

Mopub

You can use Hyperadx as a Network in **Mopub's** Mediation platform.

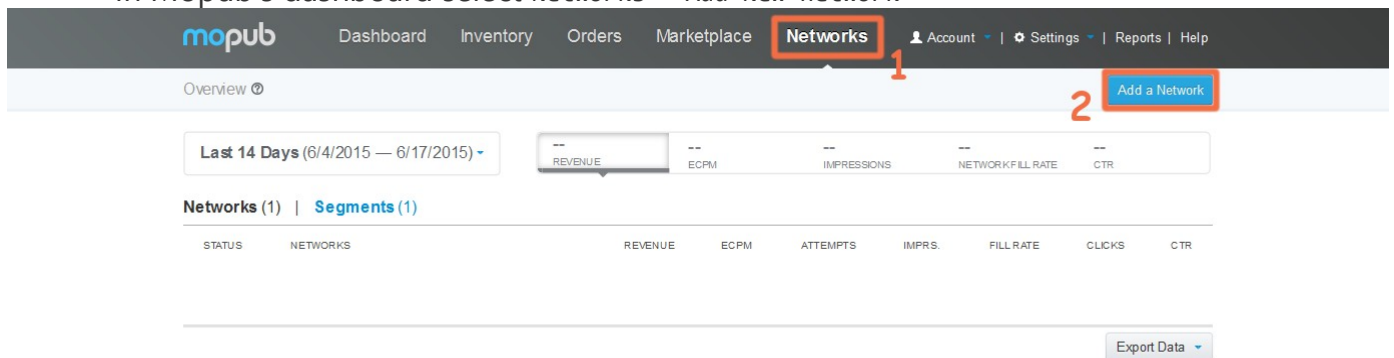
1. Setup SDKs

- Integrate with Mopub SDK (<https://github.com/mopub/mopub-android-sdk/wiki/Getting-Started>)
 - Install Hyperadx SDK
- More info how to install Hyperadx SDK on [Integration and API documentation](#)

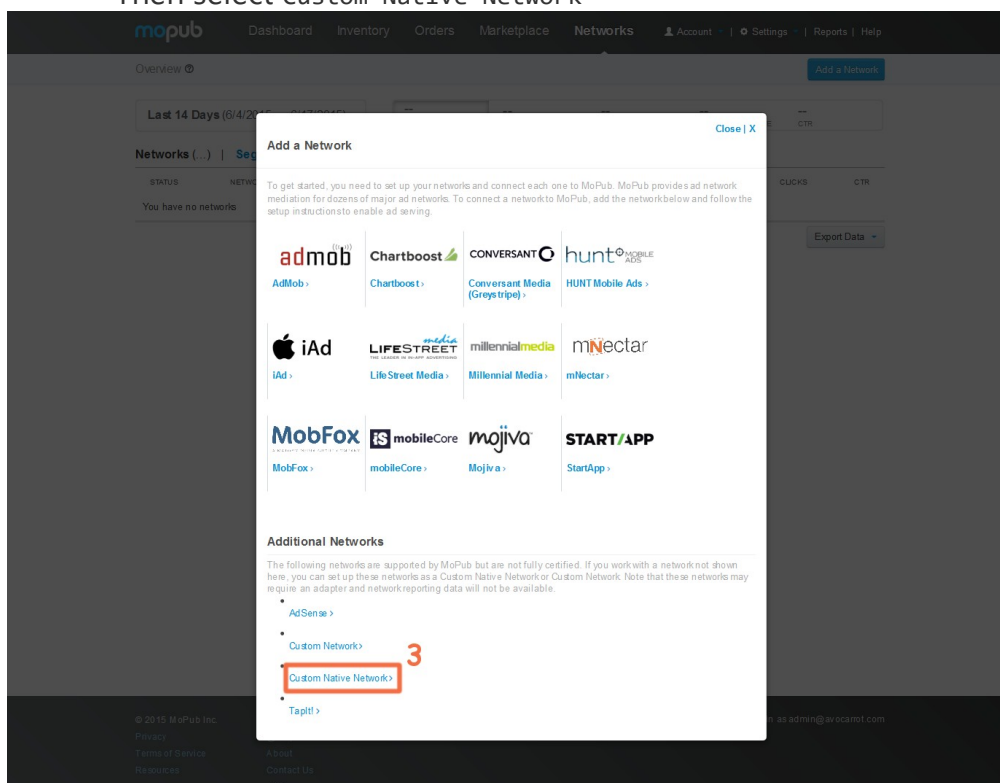
2. Setup Mopub Dashboard

Create an "Hyperadx" Network in Mopub's dashboard and connect it to your Ad Units.

- In Mopub's dashboard select Networks > Add New network



- Then select Custom Native Network




- Complete the fields accordingly to the Ad Unit that you want to use

Special Instructions

- Custom Native Networks require the development of a Custom Event Class
- You must enter a Custom Class to enable ad serving for this type of network. Custom Event Class Data can be sent down in JSON format. Note: Custom Native Networks using a Custom Method implementation will not be supported in future MoPub SDKs.

Title

Set Up Your Inventory Add a Filter

APP AND ADUNITS	PLATFORM	AD FORMAT	CUSTOM EVENT METHOD	CUSTOM EVENT CLASS	CUSTOM EVENT CLASS DATA
 Test	Android				
Fullscreen Ad	Fullscreen	No Custom Content Entered	<code>com.mopub.mobileads.HyperadxInterstitialMopub</code>	<code>{"PLACEMENT": "5b3QbMRQ"}</code>	

Custom Event Class

`com.mopub.mobileads.HyperadxInterstitialMopub`

Custom Event Class Data

`{"PLACEMENT": "<YOUR PLACEMENT>"}`

You can use the test placement "5b3QbMRQ"

2. Add adapter in your project

Create package "com.mopub.mobileads" in your project and put this class in there:

HyperadxInterstitialMopub.java

```
package com.mopub.mobileads;

import android.content.Context;
import android.util.Log;
import com.hyperadx.lib.sdk.interstitialads.HADInterstitialAd;
import com.hyperadx.lib.sdk.interstitialads.InterstitialAdListener;
import java.util.Map;
public class HyperadxInterstitialMopub extends CustomEventInterstitial {
    private static final String PLACEMENT_KEY = "PLACEMENT";
    private HADInterstitialAd interstitialAd;
    private com.hyperadx.lib.sdk.interstitialads.Ad iAd = null;
    CustomEventInterstitialListener customEventInterstitialListener;
    @Override
    protected void loadInterstitial(final Context context, final
CustomEventInterstitialListener customEventInterstitialListener, Map<String,
Object> localExtras, Map<String, String> serverExtras) {
        final String placement;
        final String appId;
        if (serverExtras != null && serverExtras.containsKey(PLACEMENT_KEY)) {
            placement = serverExtras.get(PLACEMENT_KEY);
        } else {

customEventInterstitialListener.onInterstitialFailed(MoPubErrorCode.ADAPTER_CONF
IGURATION_ERROR);
            return;
        }
        interstitialAd = new HADInterstitialAd(context, placement);
//Interstitial AD constructor
        interstitialAd.setAdListener(new InterstitialAdListener() { // Set
Listener
            @Override
            public void onAdLoaded(com.hyperadx.lib.sdk.interstitialads.Ad ad) {
// Called when AD is Loaded
                iAd = ad;
                // Toast.makeText(context, "Interstitial Ad loaded",
Toast.LENGTH_SHORT).show();
                customEventInterstitialListener.onInterstitialLoaded();
            }
            @Override
            public void onError(com.hyperadx.lib.sdk.interstitialads.Ad Ad,
String error) { // Called when load is fail
                // Toast.makeText(context, "Interstitial Ad failed to load
with error: " + error, Toast.LENGTH_SHORT).show();

customEventInterstitialListener.onInterstitialFailed(MoPubErrorCode.UNSPECIFIED)
;
            }
            @Override
            public void onInterstitialDisplayed() { // Called when Ad was
impressed
                // Toast.makeText(context, "Tracked Interstitial Ad
impression", Toast.LENGTH_SHORT).show();
                customEventInterstitialListener.onInterstitialShown();
            }
        }
    }
}
```

```

        @Override
        public void
onInterstitialDismissed(com.hyperadx.lib.sdk.interstitialads.Ad ad) { // Called
when Ad was dismissed by user
        //    Toast.makeText(context, "Interstitial Ad Dismissed",
Toast.LENGTH_SHORT).show();
        customEventInterstitialListener.onInterstitialDismissed();
    }
    @Override
    public void onAdClicked() { // Called when user click on AD
        //    Toast.makeText(context, "Tracked Interstitial Ad click",
Toast.LENGTH_SHORT).show();
        customEventInterstitialListener.onInterstitialClicked();
    }
    });
    this.customEventInterstitialListener = customEventInterstitialListener;
    interstitialAd.loadAd(); // Call to load AD
}
@Override
protected void showInterstitial() {
    if (iAd != null)
        HADInterstitialAd.show(iAd); // Call to show AD
    else
        Log.e("HADInterstitialMopub", "The Interstitial AD not ready yet.
Try again!");
}
@Override
protected void onInvalidate() {
}
}

```

This is your adapter. Now you can use Mopub as usual.