

User Locale Preferences

Ben Allen
January 2024



Current status

- Proposal allows users to transmit certain commonly used subsets of user preferences related to:
 - Preferred numbering system
 - Preferred calendar
 - First day of week
 - Temperature measurement unit
 - Hour cycle



Why is this hard?

- Our goal is to ensure that users from a wide range of languages/regions have as seamless an experience of the web as possible.
- However, the web is a hostile environment. If web clients provide servers with detailed/fine-grained information about user localization preferences, this can result in users becoming immediately individually identifiable.
- Revealing preference information can be dangerous even if relatively few bits of entropy are thereby lost
 - This information can reveal peoples' identity categories
 - This information is about the user rather than their device (can be used for cross-device tracking)



Goals

Allow users to request the following types of content tailorings:

- Any combination of values for the five locale extension tags listed above that matches the default settings in the locale (or locales) sent in the client's Accept-Language header.
- Any combination of values for the five locale extension tags that is otherwise commonly used in the locale (or locales) sent in the client's Accept-Language header.
- Any combination of values which bears little surprisal when considered in combination with other locale-related information about the user that the server already knows.
- Values for numbering system that, if not honored, would result in users receiving unintelligible content.



Current status

- Support from Apple and Mozilla
 - But this support is contingent on effective fingerprinting mitigation strategies



Commonly used preferences

How do we determine what counts as “commonly used preferences”? Let’s perform a little sleight of hand (we’ll un-perform it shortly)

- Assume that all commonly used preference combinations are the default in *some* locale
- Make those common defaults available



commonly used preferences (fw, hc, mu)

extension string	population	# locales using
-u-fw-mon-hc-h23-mu-celsius	2,714,937,996	674
-u-fw-sun-hc-h12-mu-celsius	1,665,105,458	277
-u-fw-sun-hc-h23-mu-celsius	917,309,644	199
-u-fw-sun-hc-h12-mu-fahrenhe	332,515,201	26
-u-fw-mon-hc-h12-mu-celsius	315,642,460	173
-u-fw-sat-hc-h12-mu-celsius	224,538,941	53
-u-fw-sat-hc-h23-mu-celsius	82,481,712	30
-u-fw-fri-hc-h23-mu-celsius	385,633	2
-u-fw-sun-hc-h23-mu-fahrenhe	307,290	2
-u-fw-mon-hc-h12-mu-fahrenhe	81,212	3

Considering locale

The last slide gave us a potential set of strings which, taken in isolation, would reveal relatively little about the user.

HOWEVER:

- we got those numbers by considering the defaults for each locale.
- Servers necessarily know the locale the user prefers (in Accept-Languages, etc)
- Because we're talking about the defaults for each locale, we know that the actual preferences will correlate strongly with user locale



User research required

- Consider this case: a student visiting the U.S. from the Netherlands has OS settings reflecting their preference for temperatures in Celsius
 - 1.If the student's browser locale is 'nl', the server can already determine that they most likely prefer -u-fw-mon-hc-h23-mu-celsius by inspecting the Accept-Language header. Because of the strong correlation between a user's settings and the locale the browser sends, the surprisal of this set of preferences is low.

Considering locale

Consider this case: a student visiting the U.S. from the Netherlands has OS settings reflecting their preference for temperatures in Celsius

1. If the student's browser locale is 'nl', the server can already determine that they most likely prefer -u-fw-mon-hc-h23-mu-celsius by inspecting the Accept-Language header. Because of the strong correlation between a user's settings and the locale the browser sends, the surprisal of this set of preferences is low.



Considering locale

Consider this case: a student visiting the U.S. from the Netherlands has OS settings reflecting their preference for temperatures in Celsius

- If the student's browser locale is 'nl', the server can already determine that they most likely prefer -u-fw-mon-hc-h23-mu-celsius by inspecting the Accept-Language header. Because of the strong correlation between a user's settings and the locale the browser sends, the surprisal of this set of preferences is low.



Considering locale

Consider this case: a student visiting the U.S. from the Netherlands has OS settings reflecting their preference for temperatures in Celsius

- If the student's browser locale is 'en-US', revealing this preference string will likewise leave them with a relatively large anonymity set — most people from the regions that use those preferences by default will have left their defaults unchanged



Considering locale

Determining the number of bits of information revealed in the second of the above scenarios will require user research. If about 5% of users of browsers localized to 'en-US' prefer -u-fw-mon-hc-h23-mu-celsius, either because they are, like our student, visiting the United States, or because for whatever reason they simply decided to download a browser version localized to 'en-US', our student has lost around 4.32 bits of entropy by revealing that preference. This is a significant amount of lost entropy.



Conditioning on information other than user locale

The server has additional information about the user which may reduce the surprisal of the revealed information

- We can condition, for example, on the *content* locale
- Figuring out what can/should be used to condition the surprisal of revealed information will require user research



