

- Introduction
- Authentication
- Available Widgets
- Mini Widgets
- Implementing Mobile Widgets
- Embedding a Desktop Widget
- Advanced Desktop Widget Usage
- PostMessage Events
- PostMessage alternative for WebView-only implementations
- Notifications
- Pulse
- Changelog

INTRODUCTION

i — Historically, the software outlined in this guide has been called “MoneyDesktop,” or “DMM”, but MX now uses the name MoneyMap. You’ll still find many references to “MoneyDesktop” throughout this guide, including in URLs, class names, etc.

MoneyMap is MX’s digital money management software. It consists of several *widgets*, each of which can be displayed alone or in conjunction with other widgets. This allows for a customizable and configurable product that fits the needs of both MX’s partners and their end users.

Widgets are designed to work seamlessly in either a mobile or desktop environment on top of a single codebase. Most widgets are built with a responsive design using the React Javascript framework.

Two small, lightweight *mini widgets* are also available for use in more restrictive situations; these have an independent codebase and are not responsive.

This guide will take you through the steps required to embed widgets and mini widgets into your mobile or desktop application.

AUTHENTICATION

All widgets are authenticated through user-specific, tokenized URLs provided by our [SSO API](#). These tokenized URLs expire after 10 minutes and are single-use. A new URL must be requested each time the page is rendered. This will include cases such as the user clicking refresh or the user navigating away from the page and then returning to it. Any time the page is re-loaded for any reason it must be done with a fresh widget URL.

w — Warning: Do not attempt to alter the URLs provided through the SSO API. Altering them can cause widgets to break.

AVAILABLE WIDGETS

The table below lists the widgets available to be embedded either individually or as part of the Master widget.

Strictly speaking, all widgets use a responsive design and will automatically adjust their look and functionality to accommodate the size of their container. *However*, widgets labeled as “available on mobile” have been optimized to look and work well when used in smaller, mobile-sized containers; those unavailable on mobile are still in this process of optimization.

i — In order to get accurate analytics data from MX systems, certain widgets must be called using [mobile-specific URLs](#) when used in mobile contexts such as WebViews. Analytics data on mobile vs. desktop visits is unavailable for widgets without a mobile-specific URL

Nevertheless, widgets will display properly even if the standard URL is used in a mobile context.

| Widget | Available on mobile? | Notes |
|----------------|----------------------|--|
| Accounts | Yes | For analytics purposes, a mobile-specific URL is available for this widget. |
| Budgets | Yes | For analytics purposes, a mobile-specific URL is available for this widget. |
| Cash Flow | Yes | |
| Connect | Yes | |
| Connections | Yes | |
| Debts | Yes | |
| Goals | Yes | |
| Help | Yes | Not all help material is available on mobile. |
| Investments | No | Investments is still in mobile development. |
| Master widget | Yes | The Master widget can display all or any combination of widgets; the mobile Master widget is limited only to widgets available on mobile. For analytics purposes, a mobile-specific URL is available for this widget. |
| Mini Budgets | Yes | This is a lightweight, smaller version of the Budgets widget. |
| Mini Net Worth | Yes | This is a lightweight, smaller version of the Net Worth widget. |

| | | |
|------------------------|-----|--|
| Mini spending | Yes | This is a lightweight, smaller version of the Spending widget. |
| Net Worth | Yes | |
| Notifications Settings | Yes | |
| Pulse | Yes | The pulse widget is only available when pulse functionality is enabled for the client. |
| Settings | Yes | |
| Spending | Yes | For analytics purposes, a mobile-specific URL is available for this widget. |
| Transactions | Yes | For analytics purposes, a mobile-specific URL is available for this widget. |
| Trends | Yes | |

Minimum Dimensions

MX's widgets implement a responsive design, meaning they will change in appearance when viewed at different widths. Page styles and layout are subject to change, and elements may become hidden or visible at different widths.

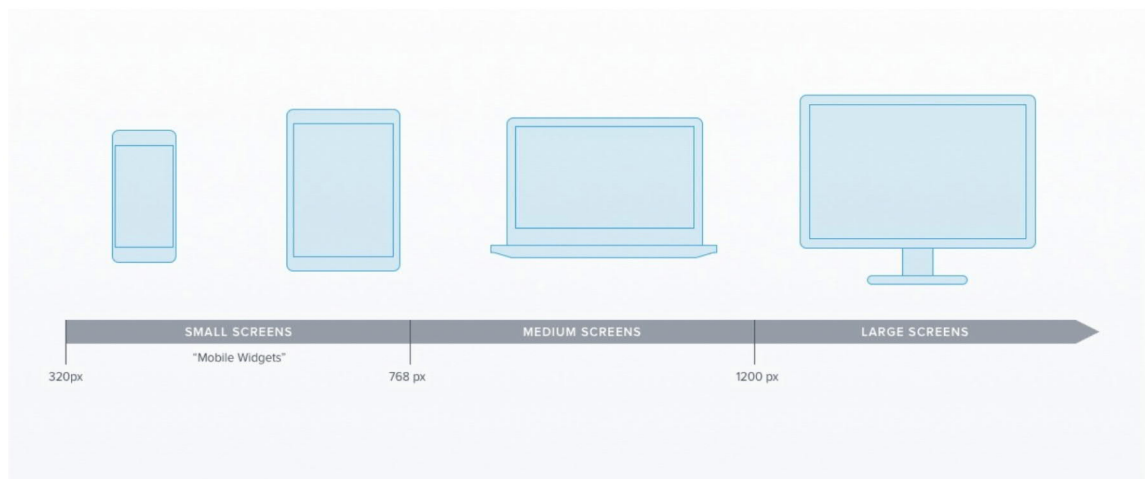
The following are the minimum dimensions supported for mobile/desktop widgets:

- Minimum height: 550px
- Desktop Master Widget minimum height: 600px
- Minimum width: 320px

Responsive Breakpoints

The image below shows the width breakpoints at which the design will change. These are based on typical industry device resolutions.

- Small: 320px to 767px
- Medium: 768px to 1199px
- Large: 1200px or greater



MINI WIDGETS

There are three small, lightweight, *mini widgets* available as well: Budgets, Net Worth, and Spending. These reflect the same design principles as the mobile/desktop widgets, but they do not share the same codebase and are not responsive.

They are suitable for both mobile and desktop applications.

i — The Transactions and Accounts mini widgets have been deprecated and are no longer supported.

Minimum Mini Widget Dimensions

Because mini widgets are designed to be lightweight for specialized situations, they are more restricted in size.

Budgets and Spending

- Minimum height: 300px
- Supported widths: 180px to 750px
- Recommended widths for best results: 300px to 450px

- Recommended widths for best results: 300px to 450px
- Net Worth**
- Minimum height: 400px
 - Supported widths: 300px to 750px
 - Recommended widths for best results: 300px to 450px

IMPLEMENTING MOBILE WIDGETS

Broadly speaking, there are two steps involved in implementing widgets in a mobile context:

1. Get a widget URL through the [SSO API](#) (and remember that there are several mobile-specific widget URLs used for analytics purposes);
2. Load that URL into a `WebView`.

Below, we outline solutions to several common problems that occur during step 2.

Minimum Size Problems

In order to embed our mobile widgets into a `WebView`, one of the requirements is a device width of at least 320 pixels. Depending on the implementation of the `WebView`, smaller devices may not be provided the full width, leading to display issues.

iOS — `WKWebView` vs. `UIWebView`

In apps that run in iOS 8 and later, MX *only* supports **`WKWebView`**.

All partners who have previously implemented **`UIWebView`** should update their implementation to use **`WKWebView`**.

This change in support was made because MX has seen far fewer issues when partners use the `WKWebView` class. Apple also makes this recommendation. For more information see Apple's developer documentation for [UIWebView](#) and [WKWebView](#).

iOS — Default Padding Problems

iOS will add additional padding to its `WebViews` by default; this sometimes causes problems.

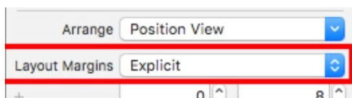
 — *Android does not typically have default padding on `WebViews`. MX recommends partners ensure they are not adding padding explicitly.*

To fix this, follow these simple steps:

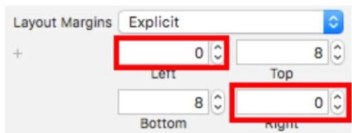
1. Select the `WebView` providing the widgets in your application and navigate to the size inspector;



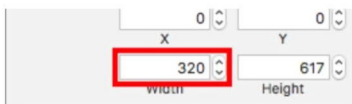
2. Change the layout margins from "Default" to "Explicit";



3. Update the left and right margins to "0";



4. Ensure the Width is *at least* 320 pixels.



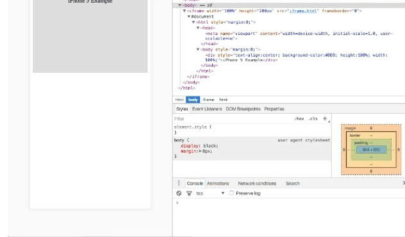
iOS and Android — Default Margin Problems

Whether using `WebViews` on Android or iOS, most browsers will have a default margin (set in the user agent stylesheet) on the body element when rendering the HTML page responsible for loading a widget. This margin is deducted from the total available width of the containing element, which will cause a problem.

To fix this, follow these simple steps:

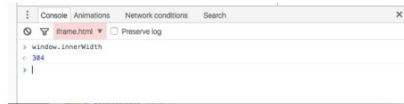
1. Determine the computed width available on the body element;





(Click to enlarge.)

i — The width available to the iframe can be confirmed by inspecting the iframe injected by MX and typing `window.innerWidth` in the javascript console. The width available to the iframe must be at least 320 pixels.



(Click to enlarge.)

2. Confirm the body and HTML elements have their padding and margin set to "0."

iOS and Android — Viewport problems

In order for mobile widgets to render properly, the *viewport* must be set in a meta tag on the HTML page used to load the widget URL.

The viewport is the size of the window through which a page is seen; it can be smaller *or larger* than the actual size of a page or device screen.

On most mobile devices, the virtual viewport is larger than the actual screen size; web pages are rendered according to the *viewport* size, then shrunk down to the actual screen size. This helps when viewing pages that aren't optimized for mobile, but for pages that *are* optimized for mobile (like the mobile widgets), the viewport meta tag is used to guarantee that the page is rendered properly.

Set a meta tag within the `<head>` element as follows:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```