Create a Banner Message

Displaying a banner involves creating an action trigger that activates an Omnichannel experience to return an appropriate JSON object. This object contains data that you can use to populate a banner message. You can use the Get Actions method to set up this trigger and obtain the JSON.

You can use the returned JSON data for multiple purposes. This example uses the JSON to display a banner message.

Creating the Experience

You must first create an Omnichannel experience within Monetate that uses an Omni JSON action type. Refer to Create an Omnichannel Experience for instructions, and Available Action Types for additional information on Omni JSON actions.

You can use the following JSON template for the required input:

```
{
 "meta":
 {
  "code": 200
 },
 "data":
 {
  "responses":
  [
    {
     "requestId": "sample-unique-request-id-ABC-123",
     "actions":
     [
       {
        "json":
        {
          "text": "Free Shipping on orders over $500",
          "color": "#FFFFFF",
          "style": "bold",
          "fontSize": 24,
          "background-color": "#697895"
        },
        "actionType": "monetate:action:OmnichannelJson",
        "component": "home_component_1",
        "actionId ": 5005228
       }
     ]
    }
  ]
 }
}
```

This is the JSON object that is returned on the trigger. Modify this template to suit your needs.

Triggering the Experience

The Omnichannel experience is triggered using the Get Actions method. This method is used when you want to satisfy an experience using a single event condition.

```
objPersonalization.getActions(context: <ContextEnum>, requestId: <String>, includeReporting: <B
```

Parameters:

- `context` is name of the event.
- `requestID` is the request ID for the API.
- `includeReporting` indicates whether the response will have impression reporting data.

- arrActionType is the type of actions you want to request. You can specify multiple actions in an array to handle.
- event is the data associated with the event.

Full Code Example

Complete code example blocks are listed below. This code triggers the configured Omni JSON experience and handles the experience by displaying a banner. Customize the example code as you see fit.

```
import MarqueeLabel
import UIKit
import monetate_ios_sdk

class CategoryViewController: UIViewController {

  @IBOutlet weak var bottomLabel: MarqueeLabel!
  @IBOutlet weak var bottomView: UIView!
  @IBOutlet weak var constraintHeightBottomView: NSLayoutConstraint!
  final var objPersonalization = Personalization(
    account: Account(
      instance: "p", domain: "localhost.org", name: "a-701b337c", shortname: "localhost"),
    user: User(deviceId: "62bd2e2d-213d-463f-83bb-12c0b2530a14"))

  override func viewDidLoad() {
    super.viewDidLoad()
    self.bottomView.isHidden = true
    self.constraintHeightBottomView.constant = 0
    bottomLabel.text = ""
  }

  override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)
    pageEvent()
  }

  func pageEvent() {
    objPersonalization.getActions(
      context: .PageView, requestId: "test_request_id", includeReporting: false,
      arrActionTypes: ["monetate:action:OmnichannelJson"],
      event: PageView(
        pageType: "Homepage", path: "n/a", url: "n/a", categories: [], breadcrumbs: [])
    ).on { (res) in
      if res.status == 200 {
        let data = JSON(res.data)
        print(data)
        self.handleAction(res: res)
      } else {
      }
    }
  }

  fileprivate func handleAction(res: APIResponse) {
    let data = JSON(res.data)
    for item in data["data"]["responses"].arrayValue {
      bottomLabel.animationCurve = .linear
      bottomLabel.speed = .duration(5)
      bottomLabel.fadeLength = 15.0

      if item["requestId"].string == res.requestId {
        for oneaction in item["actions"].arrayValue {
          let component = oneaction["component"].string ?? ""
          if component.lowercased() == "footer" {
            if let json = oneaction["json"].dictionary {
              print("final dict \(json)")
              if let text = json["text"]?.string {
                self.bottomView.isHidden = false
                self.constraintHeightBottomView.constant = 60
                bottomLabel.text = text
                if let fontStyle = json["style"]?.string, let fontSize = json["fontSize"]?.double {
                  if fontStyle == "bold" {
                    bottomLabel.font = UIFont(name: "Roboto-Bold", size: fontSize)
                  } else if fontStyle == "normal" {
                    bottomLabel.font = UIFont(name: "Roboto-Regular", size: fontSize)
                  } else if fontStyle == "italic" {
                    bottomLabel.font = UIFont(name: "Roboto-Italic", size: fontSize)
                  }
                }
                if let color = json["color"]?.string {
                  bottomLabel.textColor = MiscClass.hexStringToUIColor(hex: color)
                }
                if let backgroundColor = json["background-color"]?.string {
                  bottomView.backgroundColor = MiscClass.hexStringToUIColor(hex: backgroundColor)
                  return
                }
              }
            }
          }
        }
      }
    }
  }
}
```