

Building Applications for the Vewd App Store



Please note that parts of this article refers to older Presto-based versions of the Vewd products (Vewd Core 3.x). These versions are still live in a lot of devices that are active in all markets, but most devices shipped since 2015 are based on the newer Blink-based Vewd product.

Update history:

1. 18 April 2012: new section for currently known limitations of the Vewd App Store.
2. 27 August 2012: explicit note about lack of support for third-party multimedia plugins.
3. 5 September 2012: removed the requirement for inclusion of the Vewd App Store API.
4. 20 January 2013: added note about modal dialogs and lack of support for `alert()`, `prompt()` and `confirm()`

- [Introduction](#)
- [Building your application](#)
- [Requirements specific to Vewd App Store applications](#)
 - [Resolution](#)
 - [Navigation and functional keys](#)
 - [Modal dialogs](#)
 - [Multimedia plugins / Flash](#)
 - [Closing applications](#)
 - [Limitations of current implementations](#)
- [Submitting to the Vewd App Store](#)

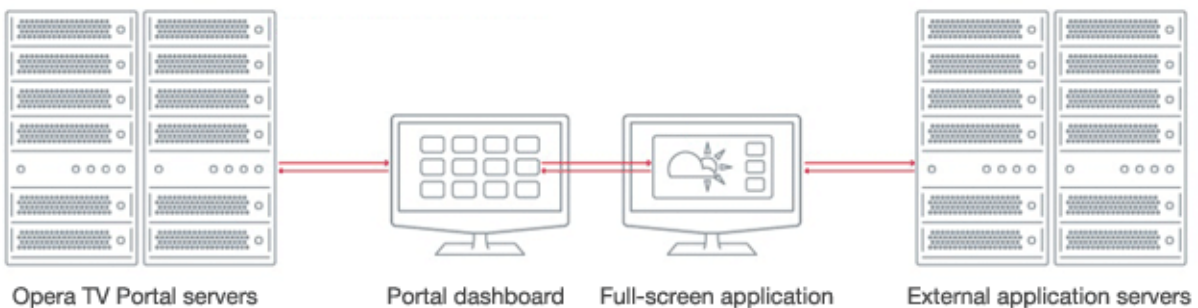
Introduction

The Vewd App Store is a moderated, hosted catalogue of TV-specific web applications. Developers can submit and share their apps through this portal. It presents end users with a storefront (itself a web-based application) which allows them quick and easy access to the applications.



Sample Vewd App Store and a sample weather application

Applications in the Vewd App Store will be presented as static thumbnail images in a dashboard. Users are able to browse the TV Store catalogue and "install" applications, adding them to their dashboard. Selecting an application launches it in full-screen mode.



A simple overview of the Vewd App Store architecture

The full-screen web applications themselves are not hosted directly on Vewd's servers. The Vewd App Store only acts as a directory, with references to the actual URLs of the applications.

Building your application

Applications for the Vewd App Store are, in essence, regular web applications, which are rendered by a customised version of [Vewd Core](#) browsing environment on the user's TV. As such, developers can tap into all the traditional web technologies (HTML5, CSS 3, JavaScript, SVG) supported by the Vewd Core. See the documentation on [web specifications supported by Vewd Core 3.x](#) (and in particular our comparison of [support in different Vewd product lines](#)) for a thorough breakdown.

Although the Vewd Core is built on the same core rendering engine as our desktop browsers, there are still platform-specific APIs and subtle integration differences that developers need to be aware of. For this reason, it is recommended that developers test their applications on an actual TV running the Vewd App Store and/or with the [Vewd TV Emulator](#).

Developing web content for TV brings with it its own challenges – from a difference in user interaction to considerations regarding device capabilities and performance optimisation. For an introduction to some of the necessary adaptations and techniques see our documentation on [creating web content for TV](#) and other articles in our [TV section](#).

Requirements specific to Vewd App Store applications

Due to the way in which applications will be integrated into, and launched from, the Vewd App Store, developers need to be aware of the following additional requirements:

Resolution

All TV Store applications have to support 1280x720 resolution. Other common TV resolutions (1920x1080, 960x540) are currently not supported. Although the TV Store browser itself supports regular scrolling, applications should be designed not to require scrolling, and if needed should provide their own custom mechanisms for displaying long lists of items or content.

Navigation and functional keys

The Vewd App Store is designed to use the standard four-way directional keys on a remote control for spatial navigation. Authors should test that their applications work correctly using the default spatial navigation built into the Vewd App Store browser. Authors may also choose to handle the navigation of their application themselves by intercepting key presses from the remote control. As the exact key codes for remote control functional keys varies between different devices, the Vewd App Store browser provides built-in constants (tailored to the device currently running the TV Store). See the article on [Function al key handling in Vewd App Store applications](#) for further details.

Modal dialogs

As the standard JavaScript methods for `alert()`, `prompt()` and `confirm()` are not supported on all platforms that run the Vewd App Store (since they require platform integration), applications that needs modal dialogs (such as data entry prompts or general alerts/messages) need to provide these using HTML/CSS/JavaScript – which also allows for these modal dialogs to be styled in line with the rest of the application.

Multimedia plugins / Flash

The Vewd App Store does not provide support for any third-party plugins, such as Adobe Flash. Any multimedia features and functionality needed by an application must be implemented using HTML5 technologies such as `<canvas>` or the native support for `<audio>` and `<video>` elements.

Closing applications

Once an application is launched from the dashboard, it is effectively opened full screen in a new window. The Vewd App Store browser (based on the Vewd Core, responsible for rendering the applications and the TV Store itself) does not present any standard interface elements to the user – it runs as a completely “chromeless” browser, with no address bar or back button. Users will be able to close the application and return to their dashboard via the remote control's *EXIT* and/or *RETURN* key. Developers should nonetheless provide an explicit option or button in their UI to close the application, with a simple call to the `window.close()` method.

Limitations of current implementations

There are known limitations in some of the current devices that are shipping with the Vewd App Store, generally due to the way certain features have been integrated on the devices by the manufacturers and which version of the TV Store software has been used.

- Missing [CORS \(Cross-Origin Resource Sharing\)](#) support: if your application relies on cross-origin resources (particularly via `XMLHttpRequest`), you will need to implement some form of proxy on your application's origin domain to relay the resources from other domains.
- Due to the integration with external media playback frameworks on certain devices, it may not be possible to guarantee simultaneous playback of more than a single `audio` or `video` element.

Submitting to the Vewd App Store

Once your application is ready, it can be submitted to the Vewd App Store. Log in to the [Vewd App Store – Submission portal](#) and follow the steps required in the submission dialog.

This article summarises the most important guidelines that applications need to follow for inclusion in the Vewd App Store. Further restrictions and requirements are listed in the **acceptance criteria** of the [Vewd App Store application publishing guidelines](#). Please ensure that your application fulfills those criteria – as well as the best practices and suggestions laid out in our developer articles – before submitting your application.