Work Item: Universal Acceptance support using Javascript libraries React, Angular and Node including Email processing and HTML generation

Ver.: 2024-09-24

Purpose

To identify commonly-used Javascript packages available in NPM (a package manager for the JavaScript programming language) for implementing these Universal Acceptance (UA)-related features in JavaScript-based applications: linkification, accessing inbound email to Email Address Internationalization (EAI) addresses, composing outgoing email messages to EAI addresses, and sending such email. To provide minimum viable code samples demonstrating how to implement these UA-related features. To describe deficiencies in these packages which interfere with UA, to help Universal Acceptance Steering Group (UASG) report and track them with package developers.

- UA Working Group proposing the work item: Technology WG
- Reference to the Action plan: FY24
- Reference to work item(s): T1

Description of Work

Software that uses React, Angular and Node.js often performs functions such as receiving email from a human (perhaps a customer), storing information in a ticket (e.g., “customer x sent message y to customer service at time z”), composing replies (e.g., “We are sorry that we failed to reply within four hours.”), generating web pages (e.g., showing customer service employees today’s incoming mail) and similar tasks.

The NPM package library contains packages to help writing almost anything under the sun, including all of the example tasks mentioned in the previous paragraph.

This work item tests whether it’s possible and simple to write such services that handle Internationalized Domain Names (IDNs) (e.g., تجربة-القبول-الشامل.موريتانيا/تجربة) and EAI addresses (e.g.,ईमेल-परीक्षण@सार्वभौमिक-स्वीकृति-परीक्षण.संगठन) in the same way as

legacy addresses (https://icann.org and uaprogram@icann.org). The code may run in the frontend (using React or Angular), in the backend (node.js) or a mixture of both. In addition to testing, this work item includes, to the extent possible, developing code on how to implement UA-related functionalities. Only packages from NPM, and only code in Javascript are in scope.

The relevant classes of work are:

a. Linkification, i.e., adding links to running text where users think a link is intended.
b. Generating HTML from the templating language in use (often Markdown, as detailed below).
c. Accessing inbound mail on a mail server (IMAP, EWS, Gmail).
d. Processing email, as used when a site/service needs to send an auto response, create tickets or similar.
e. Composing a message for sending, as used when a site/service needs to send a message such as "Your parcel has shipped." or "The plumber will arrive at 12:00 tomorrow."
f. Sending email, either via SMTP or via ESPs such as Sendgrid.

The project will consist of the following steps.

1. For each of these, there are likely to be several NPM packages to perform this kind of task. The next step is therefore to identify 1-5 frequently used packages for each purpose. It is likely that some NPM packages support more than one task, e.g., both composing and sending email. (It is unlikely but possible that React, Angular and/or Node.js contains builtin functionality for any of these tasks. If that is the case, the React/Angular/Node.js is to be tested as if it were a popular NPM package.)

   The working group reviews the list of packages and the vendor's list of what is to be tested for each package. The working group will use no more than two weeks for their review.

2. Test each package's relevant functionality and identify cases where it functions correctly for legacy email addresses/IDNs and fails for Unicode ones.

   a. Any relevant test cases developed in UASG 018 or UASG 037 should be reviewed and updated. The community will review the updated test cases and testing plan before testing.
   b. Showcase UA Readiness level of each package in red/yellow/green color codes as used in UASG 037.

3. Produce a clear and concise description of each deficiency. The preferred format is a simple reproducer, suitable for use in a unit test or bug report.
4. Develop minimal viable products (MVPs) for each function that is UA-ready, as example code and best practices for developers to follow in their own work. The sample code should demonstrate how to implement complete UA readiness, covering IDNs and EAI addresses.

The work builds on testing of programming languages and frameworks (UASG 018, UASG 018A, UASG 037) to check how effectively they support UA of domain names and email addresses, and UASG 043 to provide UA-ready code samples. This includes checking if they appropriately support input, validation, storage, processing and display of all domain names and email addresses.

From the UASG 004, ten email addresses that include test numbers from #32 through #40 and #56, should be included in the proposal for testing, based on covering the variety of issues to be covered and a sample of different scripts. When domain names are needed for testing, the domain names in these ten email addresses should be used.

This includes the functionality for handling all aspects of UA as discussed in UASG 026, covering the variety of test domain names, IDNs and internationalized email addresses in UASG 004. The testing should also cover basic Unicode processing, including normalization to NFC form for IDNs, compatibility to the latest IDNA2008 standard, and support of both A-labels in ASCII and U-labels in UTF-8 format. The testing should also cover URL resolution and email addresses, including EAI support with mailbox in UTF-8 format.

Deliverables

The work will have the following deliverables:

1. The code written/updated by the vendor in the course of testing NPM packages, React, Angular and Node.js to show how to implement UA-related functions.
2. The reproduction materials for each bug/deficiency found. The vendor is not expected to report the bugs; ICANN may or will do that.
3. Draft report on recommended best paths to accomplish the classes of work from the description using the NPM ecosystem
    a. Description of why these packages were selected, an overview of their overall level of UA readiness, and a description of the deficiencies found.
    b. In cases where more than one package provides a given

functionality and only some of the packages work correctly, the report should advise which package works best.

c. Summary information.

4. Presentation (using UASG PowerPoint template provided) covering the contents of the report.

Timeline

- Tentative start date: Date of signing of the contract.
- Tentative end date: 4 months from start date of the contract.

References

NPM packages are frequently used in/together with both React, Angular and Node.js package manager. https://en.wikipedia.org/wiki/Npm

About React: https://en.wikipedia.org/wiki/React_(software)

About Angular: https://en.wikipedia.org/wiki/Angular_(web_framework)

About Node.js: https://en.wikipedia.org/wiki/Node.js

UASG 033 lists top projects with libraries.

To refer to previous documents, see document inventory.

Proposal Submission

The proposal should include the expertise of the organization which demonstrates the expertise available and understanding of the UA related issues.

The proposal should be submitted to: UAProgram@icann.org before the submission due date.

Conflict of Interest

To help avoid any perceived or actual conflict of interest (COI), UASG leaders, UASG Ambassadors, members holding working group's leadership positions in the UASG, and any organization(s) affiliated with individuals in these UASG roles, are prohibited from participating in this SOW. In addition, ICANN org COI applies.

Suggested Markdown and Linkification Tests

The following two paragraphs is a suggested text to test HTML generation for Markdown or Markdown-like software (Kramdown, Commonmark, RST, etc). It may be necessary to modify this for the software's input format – this is the suggested text.

The first paragraph includes links to [ICANN](https://icann.org), [a site with a long TLD](https://universal-acceptance-test.international), [a link containing

ï](https://universal-acceptance-test.international/naïve) and [an IDN.](https://普遍接受-测试.世界)

The second paragraph includes two email addresses – [an indic address](mailto:युएअसजी@डाटामेल.भारत) and [an arabic address](mailto:دون@رسيل.السعودية) one.

The following two paragraphs is suggested text to test whether domains, pseudo-URLs and email addresses are recognized in plain text, for software that auto detects links:

The first paragraph includes links to icann.org, universal-acceptance-test.international, universal-acceptance-test.international/naïve and 普遍接受-测试.世界.

The second paragraph includes two email addresses, namely युएअसजी@डाटामेल.भारत and دون@رسيل.السعودية. Note that the trailing full stop is not part of the email address.