

DASHConfig

A DASHConfig instance is used to configure the DASH instance for use.

```
/// A unique identifier for the DASH Foundation (provided by DASH)
var appId: String
```

DASH

A DASH instance can be used as an app wide singleton (by using team). It is the main class for interacting with the SDK. Use this instance to handle deep link URLs and to instantiate a view controller to show the DASH interface.

```
/// Singleton use of DASH
static let team: DASH
```

```
/// Initializes DASH. Call this once before any other methods.
func start(with config: DASHConfig)
```

```
/// Sets the current user's email. Email is used to uniquely identify a user in the DASH system.
Email is cached locally by DASH for ease of use.
func setUserEmail(_ userEmail: String?)
```

```
/// Clears out local and cached user email data
func clearUserEmail()
```

```
/// Used to set the push device token for the current app. Set this with the data returned from
the remote notifications delegate method. This is used to send DASH outbid notifications on
the team's behalf. Set each time token changes. Token will be cached locally by DASH for ease
of use.
func setUserPushToken(with data: Data?)
```

```
/// Used to set the push device token for the current app if already in string format. This is used
to send DASH outbid notifications on the team's behalf. Set each time token changes. Token
will be cached locally by DASH for ease of use.
func setUserPushToken(with data: String)
```

```
/// Clears out local and cached push tokens in DASH
func clearPushToken()
```

```
/// Returns true if the notification passed in should be handled by DASH. If true, you should tell
DASH to handle the notification and present the DASH view controller.
func canHandleNotification(_ notification: UNNotification) -> Bool
```

/// Tells DASH to handle the notification. The next presentation of the DASH view controller will handle the notification. If DASH view controller is already presented, this will reload the interface to handle the notification. EX: If a DASH outbid notification is set here, the next presentation will navigate directly to the respective auction item.

func setNotificationData(from notification: UNNotification)

/// Returns whether DASH currently has data as result of a notification (and subsequently DASH view controller should be presented)

func hasNotificationData() -> Bool

/// Clears out the pending notification data provided by handleNotification()

func clearNotificationData()

/// Returns a DASH view controller for display. Can be pushed on a navigation stack as is or presented modally when embedded in a UINavigationController

func dashViewController() -> DASHViewController?

DASHViewController

A DASH View controller instance is the user interface for the DASH framework.

/// Delegate used to listen for error in the DASH view controller

var delegate: DASHViewControllerDelegate?

/// Refreshes the current page. If startFromBeginning is true, the interface is reloaded to the beginning state.

func reloadInterface(startFromBeginning: Bool = default)

DASHViewControllerDelegate

Delegate for the DASHViewController

/// Called when the DASH view controller encounters an error.

func dashViewController(_ dashViewController: DASHViewController, didFailWith error: DASH.Error)