

ENABLING DISCONNECTED SCENARIOS WITH HTML5

W3C WORKSHOP ON THE FUTURE OF OFF-LINE WEB APPLICATIONS
5 NOVEMBER 2011
REDWOOD CITY, CA, USA

Israel Hilerio, Ph.D. (israelh@microsoft.com), Internet Explorer Program Manager, Microsoft Corporation
Adrian Bateman (adrianba@microsoft.com), Internet Explorer Program Manager, Microsoft Corporation

INTRODUCTION

Building web applications that can deal with disconnected scenarios typically involves managing the content that comprises the applications “code” and the data that personalizes the application for a specific user. The best written web applications provide use this separation of content and data to provide a superior end-user experience. For example, web games allow you to play games by providing rich content (e.g. HTML, JS, CSS, Img, Audio, etc.) while at the same time locally keeping track of your scores and settings separately (e.g. JS Objects, cookies, or web storage data).

With the advent of social networks and information sharing there are many web applications that deliver tailored experiences by storing data about you. Business web applications keep track of your settings and preferences. There are few one dimensional web applications that only deliver content without a need for data storage (documentation sites).

In this new world, both content and data play an integral part in delivering a complete solution. As web applications become more pervasive and more people store their information in the cloud there is less tolerance for not having access to important data while disconnected.

The situation is more visible when we access the cloud from portable devices (laptops, tablets, phones, etc.) where connectivity varies. This is the case when using metered networks, networks with data roaming charges, or spotty network coverage. People want access to their data without having to be connected and paying for connection charges.

We believe that in order for web applications to provide robust solutions they will need to deal with these types of disconnected scenarios and design solutions that handle both offline content and data. This will enable end-users to complete their operations uninterrupted and for content updates to happen behind the scenes when there is network connectivity.

The goal of this paper is to demonstrate the need to simultaneously deal with both content and data when architecting an offline solution. Dealing only with either content or data will create an incomplete solution that can't be successfully deployed.

SCENARIOS

TEMPORARY LACK OF NETWORK CONNECTIVITY

There are situations when you are using a portable device and your network connectivity fluctuates. When this happens today and network requests are made the flow of your application is disrupted and users may see a page not found message or similar error. This is an unexpected situation for the end-user.

This error can happen at very inopportune times when you are:

- Submitting your credit card to purchase a store item
- Looking for a friend's contact information
- Sending an email to a friend
- Etc.

ACCESS INFORMATION WHILE DISCONNECTED

There are situations when you want to use your portable device to access cloud data without having access to a network. One reason can be that you don't have access to a network or the data roaming charges are prohibitive. While many users could understand why this doesn't work they are still disappointed by the limitations of the technology. They just want to access their data.

Both of these scenarios require the web apps to pre-fetch content to display offline and facilitate information access via the querying of data with JavaScript. In some cases, more sophisticated applications will have to queue up requests and version resources to later upload to the cloud when connectivity is restored.

What is important to understand is that caching content without addressing how to store/access local data will provide an incomplete solution.

TECHNOLOGIES

The open web platform provides us with many new technologies that when used together provide a complete solution for managing both content and data while disconnected. These technologies are:

CONTENT

- Application Caches – cache web resources locally and map cached resources to URL requests
- Widget Package – enables the deployment of web applications independent of the browser

DATA

- Indexed Databases – store/cache and index large number of objects locally to enable efficiently searches
- Web Storage – store/cache small amounts of textual information for later use
- File and FileReader APIs – provides local access to the files and blobs

DATA SYNCHRONIZATION

- WebSockets and XHR – provides the network connection to synchronize information between the client and the server

MISCELLANEOUS SUPPORT

- Online/offline events/status - provides a mechanism to detect network connections

CONCLUSION

Technologies like Application Cache and Widget Packaging alone are insufficient for creating sophisticated disconnected applications. Without a comprehensive approach for dealing with offline content as well as offline data, developers will not invest the time and effort it takes to support these disconnected scenarios. Areas like data synchronization and more reliable offline/online events can be challenging but we believe best practice and libraries will help address these.