

Technical Disclosure Commons

Defensive Publications Series

March 29, 2019

Transferable Deep Links

Victor Cărbune

Thomas Deselaers

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Cărbune, Victor and Deselaers, Thomas, "Transferable Deep Links", Technical Disclosure Commons, (March 29, 2019)
https://www.tdcommons.org/dpubs_series/2094



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Transferable Deep Links

ABSTRACT

The disclosure describes techniques to automatically generate shareable deep links from applications and/or web pages. A system level application programming interface (API) provides automatically transferable deep links from application or web pages. Per the techniques, applications or web pages store a given state along with an associated hash. When a link to a shopping cart is shared by a user, the recipient sees the same content as the sender which is enabled by the use of the associated hash. While the applications or web pages control shareable status, the sharable deep link techniques are provided via a browser or operating system and are made available system-wide.

KEYWORDS

- deep link
- application state
- app state
- browser state
- state transfer
- state object
- serialization

BACKGROUND

In-app links can be generated from a software application, e.g., a mobile app, or a browser, and sent via a communication application e.g., email, chat, etc. to share links to online content. Current techniques for link sharing allow such links to be accessible to other users only when they are public; if the link is shared with the same user, the link can be accessible, but

associated state of the application or the browser is not included. Simply clicking the hyperlink does not take the recipient to the same content as what the sender is viewing. For example, a hyperlink that links a specific message, generated from within an email application, does not enable another user to view the specific message by selecting the hyperlink.

DESCRIPTION

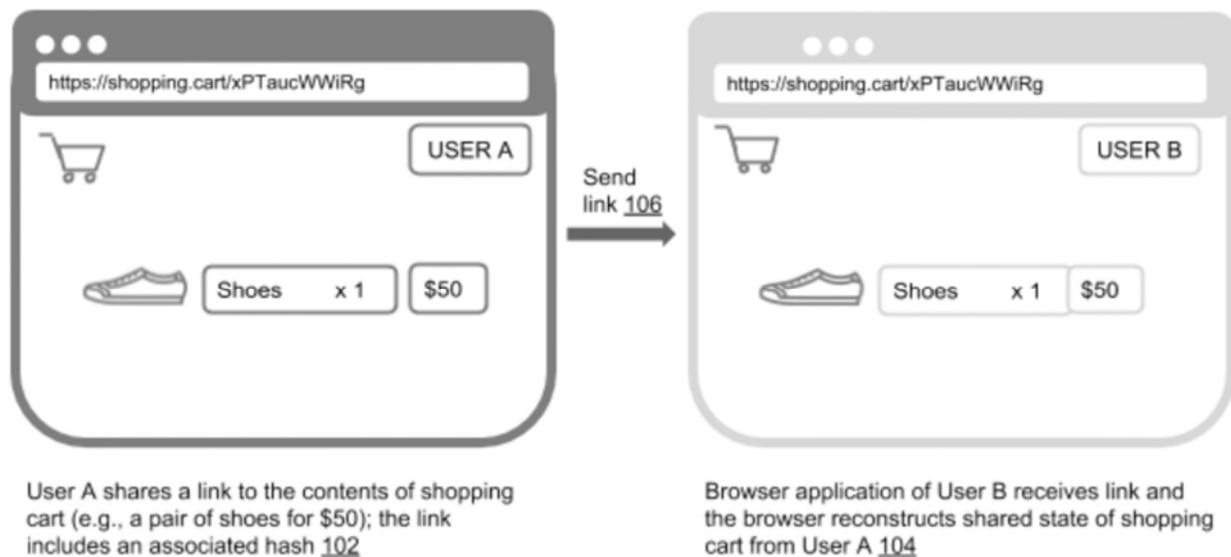


Fig. 1: Sharing deep link with another user to reconstruct application state

Fig. 1 illustrates an example of use of the techniques of this disclosure that enable sharing of a deep link to reconstruct application state. In the example, user A (sender) is viewing a webpage (102) from a shopping website. Specifically, user A has added a pair of shoes to their shopping cart. When the user shares a link (<https://shopping.cart/xPTaucWWiRg>) to the shopping cart with another user B (recipient), the link includes a reference to a one-way hash (“xPTaucWWiRg”). When user B opens the link via a web browser, the browser reconstructs the state of the shopping cart (104) based on the one-way hash. User B is thus able to view the shopping cart exactly in the same state as when shared by user A.

The techniques automatically generate shareable deep links from user-specific applications or webpages. A system level API is provided to automatically generate transferable deep links. The API includes the following features:

1. **Serialization and deserialization of state objects:** A suitable technique to hash objects is utilized for encapsulating recursively the state of the different parts of a web page (or application) at the sender's end. With user permission, the hash is stored on a server, e.g., by an application provider that provides the deep links.
2. **Automatic generation of shared link:** Each collection of state objects is hashed and a single unique hash is obtained that corresponds to the deep link. A cryptographically strong hash is selected to prevent attackers from reconstructing the contents of the state.
3. **Creation of ad-hoc user shared context:** A mechanism to specify sharing is provided with application specific privacy settings. For example, a deep link generated from an email application is valid only if the recipient of the deep link has also received the same email. When the deep link is for a shopping cart, the recipient effectively shares the same basket view as the sender, but does not receive sender information, e.g., sender's payment mechanisms set up at the shopping website.

The system level API enables deep links to be shared without individual app developers having to build the functionality. The application can simply store a given state and provide an associated hash via the API. The functionality of serialization/deserialization is provided at the rendered level (e.g., by a browser). The application retains control over the views that are serializable (and shareable).

To make links transferable, techniques are provided that enable the browser (or application) to reconstruct the state. For example, for an email application or website, the

browser, with user permission, can obtain and transfer a code that is included in all emails, e.g. a hashed representation of the transmission codes of the sending email service. The transmission codes are cryptographic random numbers, which are searchable and likely unique. The recipient's email client can search the recipient's inbox for this content and render it if found. In other scenarios, e.g. online shopping, when the sender provides a command to generate a deep link and provides permission, the browser encodes the state of the website in a compact representation that can be transferred.

The techniques permit a browser provider (or other centralized provider) to provide a mechanism for transfer for deep links and obviate the need for app or website developers to implement their own mechanisms for transferable deep links. The implemented mechanism can be limited to certain categories of web pages, e.g., email, shopping websites, etc. The described features can be implemented incrementally, e.g., initially, the deep link mechanism can be implemented to support a limited number of apps or websites, e.g., based on popularity. Users can be provided with the ability to request addition of deep link features for other websites. Such users can provide additional information, e.g. a visual action history of the page and a selection of the relevant actions that allow making the link transferable. The techniques can be implemented as part of a browser application or an operating system.

Further to the descriptions above, a user is provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data is treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a

user's identity is treated so that no personally identifiable information can be determined for the user, or a user's geographic location is generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user has control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

The disclosure describes techniques to automatically generate shareable deep links from applications and/or web pages. A system level application programming interface (API) provides automatically transferable deep links from application or web pages. Per the techniques, applications or web pages store a given state along with an associated hash. When a link to a shopping cart is shared by a user, the recipient sees the same content as the sender which is enabled by the use of the associated hash. While the applications or web pages control shareable status, the sharable deep link techniques are provided via a browser or operating system and are made available system-wide.

REFERENCES

1. Ma, Yun, Xuanzhe Liu, Ruogu Du, Ziniu Hu, Yi Liu, Meihua Yu, and Gang Huang. "DroidLink: Automated generation of deep links for Android apps." arXiv preprint arXiv:1605.06928 (2016).
2. Venkat Raman, Tiruvilwamalai, Chengling Chan, Mukarram Tariq, and Stefan Lafon. "Mobile application state identifier framework." U.S. Patent 9,621,650, issued April 11, 2017.