

urltestdata.json

These tests are for browsers, but the data for `a-element.html`, `url-constructor.html`, `a-element-xhtml.xhtml`, and `failure.html` is in `resources/urltestdata.json` and can be re-used by non-browser implementations. This file contains a JSON array of comments as strings and test cases as objects. The keys for each test case are:

- `base` : an absolute URL as a string whose [parsing](#) without a base of its own must succeed. This key is always present, and may have a value like `"about:blank"` when `input` is an absolute URL.
- `input` : an URL as a string to be [parsed](#) with `base` as its base URL.
- Either:
 - `failure` with the value `true`, indicating that parsing `input` should return failure,
 - or `href`, `origin`, `protocol`, `username`, `password`, `host`, `hostname`, `port`, `pathname`, `search`, and `hash` with string values; indicating that parsing `input` should return an URL record and that the getters of each corresponding attribute in that URL's [API](#) should return the corresponding value.

The `origin` key may be missing. In that case, the API's `origin` attribute is not tested.

In addition to testing that parsing `input` against `base` gives the result, a test harness for the `URL` constructor (or similar APIs) should additionally test the following pattern: if `failure` is true, parsing `about:blank` against `input` must give failure. This tests that the logic for converting base URLs into strings properly fails the whole parsing algorithm if the base URL cannot be parsed.

setters_tests.json

`resources/setters_tests.json` is self-documented.

toascii.json

`resources/toascii.json` is a JSON resource containing an array where each item is an object consisting of an optional `comment` field and mandatory `input` and `output` fields. `input` is the domain to be parsed according to the rules of UTS #46 (as stipulated by the URL Standard). `output` gives the expected output of the parser after serialization. An `output` of `null` means parsing is expected to fail.

URL parser's encoding argument

Tests in `/encoding` and `/html/infrastructure/urls/resolving-urls/query-encoding/` cover the encoding argument to the URL parser.

There's also limited coverage in `resources/percent-encoding.json` for percent-encode after encoding with `percentEncodeSet` set to special-query percent-encode set and `spaceAsPlus` set to false. (Improvements to expand coverage here are welcome.)

Specification

The tests in this directory assert conformance with [the URL Standard](#).