TEALIUM WEB PLATFORM INTEGRATIONS

Hybris Integration

Tealium Addon for Hybris Setup Guide

2 May 2016

© 2016 Tealium, Inc. All rights reserved.

Prerequisites

- A Tealium iQ account
- hybris v5.0.1+ extracted and built, including the B2C Commerce Accelerator.

Definitions

- Universal Data Object (UDO) = The Data Layer object of key/value pairs. This object is generated in the page as the global JavaScript object utag_data. The Tealium utag.js uses utag_data as the base Data Layer for your tags
- Tealium Account = The account assigned to your purchase of Tealium iQ
- Tealium Profile = The specific web property name for your website

Installing Addon

1. Github: https://github.com/Tealium/integration-hybris
2. Place the "tealiumiqaddon" directory into ${HYBRIS_BIN}/custom. This directory is in the /hybris/bin/custom/ folder in this repo.
3. Add <extension dir="${HYBRIS_BIN}/custom/tealiumiqaddon"/\> to your config/localextensions.xml.
4. Add tealiumiqaddon to yacceleratorstorefront by using:
   - sudo ant addoninstall -Daddonnames="tealiumiqaddon" -DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"
5. Update Files:
   - ○ ${HYBRIS_BIN}/ext-template/yacceleratorstorefront/web/webroot/WEB-INF/tags/desktop/template/master.tag by adding:
     1. <%@ taglib prefix="tealiumiqaddon" tagdir="/WEB-INF/tags/addons/tealiumiqaddon/shared/analytics" %> at the top of the file
     2. <tealiumiqaddon:sync/> after the <head> tag
     3. <tealiumiqaddon:tealium/> after the <body> tag
   - ○ ${HYBRIS_BIN}/custom/tealiumiqaddon/project.properties.template by changing:

1. **tealiumiqaddon.account**, **tealiumiqaddon.profile**, and **tealiumiqaddon.target** to your tealiumIQ account specific information.
2. Modify tealiumiqaddon.utagSyncEnabled = 1 if you want to enable **utag.sync.js** injection into the <head> of your HTML page. This feature is optional and disabled by default.

Adding Custom Data

By default, the add on provides a comprehensive list of standard e-commerce variables. If these default values are not sufficient for your installation, you can extend the default page types as well as create new custom page types.

1. Create a new class implementing the interface com.tealium.dataconnector.hybris.HybrisDataConverter.HybrisCustomDataConverter and implement all methods of the interface.

```
package com.tealium.dataconnector.hybris;
import com.tealium.dataconnector.hybris.HybrisDataConverter.HybrisCustomDataConverter;
import com.tealium.dataconnector.hybris.HybrisDataConverter.HybrisCustomPageTypeCustomData;
import com.tealium.util.udohelpers.UDO;
import com.tealium.util.udohelpers.exceptions.UDOUpdateException;

public class TealiumCustomData implements HybrisCustomDataConverter {
    //... add unimplemented methods of interface.
}
```

2. Make sure **if you are not adding values** to any of the methods of the interface, return the udo object.

```
@Override
public UDO homePage(UDO udo) {
    return udo;
}
```

3. To add or modify values to a default page, add to the override method for that page:

```
@Override
public UDO searchPage(UDO udo) {
    try {
        udo.setValue("page_name", "new search page name");
        udo.setValue("custom_key", "custom_value");
    } catch (UDOUpdateException e) {
        e.printStackTrace();
    }
    return udo;
}
```

4. To add a new page type with values, add a static variable to your class and add new pages to the addCustomPages method

```
public class TealiumCustomData implements HybrisCustomDataConverter {
    private static Map<String, HybrisCustomPageTypeCustomData> customPagesMap;

    @Override
    public Map<String, HybrisCustomPageTypeCustomData> getHybrisCustomPageTypes() {
        return customPagesMap;
    }
    //... other methods
    @Override
    public void addCustomPages() {
        if (customPagesMap == null){
            customPagesMap = new HashMap<>();
        }
        customPagesMap.put("custom_one", new HybrisCustomPageTypeCustomData(){

            @Override
            public UDO getCustomDataUdo(UDO udo) {
                try {
                    udo.setValue("page_name", "custom page 1");
                    udo.setValue("custom_page1_key", "custom value");
                } catch (UDOUpdateException e) {
                    e.printStackTrace();
                }
                return udo;
            }

        });

        customPagesMap.put("custom_two", new HybrisCustomPageTypeCustomData(){

            @Override
            public UDO getCustomDataUdo(UDO udo) {
                try {
                    udo.setValue("page_name", "custom page 2");
                    udo.setValue("custom_page2_key", "custom value");
                } catch (UDOUpdateException e) {
                    e.printStackTrace();
                }
                return udo;
            }

        });
    }
}
```

5. Modify: ${HYBRIS_BIN}/custom/tealiumiqaddonweb/webroot/WEB-INF/tags/addons/tealiumiqaddon/shared/analytics/data.tag
    1. Replace CLASS_NAME with your class name (i.e. TealiumCustomData)
    2. Add your new class: <%@ tag import="com.tealium.dataconnector.hybris.**CLASS_NAME**" %>
    3. Register the new class with HybrisDataConverter: <% HybrisDataConverter.registerCustomDataClass("ID", new **CLASS_NAME**()); %>

Finish

Rebuild and restart hybris.

```
sudo ant all
sudo ./hybrisserver.sh
```

Default data sources

All Pages

page_name

Contains a user-friendly page name

site_region

Includes values for region or localized version, e.g. en_US

site_currency

Contains the currency shown on site, e.g. USD

page_type

Contains a user-friendly page type, e.g. cart page

Search Page

search_results

Contains the number of results returned with a search query

search_keyword

Contains the search query conducted by user

Category Pages

page_category_name

Contains a user-friendly category name, e.g. appliances

Product Page

product_id

Contains product ID(s) - multiple values should be comma-separated

product_sku

Contains product SKU(s) - multiple values should be comma-separated

product_name

Contains product name(s) - multiple values should be comma-separated

product_brand

Contains product brand(s) - multiple values should be comma-separated

product_category

Contains product category(s) - multiple values should be comma-separated

product_subcategory

Contains product subcategory(s) - multiple values should be comma-separated

product_unit_price

Contains product unit price(s) - multiple values should be comma-separated

Cart Page

product_id

Contains product ID(s) - multiple values should be comma-separated

product_sku

Contains product SKU(s) - multiple values should be comma-separated

product_name

Contains product name(s) - multiple values should be comma-separated product_brand

Contains product brand(s) - multiple values should be comma-separated

product_category

Contains product category(s) - multiple values should be comma-separated

product_subcategory

Contains product subcategory(s) - multiple values should be comma-separated

product_unit_price

Contains product unit price(s) - multiple values should be comma-separated

product_quantity

Contains product quantity(s) - multiple values should be comma-separated

Order Confirmation

order_id

Contains the order or transaction ID

order_subtotal

Contains the order payment type

order_payment_type

Contains the order payment type

order_total

Contains the order total value

order_discount

Contains the order discount (if any)

order_shipping

Contains the order shipping value

order_tax

Contains the order tax amount

order_currency

Contains the currency associated with the transaction, e.g. USD'

order_coupon_code

Contains the order coupon code

order_type

Contains the order/cart

product_id

Contains product ID(s) - multiple values should be comma-separated

product_sku

Contains product SKU(s) - multiple values should be comma-separated

product_name

Contains product name(s) - multiple values should be comma-separated

product_brand

Contains product brand(s) - multiple values should be comma-separated

product_category

Contains product category(s) - multiple values should be comma-separated

product_subcategory

Contains product subcategory(s) - multiple values should be comma-separated product_unit_price

Contains product unit price(s) - multiple values should be comma-separated

product_quantity

Contains product quantity(s) - multiple values should be comma-separated

customer_email

Contains the customer email

Customer Info Page

customer_name

Contains the customer username

customer_email

Contains the customer email

gender

Contains the customer gender based on salutation