tableView:cellForNextPageAtIndexPath:
 - PFQueryTableViewController, 185
- tableView:cellForRowAtIndexPath:object:
 - PFQueryTableViewController, 185
- TapLeft
 - Game, 43
- TapRight
 - Game, 43
- targetClass
 - PFRelation, 188
- targetQueryParameters
 - BFURL, 37
- targetURL
 - BFURL, 37
- targets
 - BFAppLink, 16
- task
 - BFTaskCompletionSource, 35
- taskCompletionSource
 - BFTaskCompletionSource, 34
- taskForCompletionOfAllTasks:
 - BFTask, 30
- taskForCompletionOfAllTasksWithResults:
 - BFTask, 31
- taskFromExecutor:withBlock:
 - BFTask, 31
- taskWithDelay:
 - BFTask, 31
- taskWithError:
 - BFTask, 31
- taskWithException:
 - BFTask, 31
- taskWithResult:
 - BFTask, 31
- textColor
 - BFAppLinkReturnToRefererView, 24
- textKey
 - PFQueryTableViewController, 187

- titleLabel
 - PFCollectionViewCell, [67](#)
- timeZone
 - PFIInstallation, [96](#)
- title
 - PFFProduct, [141](#)
- trace
 - PFFQuery, [178](#)
- trackAppOpenedWithLaunchOptions:
 - PFFAnalytics, [60](#)
- trackAppOpenedWithLaunchOptionsInBackground↔
 - :block:
 - PFFAnalytics, [60](#)
- trackAppOpenedWithRemoteNotificationPayload:
 - PFFAnalytics, [60](#)
- trackAppOpenedWithRemoteNotificationPayloadIn↔
 - Background:block:
 - PFFAnalytics, [60](#)
- trackEvent:
 - PFFAnalytics, [61](#)
- trackEvent:dimensions:
 - PFFAnalytics, [61](#)
- trackEventInBackground:block:
 - PFFAnalytics, [61](#)
- trackEventInBackground:dimensions:block:
 - PFFAnalytics, [62](#)
- trySetCancelled
 - BFTaskCompletionSource, [34](#)
- trySetError:
 - BFTaskCompletionSource, [34](#)
- trySetException:
 - BFTaskCompletionSource, [34](#)
- trySetResult:
 - BFTaskCompletionSource, [35](#)
- twitter
 - PFTwitterUtils, [205](#)
- twitterButton
 - PFFLoginView, [99](#)
- UI_APPEARANCE_SELECTOR
 - PFFTextField, [201](#)
- URL
 - BFAppLinkTarget, [26](#)
- URLWithInboundURL:sourceApplication:
 - BFURL, [36](#)
- URLWithURL:
 - BFURL, [36](#)
- unlinkUser:
 - PFFFacebookUtils, [81](#)
 - PFTwitterUtils, [205](#)
- unlinkUser:error:
 - PFFFacebookUtils, [81](#)
 - PFTwitterUtils, [206](#)
- unlinkUserInBackground:
 - PFFFacebookUtils, [81](#)
 - PFTwitterUtils, [206](#)
- unlinkUserInBackground:block:
 - PFFFacebookUtils, [82](#)
 - PFTwitterUtils, [206](#)
- unlinkUserInBackground:target:selector:
 - PFFacebookUtils, [82](#)
 - PFTwitterUtils, [206](#)
- unpin
 - PFFObject, [131](#)
- unpin:
 - PFFObject, [131](#)
- unpinAll:
 - PFFObject, [131](#)
- unpinAll:error:
 - PFFObject, [132](#)
- unpinAll:withName:
 - PFFObject, [132](#)
- unpinAll:withName:error:
 - PFFObject, [132](#)
- unpinAllInBackground:
 - PFFObject, [133](#)
- unpinAllInBackground:block:
 - PFFObject, [133](#)
- unpinAllInBackground:withName:
 - PFFObject, [133](#)
- unpinAllInBackground:withName:block:
 - PFFObject, [134](#)
- unpinAllObjects
 - PFFObject, [134](#)
- unpinAllObjects:
 - PFFObject, [134](#)
- unpinAllObjectsInBackground
 - PFFObject, [134](#)
- unpinAllObjectsInBackgroundWithBlock:
 - PFFObject, [135](#)
- unpinAllObjectsInBackgroundWithName:
 - PFFObject, [135](#)
- unpinAllObjectsInBackgroundWithName:block:
 - PFFObject, [135](#)
- unpinAllObjectsWithName:
 - PFFObject, [135](#)
- unpinAllObjectsWithName:error:
 - PFFObject, [136](#)
- unpinInBackground
 - PFFObject, [136](#)
- unpinInBackgroundWithBlock:
 - PFFObject, [136](#)
- unpinInBackgroundWithName:
 - PFFObject, [136](#)
- unpinInBackgroundWithName:block:
 - PFFObject, [137](#)
- unpinWithName:
 - PFFObject, [137](#)
- unpinWithName:error:
 - PFFObject, [137](#)
- unsubscribeFromChannel:error:
 - PFFPush, [156](#)
- unsubscribeFromChannelInBackground:
 - PFFPush, [156](#)
- unsubscribeFromChannelInBackground:block:
 - PFFPush, [156](#)

- unsubscribeFromChannelInBackground:target:selector↔
 - :
 - PFPush, 156
- updateFromObject:
 - PFCollectionViewCell, 67
- updatedAt
 - PFObjct, 138
- url
 - PFFile, 89
- user
 - PFUser, 214
- userId
 - PF_Twitter, 53
- userNameField
 - LoginViewController, 48
 - ProfileViewController, 216
- userNameLabel
 - ProfileViewController, 216
- username
 - PFUser, 215
- username1
 - leaderBoardViewController, 45
- username10
 - leaderBoardViewController, 45
- username2
 - leaderBoardViewController, 45
- username3
 - leaderBoardViewController, 45
- username4
 - leaderBoardViewController, 46
- username5
 - leaderBoardViewController, 46
- username6
 - leaderBoardViewController, 46
- username7
 - leaderBoardViewController, 46
- username8
 - leaderBoardViewController, 46
- username9
 - leaderBoardViewController, 46
- usernameField
 - PFLoginView, 99
 - PFSignUpView, 194
- users
 - PFRole, 190
- version
 - Bolts, 38
- view
 - BFApplinkReturnToRefererController, 22
- ViewController, 217
- ViewController(), 218
- waitUntilFinished
 - BFTask, 32
- webURL
 - BFApplink, 17
- WelcomeLabel
 - Game, 43
- whereKey:containedIn:
 - PFQuery, 170
- whereKey:containsAllObjectsInArray:
 - PFQuery, 170
- whereKey:containsString:
 - PFQuery, 170
- whereKey:doesNotMatchKey:inQuery:
 - PFQuery, 171
- whereKey:doesNotMatchQuery:
 - PFQuery, 171
- whereKey:equalTo:
 - PFQuery, 171
- whereKey:greaterThan:
 - PFQuery, 172
- whereKey:greaterThanOrEqualTo:
 - PFQuery, 172
- whereKey:hasPrefix:
 - PFQuery, 172
- whereKey:hasSuffix:
 - PFQuery, 172
- whereKey:lessThan:
 - PFQuery, 173
- whereKey:lessThanOrEqualTo:
 - PFQuery, 173
- whereKey:matchesKey:inQuery:
 - PFQuery, 173
- whereKey:matchesQuery:
 - PFQuery, 173
- whereKey:matchesRegex:
 - PFQuery, 174
- whereKey:matchesRegex:modifiers:
 - PFQuery, 174
- whereKey:nearGeoPoint:
 - PFQuery, 174
- whereKey:nearGeoPoint:withinKilometers:
 - PFQuery, 175
- whereKey:nearGeoPoint:withinMiles:
 - PFQuery, 175
- whereKey:nearGeoPoint:withinRadians:
 - PFQuery, 175
- whereKey:notContainedIn:
 - PFQuery, 176
- whereKey:notEqualTo:
 - PFQuery, 176
- whereKey:withinGeoBoxFromSouthwest:toNortheast:
 - PFQuery, 176
- whereKeyDoesNotExist:
 - PFQuery, 176
- whereKeyExists:
 - PFQuery, 177
- window
 - AppDelegate, 15