Client Side API - what is it & requirements

Frederick Hirsch
22 April 2015

Charter

Client-side API: A script interface and events to ease the creation of annotation systems in a browser, a reading system, or a JavaScript plugin

Implied requirements

- *Script* interface: JavaScript defined with WebIDL
- *Events*: HTML5 events
- In a *browser*: asynchronous interfaces using promises
- *Reading system*: offline functionality, s*ynchronization*
- *JavaScript plugin*: incremental deployment strategy

Core functionality

- Methods to enable creating, retrieving, editing, deleting annotations *associated with page content*
- Methods to find targets, do robust anchoring *on web page*
- Methods to support using web annotation protocol
- Annotation Sets: Create, Retrieve, Modify, Delete
- Local persistence, caching, synchronization
- Integrated Authentication, authorization, confidentiality, integrity
- Search: By target - domain, URL, type ; By creator; By date; Body content; other

Considerations

- Sensible HTML5 integration
- Optional : Client API not required for annotations - server side implementation possible
- Design to ease server side implementation - eg limitations on hosted apache servers
- Conveniences vs essential core functionality
- Enable user driven privacy and legal statements
- Accessibility, internationalization

Questions

- UI implementation support (?)
- Social support, including Activity Streams 2.0 integration (?)
- Discovery
- Federation

Detailed requirements

- From annotation use cases - http://www.w3.org/TR/dpub-annotation-uc/
- “Ability to target a point within a Publication
- Ability to target a range of characters within a Publication's text
- Ability to target an Embedded Resource
- Ability to target a segment (point, range, area, volume) of an Embedded Resource
- Ability to target alternate accessibility representation of an Embedded Resource
- Ability to target multiple resources with a single Annotation
- Ability to provide multiple resources as the body of a single Annotation
- Ability to specify that only one of the resources is required or should be rendered
- Ability to specify that all of the resources are required or should be rendered, perhaps in a particular order
- Ability to associate style information with bodies and targets
- Ability to associate timestamps with bodies and targets to determine the appropriate representation
- Ability to associate HTTP request information with bodies and targets to determine the appropriate representation
- Ability to have robust annotations that reference the same work in different media types