Pages

Assessing the use of Google web fonts

Created by Sue Anna Joe, last modified on Jun 25, 2018

Intro

UX would like to use two Google web fonts on our sites:

- Rubik for body text
- Montserrat for headers

Cliff Notes summary

- Rubik and Montserrat do not support Greek or our Asian languages, but these characters will be rendered using the device's system font.
- Since Rubik does not support the Cyrillic Extended subset, I would recommend we use only the Latin and Latin Extended subsets for both fonts. This would ensure consistent character rendering.
- Desktop testing showed a 1% increase in page load. The experience did not seem to be impacted much at all.
- The fonts did not appear to break the layout in Web. I would not expect any issues in Touch, but of course it would need testing.
- Text flashing was observed with cached disabled. This might not be a big concern. It's a bit edge case-y but can definitely happen for first-time users with an even worse experience for first-time users on a slow connection. After the flash the fonts will be cached.
- We should use the font API to serve the font files.

Character set support for non-English languages

Zoosk is available in 26 languages, some of which require character subsets so that all characters are rendered with visual consistency.

Zoosk-supported languages requiring the Latin Extended subset: (see source)

- Serbian and Croatian ?, ?, ?, Š, Ž
- Catalan À, É, È, Í, Ï, Ó, Ò, Ú, Ü, Ç
- Danish, Norwegian Æ, Å, Ø
- English £, ¢
- Finnish À, Ä, Ö, Š, Ž
- French Æ, Œ, Â, À, É, È, Ê, Ë, Î, Ï, Ô, Ù, Û, Ü, Ÿ, Ç, », «
- German Ä, Ö, Ü, ß
- Hungarian Á, É, Í, Ó, Ö, ?, Ú, Ü, ?
- Polish ?, ?, ?, ?, ?, Ó, ?, ?, ?
- Portuguese Á, Â, Ã, À, É, Ê, Ó, Ô, Õ, Ú, Ç
- Romanian ?, Â, Î, ?, ?
- Turkish ?, I (dotless lowercase), ?, Ö, ?, Ü
- Spanish Ñ, ¿, ¡
- Swedish Å, Ä, Ö

Zoosk-supported languages requiring the Cyrillic subset:

- Russian
- Serbian

Zoosk-supported languages requiring the Cyrillic Extended subset:

- Russian
- Serbian

Our supported Asian languages and Greek require their own character sets.

The two fonts support some but not all subsets:

	Latin	Latin Extended	Cyrillic	Cyrillic Extended	Vietnamese*	Hebrew*
Montserrat	✓	✓	✓	✓	✓	✗
Rubik	✓	✓	✓	✗	✗	✓

*Currently not supported on Zoosk products

Testing

When I incorporated the fonts on my VM, I compared Greek and traditional Chinese on my dev environment with production and found that:

- Since Rubik and Monserrat don't support Greek and Asian languages, the characters are rendered properly using system fonts.
- The characters look the same except for a slight increase in line height in development mode.

Recommendations

Since Rubik does not support Cyrillic Extended, I advise we use only the Latin and Latin Extended subsets. This means text in Russian and Serbian, like Asian languages and Greek, would be rendered by system fonts and look as they are in production. This will ensure the consistency in character rendering, and it will reduce the amount of bytes we're loading.

Using the Google Font API vs. hosting the font files on Akamai

Google Font API

Using Google's Font API means adding a link tag in our sites' head tags:

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat:500,800|Rubik:400,400i,700&amp;subset=latin-ext">
```

Akamai hosting

Hosting the font files ourselves requires:

- Using @font-face in our CSS
- Adding the font files to Zephyr
- Hosting them on Akamai
- Checking and installing future font updates
- Downloading of extra font files initially and when cache is cleared

This method requires the following CSS:

```css
/* montserrat-500 - latin_latin-ext */
@font-face {
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 500;
    src: local('Montserrat Medium'), local('Montserrat-Medium'),
    font-url('montserrat-v12-latin_latin-ext-500.woff2') format('woff2'), /* Super Modern Browsers */
    font-url('montserrat-v12-latin_latin-ext-500.woff') format('woff'), /* Modern Browsers */
    font-url('montserrat-v12-latin_latin-ext-500.ttf') format('truetype') /* Safari, Android, iOS */
}

/* montserrat-800 - latin_latin-ext */
@font-face {
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 800;
    src: local('Montserrat ExtraBold'), local('Montserrat-ExtraBold'),
    font-url('montserrat-v12-latin_latin-ext-800.woff2') format('woff2'), /* Super Modern Browsers */
    font-url('montserrat-v12-latin_latin-ext-800.woff') format('woff'), /* Modern Browsers */
    font-url('montserrat-v12-latin_latin-ext-800.ttf') format('truetype') /* Safari, Android, iOS */
}

/* rubik-regular - latin_latin-ext */
@font-face {
    font-family: 'Rubik';
    font-style: normal;
    font-weight: 400;
    src: local('Rubik'), local('Rubik-Regular'),
    font-url('rubik-v7-latin_latin-ext-regular.woff2') format('woff2'), /* Super Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-regular.woff') format('woff'), /* Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-regular.ttf') format('truetype') /* Safari, Android, iOS */
}

/* rubik-italic - latin_latin-ext */
@font-face {
    font-family: 'Rubik';
    font-style: italic;
    font-weight: 400;
    src: local('Rubik Italic'), local('Rubik-Italic'),
    font-url('rubik-v7-latin_latin-ext-italic.woff2') format('woff2'), /* Super Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-italic.woff') format('woff'), /* Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-italic.ttf') format('truetype') /* Safari, Android, iOS */
}

/* rubik-700 - latin_latin-ext */
@font-face {
    font-family: 'Rubik';
    font-style: normal;
    font-weight: 700;
    src: local('Rubik Bold'), local('Rubik-Bold'),
    font-url('rubik-v7-latin_latin-ext-700.woff2') format('woff2'), /* Super Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-700.woff') format('woff'), /* Modern Browsers */
    font-url('rubik-v7-latin_latin-ext-700.ttf') format('truetype') /* Safari, Android, iOS */
}
```

Performance

I ran several tests in my dev VM to gather performance data:

- No Google web fonts with and without caching
- Google Font API with and without caching
- Self-served font files with @font-face and caching

Since I can't embed my private Google spreadsheet, here is a screenshot of the average data in milliseconds:

| | with cache | | | | | with cache | | | | | no cache | | | |
	no fonts	font api, requested fonts, latin and latin ext only				no fonts	@font-face, requested fonts, latin and lating ext only				no fonts	font api, requested fonts, latin and latin ext only		
ave finish	2138	2169	31	1.45%		2138	2445	307	14.36%		2287	2626	339	14.82%
ave dom content loaded	1180	1193	13	1.10%		1180	1268	88	7.46%		1262	1399	137	10.86%
ave load	2148	2173	25	1.16%		2148	2464	316	14.71%		2301	2643	342	14.86%

I also ran tests on a slow 3G connection via Chrome dev tools:

| | slow 3g with cache | | | |
	no fonts	font api, requested fonts, latin and latin ext only		
ave finish	16084	17472	1388	8.63%
ave dom content loaded	6840	6902	62	0.91%
ave load	16095	17486	1391	8.64%

Rendering

Live rendering

I took a few videos to confirm text rendering in different scenarios

I took a few videos to confirm text rendering in different scenarios.

#1 - With cache enabled

- Google Font API
- Includes both fonts and desired weights
- Latin and Latin Extended subsets

✅ The Rubik body text and Montserrat headers rendered without flashing of any kind.
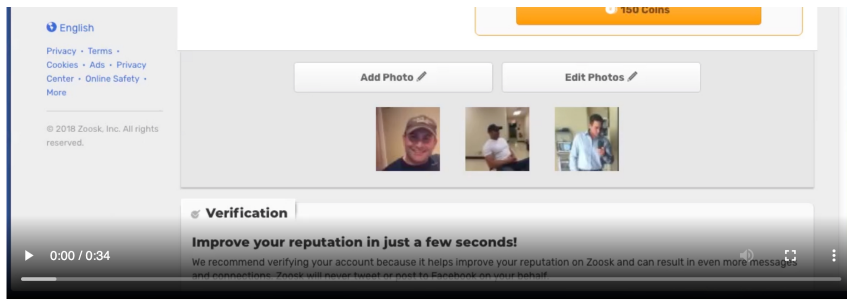

#2 - With cache disabled

- Google Font API
- Includes both fonts and desired weights
- Latin and Latin Extended subsets

⚠️ Most Rubik body text rendered without flashing of any kind, but small bold body text was initially invisible then popped in, rendered in Rubik. However Montserrat headers did not initially appear and then later appeared in the expected font. This might not be a big concern since this would happen to users who cleared or disabled cache (edge case-y) or landed on Zoosk for the first time.


#3 - With cache enabled on a slow 3G connection

- Google Font API
- Includes both fonts and desired weights
- Latin and Latin Extended subsets

✅ The Rubik body text and Montserrat headers rendered without flashing of any kind.

#4 - With cache disabled on a slow 3G connection

- Google Font API
- Includes both fonts and desired weights
- Latin and Latin Extended subsets

⚠️ Same results as #2, but when the headers did appear they were rendered in Helvetica then changed to Montserrat. Again, this might be an edge-case concern.
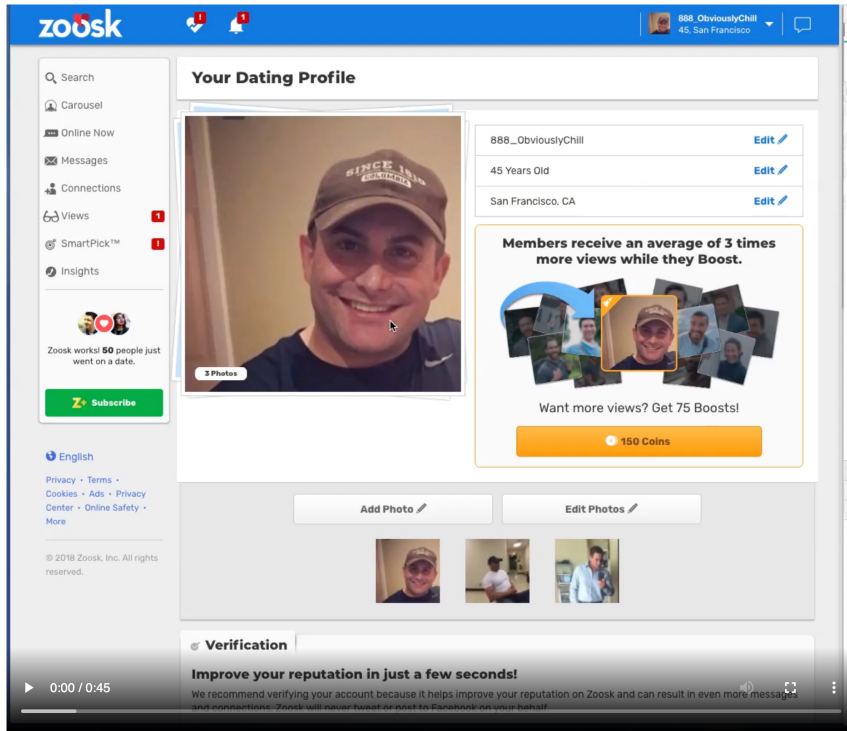
#5 - With cache enabled and @font-face/self-served font files

- Includes both fonts and desired weights
- Latin and Latin Extended subsets

❌ The body text (including small bold text) in the left nav as well as Montserrat headers were invisible then popped in as the expected fonts. This is a problem because cache was enabled. Because of this I would recommend not serving the font files ourselves.
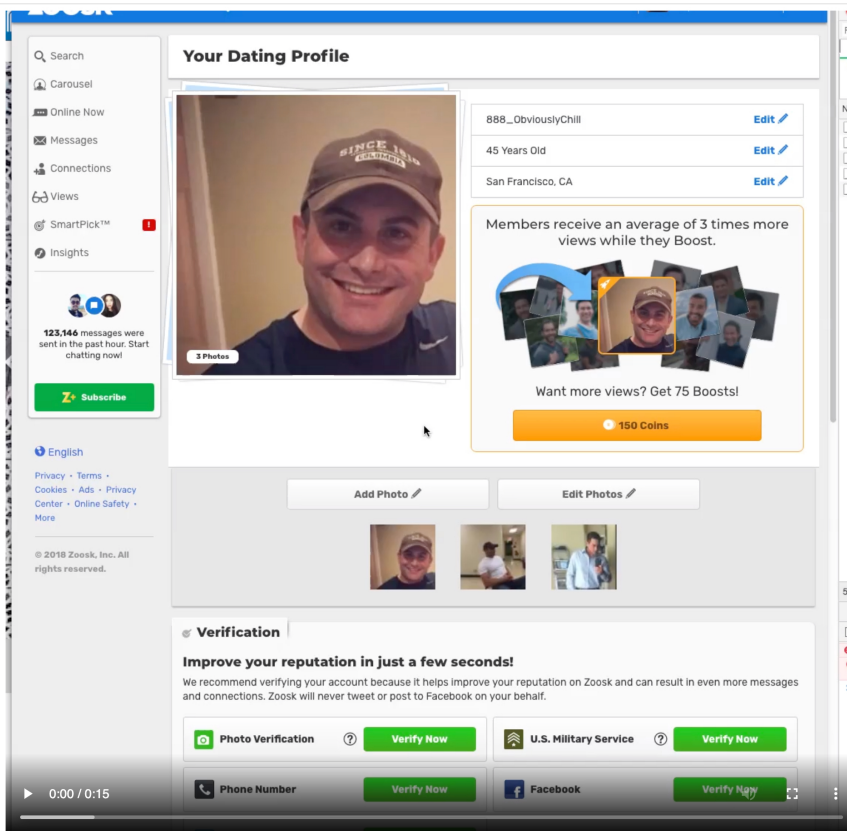
Rendering on different computers

To manage expectations, I included screenshots from a few different browsers and OSs to show differences in text rendering. I changed the body text to Rubik and chose a few specific elements to use Montserrat Extra Bold (700). It's best to zoom into the images for greater detail.
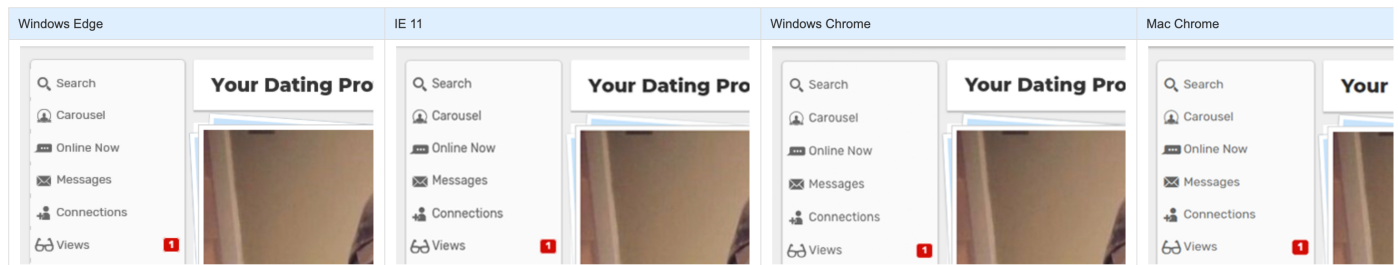
Windows

Unlike Mac browsers, Edge and IE 11 do not support anti-aliasing, which means text edges are not blurred to give a smoother appearance. Windows Chrome, however, does.

Windows Edge	IE 11

Windows Chrome	

Macintosh

Mac Chrome non-retina screen	Mac Chrome retina screen
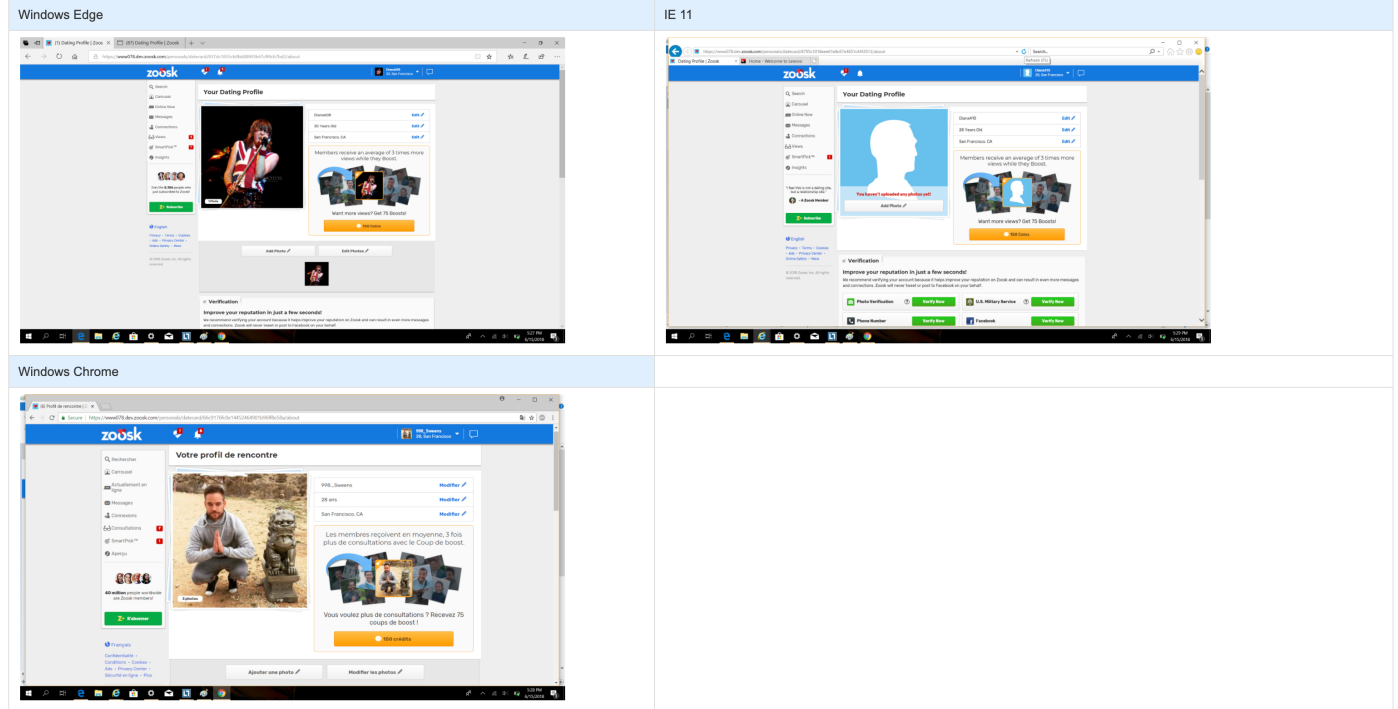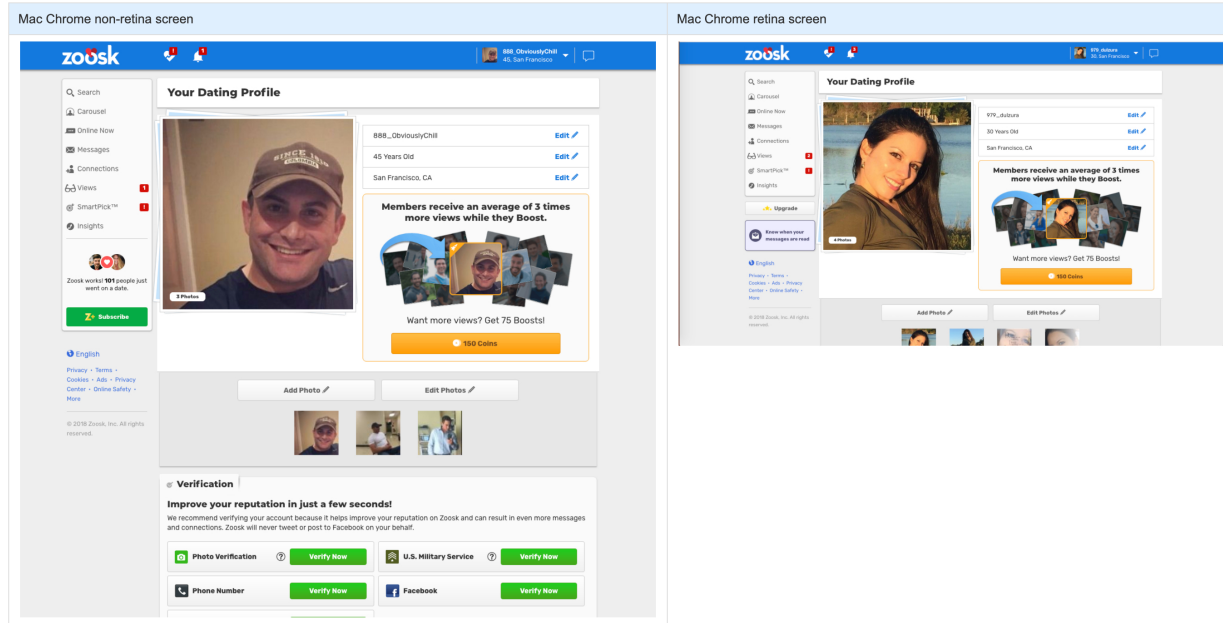
Zoomed in screenshots of browsers/OS

Windows Edge	IE 11	Windows Chrome	Mac Chrome

Do the new fonts break any layouts due to character size and spacing?

Not that I saw, but I didn't do an exhaustive test. I hit the major areas of the desktop site. Touch should get the same treatment.

Options for optimizing performance and visual rendering

- Move the link tag as close to the head tag as possible

👍 Like   Be the first to like this

No labels ✏️

Write a comment…