Web Engineering Questions

Brad Robertson

May 2, 2023

1 Lecture 1

How are the terms HyperText and HyperMedia defined?

Hypertext is text which contains links to other texts. HyperMedia is a term used for hypertext which is not constrained to be text: it can include graphics, video and sound.

What initiatives and tools are considered the precursors of the Web?

Project Xanadu led by Ted Nelson

What are the steps and stakeholders involved in W3C's standardization process?

"Hosted" by MIT (USA), ERCIM (France), Keio University (Japan), and Beihang University (China)

Main standardization body for the Web through a process:
1. Expression of interest by members/the public
2. When interest reaches sufficient mass, a new
Activity or Working Group is formed on the topic
3. The charter for the Working Group is defined, if necessary
4. The Working Group produces specification(s) and guideline(s) in revision/review cycles
5. The Advisory Committee decides if/when they can be published as Recommendations

2 Lecture 2

What do the acronyms ISP, POP, and IXP stand for and what is their purpose?

- ISP: Internet Service Provider, company that offers internet connectivity.
- POP: Points of Presence. Location where customer packets enter the fully digital ISP network.
- IXP: Internet eXchange Points. Peering ISP Networks

How are the terms URI, URL, and URN defined, what is their purpose, and what is the relation between them?

URI: The Web's global identifier for referencing an abstract or physical resource.
Can be a URL (Locator - uses network location), URN (Name - independent of location), or both.

What is the difference between a proxy, a gateway, and a tunnel server?

> Proxy: A program that can act on behalf of an origin server
> Gateway: A server that acts as intermediary for some other server.Unlike proxy, client interacts with gateway as if it were the origin server for the request
> Tunnel: Program that acts as a "blind" relay between two connections.Tunnel only routes messages, i.e. it does not peek at them

Under which conditions are HTTP requests safe/idempotent? Which HTTP methods are considered safe/idempotent?

HyperText Transfer Protocol (HTTP) is a simple request/response protocol.
Safe requests are non state-altering.GET, HEAD, OPTIONS (and TRACE) are considered to be safe.
Idempotent requests are side-effect free. GET, HEAD, OPTIONS, PUT, DELETE, are idempotent.

What are the differences between PUT and POST in terms of request URI semantics, and pragmatics (i.e. how they are to be used)?

URI in a POST identifies the resource, i.e. the service that will handle the enclosed entity.
URI in a PUT identifies the entity enclosed within the request

Which proxy types are common on the Web, and what is their function?

> Forward Proxy a.k.a. Proxy: external facing stand-ins for the origin server(s) (N:1 relation to origin server)
> Reverse proxy: internal facing proxy, data has single entry point usually with load balancing between multiple origin servers (1:N to origin servers)
> Web Accelerator: proxy with prefetching + compression, speeding up encryption, image manipulation, etc.

What are the benefits of caching?

> Cache is storage for temporary persisting response messages. Purpose is to improve response time for requesting same response message.
> Reduces network bandwidth usage : cost reduction for both content providers and consumers. Decreases perceived delays on the user side : increases user-perceived content value.
> Removes load from origin server : reduces infrastructure/support costs on provider side, allows for faster serving of non-cached resources on client side

How is the principle of Semantic Transparency for HTTP defined? Under which conditions is it violated by the use of caches?

> If resource on origin server is changed, the copy stored in the cache is outdated:
cache invalidation.
> HTTP fundamental design principle: Semantic Transparency
The usage of cache (or any proxy server) must have no impact on client or origin server
> Each request produces the same response as if the request would have been served directly by the origin server itself
>Deviations only tolerable on explicit request from client or origin server, or a warning must be produced.

How can a server stop a proxy or client from caching a response?

> Server can request that no copy of response is cached E.g. if response is known to dynamically change at each request
>Server simply specifies a past date as expiry date
>Cache-Control header field is set to must-re-validate

How can a cache determine if it is fresh in the absence of an Expires header? If Cache-control is missing too?

In absence of explicitly defined expiry of responses, caches may determine validity heuristically, based on Last-Modified header (mandatory) – if no change in recent past then probably no change in immediate future either.

How does the Basic authentication scheme for HTTP work? Under which conditions should it be used?

> Basic scheme transmits credentials as user ID/password pairs in base64 encoding i.e. not encrypted. Insecure as credentials are in plain text
> HTTPS (HTTP over TLS) assumed to be used for security purposes
> HTTP clients used to allow credential embedding in the URL i.e. https://username:password@www.example.com/ This is deprecated and restricted by modern browsers

How do CDNs work and what benefits do they offer?

> Content Delivery Networks - "Inverse"/edge proxy
> Provider puts a copy of a resource on CDN node
> CDN copies it on multiple local nodes
>Client's requests for it are directed to nearest node to use

<u>CDN Benefits</u>

> Reduced latency (less network hops)
> Scaling to demand
> Increased reliability
> Abstraction from data transfer
> Security against DDOS attacks

3 Lecture 3

Which constraints define the REST architectural style, and how are they related with each other?

REpresentational State Transfer (REST)

> An architectural style (set of constraints) for building large-scale distributed hypermedia systems.
1. Resource Identification e.g. through URIs
2. Uniform Interface e.g. through HTTP verbs
3. Self-describing Messages: decouple resource from representations
4. Hypermedia as the Engine of Application State (HATEOAS): links between resources
5. Stateless Interactions: separation of resource state from client state

How are the REST style constraints related to its goals?

> Performance
• [RI+UI] Caching is enabled → "the closer the cache, the faster the response"
> Scalability
• Caching allows serving requests without accessing origin server
• [HATEOAS+SI] All required state is contained in request
> Simplicity
• [UI] Standardized interface to all resources allows of simpler interaction design
> Data independence
• [SDM] Different formats are available for each resource
• Content can be tailored to client's capabilities

What are the maturity levels in Richardson's model? Which REST principles are they related to?

Level 3: Hypermedia Controls (4) + (5)
Level 2: HTTP Verbs (2) + (3)
Level 1: Resources (1)
Level 0: Plain Old XML

What are the principles that should govern the design of RESTful APIs?

>Information abstraction of a key element constitutes a resource
> Resource representation is a sequence of bytes, plus representation metadata; the representation is negotiable
> All interactions are context-free i.e. state agnostic
> Components can perform only a small set of well-defined methods
> Idempotency of operations and representation metadata is encouraged
> Presence of intermediaries is promoted

What steps should be followed for the design of a RESTful API according to Masse's book?

Define data model (+users)
Split data into resources
Name resources with URIs
Define supported uniform interface subset
Design representations (accepted)
Design representations (served)
Link between resources
Model typical sequence of events
Define error conditions

4 Lecture 4

What are the functionalities encapsulated in each of the application layers in the respective pattern by Fowler?

Application Layers

Presentation Layer

> Rendering of data
> Reaction to events
> Communication/conversation logic e.g. sessions

Application Logic Layer

> Functions offered via presentation layer
> Logic performed by the application

Resource Layer

> Logic to access data needed by application logic
> Database, files, content, queues, other support functions

What is the relation between layers and tiers? Which architectural decision is important for this decision?

Tiers are physical organization units
> Client-Server as a 2-tier (at least) model

Layers as functional organization units
> Splitting/aggregating layers in one or more tiers results into different topologies
> Key decision: how much functionality is "shipped" to the client. Thin client model has server handling 3 layers wheras thick client model has client handling at least one of the layers.

Where does the application split lie in the case of remote presentation/distributed/remote data applications? Name examples of such types of applications

Remote-Presentation Applications

> Thick client model
> Single Page Applications (SPAs)
> Multi-device Web applications
> Built on top of APIs - client handles presentation

Distributed Applications

> Thick client model
> Single Page Applications (SPAs)
> Many current Web applications and sites
> Application logic on client delegates (some) processing to function on server, client handles presentation and a bit of application logic.

Remote-Data Applications

> Thick client model
> Mobile/platform games
> Access via high-level interface to resources

5 Lecture 5

What is the difference between text elements with semantics and those without in HTML? Which ones are recommended to be used, and why?

HyperText Markup Language (HTML) is the default language of the Web
, , etc. have semantics defined by the language itself
> Enforced by the browser
> Understood by search engines, third-party viewers like screen readers, etc.
> Up to HTML5 presentational elements like <b> (bold), <i> (italics), and <u> (underlined) were also allowed but without semantics i.e. results may vary
> HTML5 gave them semantic roles, but they should be avoided for accessibility purposes.

How was embedded video and audio handled up to HTML5? How are they handled by HTML5? What are the implications of this mechanism?

Up to HTML5 embedded video/audio handled by non-native Web technologies (e.g. Flash)
> HTML5 defines native mechanisms for media
> Note: effectively breaks compliance to SGML because they enforce control instead of markup.

What are the types of HTML form validation available, and when are they to be used?

Types of validation:
• Client-side, i.e. in the browser before submitting
> Through JavaScript
> By means of built-in form validation as of HTML5
• Server-side, i.e. through application logic
> Less user friendly/reactive
> Supported by the vast majority of server-side frameworks

How is the cascade mechanism used to resolve conflicts between CSS rules?

> Cascade mechanism resolves potentially conflicting rules selecting the same element:
1. Importance (through !important attribute)
2. Specificity (how many elements could be matched)
3. Source Order (later rules in the sheet win)

What is the order of specificity in for CSS rules?

CSS Specificity (less to more)

Element
Class/Attribute/Pseudo-class
Element ID
Style attribute

Under which conditions would you recommend the use of a SPA-style application?

Single Page Applications (SPAs) are heavily client-side dynamic Web pages. They are well-suited to applications with Rich, Complex User Interface Requirements

What options are available for creating server-side dynamic Web pages?

> Content Management Systems (CMS) such as Wordpress
> Static site generators, e.g. Jekyll

How is the MVX pattern defined for JavaScript-using Web applications, where X= C, P, VM?

- MVC: Model View Controller (e.g. Django, Angular)
- MVP: Model View Presenter (e.g. Backbone.js)
- MVVM: Model View ViewModel (e.g. Vue)