M1 Use Cases Enablement (Phase 1) – App: Data Layer Requirements

This document will focus on the creation of a data layer object (within screens on the native app), as well as the declaration of variables to sit inside that data layer object.

Design

Within the Tealium native app initialisation script (i.e. the already implemented Tracking Helper Class, like shown below), we should declare the data layer variable key-value pairs to be collected.

```
class TealiumHelper {
    static let shared = TealiumHelper()
    let config = TealiumConfig(account: "ACCOUNT",
                               profile: "PROFILE",
                               environment: "ENVIRONMENT",
                               datasource: "DATASOURCE")
    var tealium: Tealium?

    private init() {

        // add necessary config options
        config.batchingEnabled = true
        config.logLevel = .debug

        // add desired Collectors - no need to include if want all
compiled collectors
        config.collectors = [Collectors.Lifecycle,
                             Collectors.Location,
                             Collectors.VisitorService]

        // add desired Dispatchers - no need to include if want compiled
dispatchers
        config.dispatchers = [Dispatchers.TagManagement,
                              Dispatchers.RemoteCommands]

        tealium = Tealium(config: config)

        tealium?.dataLayer.add(key: "emailToken", value: "<TOKENIZED-
EMAIL-VALUE> or null", expiry: .untilRestart)
        tealium?.dataLayer.add(key: "emailHash", value: "<HASHED-VALUE> or
null", expiry: .untilRestart)

    }

    public func start() {
        _ = TealiumHelper.shared
    }
    class func trackView(title: String, data: [String: Any]?) {
      let tealView = TealiumView(title, dataLayer: data)
      TealiumHelper.shared.tealium?.track(tealView)
    }

    class func trackEvent(title: String, data: [String: Any]?) {
      let tealEvent = TealiumEvent(title, dataLayer: data)
      TealiumHelper.shared.tealium?.track(tealEvent)
    }
}
```

Implementation

The following provides a list of the variables required (and required variable syntax) for ALL SCREENS on the native app platform:

`emailToken` - Email ID Token (Salesforce/Mulesoft-generated)

This field returns the tokenized output value of the user's email address. Tokenization is done in Mulesoft/Salesforce. The value to populate here should be the tokenized email address value only, without any prefixes or suffixes.

```
tealium?.dataLayer.add(key: "emailToken", value: "<TOKENIZED-EMAIL-
VALUE>", expiry: .untilRestart)
```

If the user is not logged in, populate the variable with the null value (string): `"null"`.

```
tealium?.dataLayer.add(key: "emailToken", value: "null", expiry:
.untilRestart)
```

`emailHash` - Email ID Hash

This field returns the hashed output value of the user's email address, using the same hashing algorithm as the Salesforce account (SHA-256). The value to populate here should be the SHA-256 hashed email address value only, without any prefixes or suffixes.

```
tealium?.dataLayer.add(key: "emailHash", value: "<HASHED-VALUE>", expiry:
.untilRestart)
```

If the user is not logged in, populate the variable with the null value (string): `"null"`.

```
tealium?.dataLayer.add(key: "emailHash", value: "null", expiry:
.untilRestart)
```

Resources

- Tealium docs - iOS Swift - Quick start
- Tealium docs - iOS Swift - Data Layer