

Web Services

Dominic Duggan
Stevens Institute of Technology

1

1

REPRESENTATIONAL STATE TRANSFER (REST)

2

2

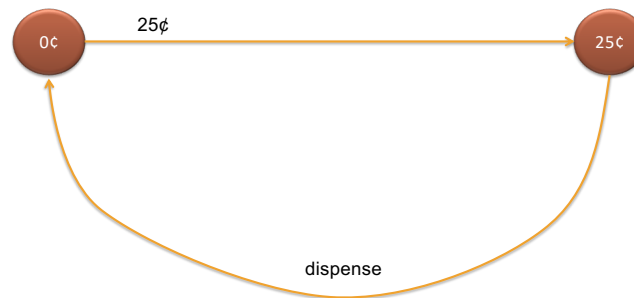
Representational State Transfer

- Software architecture for the Web
- Web browsing as navigation of hypermedia network

3

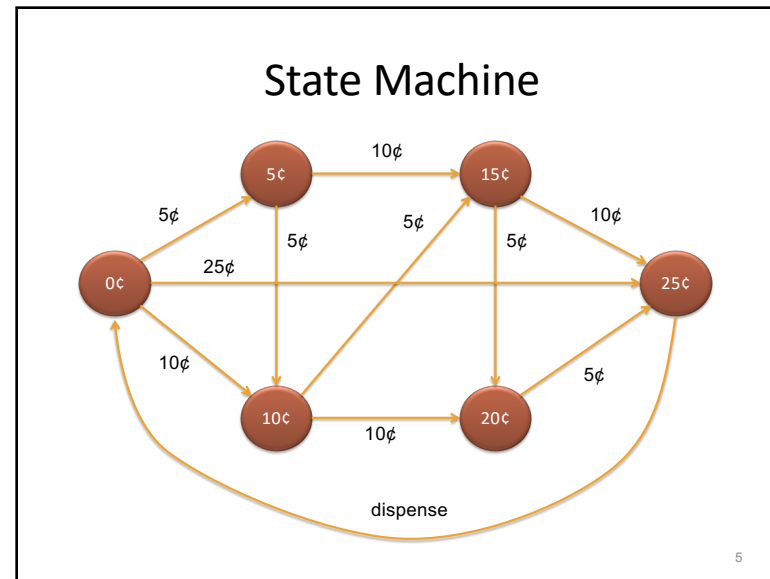
3

State Machine

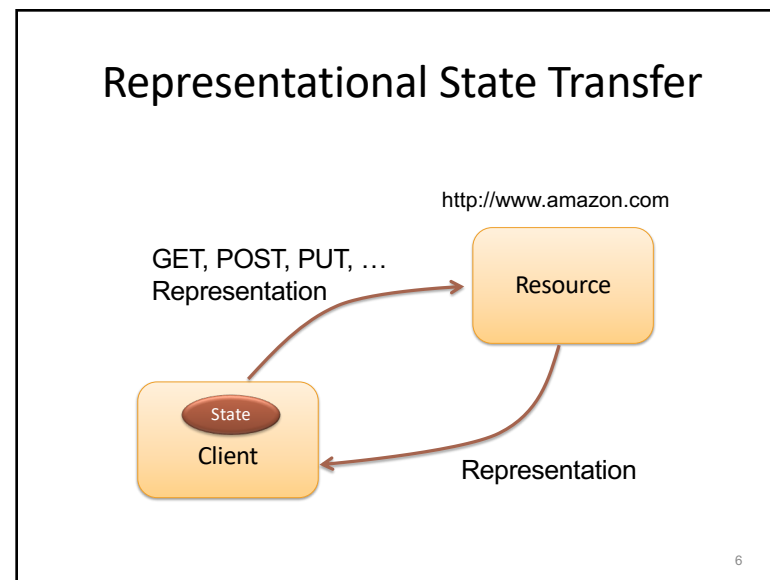


4

4

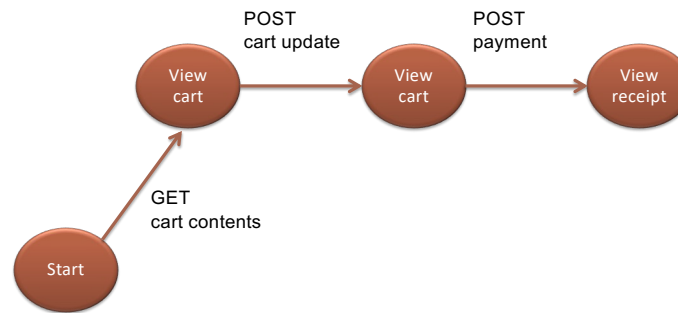


5



6

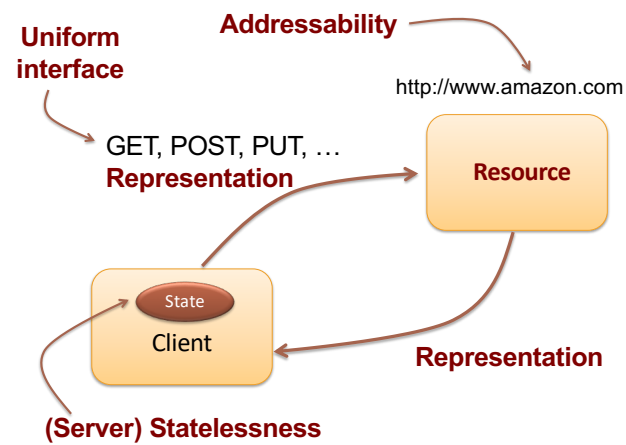
Client as State Machine



7

7

Representational State Transfer



8

8

Representational State Transfer

- Originally a software architecture for the Web
- Emerged as an alternative architecture for Web services
 - Resource-oriented architecture

SOAP / WSDL	REST
Service (operation) oriented	Resource oriented
One endpoint URL	URL for each individual resource
Application-defined verbs	Fixed set of HTTP verbs

9

REST Verbs

- Retrieve: HTTP GET
- Create:
 - HTTP PUT for new URI or
 - HTTP POST for existing URI (server decides result URI)
- Modify: HTTP PUT to existing URI
- Delete: HTTP DELETE

- Partial update: HTTP PATCH
- Record request in response: HTTP TRACE
- Tunnel from proxy to server: HTTP CONNECT
- Retrieve metadata only: HTTP HEAD
- Check which methods are supported: HTTP OPTIONS

10

10

Example: Amazon Simple Storage Service (S3)

- S3 is based on two concepts
 - Buckets
 - Named container
 - Objects
 - Named piece of data, with metadata
 - Stored in buckets

11

11

S3 RPC Interface

- Object-oriented interface to S3
 - CreateBucket
 - ListAllMyBuckets
- Getter/setter methods on bucket and object “objects”
 - S3Object.name()
 - S3Object.setValue()
 - S3Bucket.getObjects()

12

12

S3 REST Interface

- Three types of resources
 - List of your buckets
`https://s3.amazonaws.com`
 - A particular bucket (virtual host)
`https://name-of-bucket.s3.amazonaws.com`
 - A particular s3 object inside a bucket
`https://name-of-bucket.s3.amazonaws.com/name-of-object`

13

13

S3 REST Interface

- Example:
 - A particular bucket
`https://jeddak.s3.amazonaws.com`
 - A particular s3 object inside a bucket
 - Object names:
`docs/manual.pdf`, `docs/security.pdf`, `talks/snt.pdf`
 - Resource URIs:
`https://jeddak.s3.amazonaws.com/docs/manual.pdf`
`https://jeddak.s3.amazonaws.com/docs/security.pdf`
`https://jeddak.s3.amazonaws.com/talks/snt.pdf`

14

14

S3 REST Interface

- Use HTTP methods as verbs

Verb	Bucket list	Bucket	Object
GET	List buckets	List bucket objects	Get value and metadata
HEAD			Get metadata
PUT		Create bucket	Set object value and metadata
DELETE		Delete bucket	Delete object

15

HTTP

16

16

HTTP Request

GET /index.html HTTP/1.1

Host: www.example.org

...request headers...

17

17

HTTP Response

HTTP/1.1 200 OK

Date: Mon, 1 May 2011 21:38:14 GMT

Server: Apache/1.3.34 (Debian) mod_ssl/2.8.25
OpenSSL/0.9.8c ...

Last-Modified: Wed, 25 Nov 2009 12:27:01 GMT

ETag: "7496a6-a0c-4b0d2295"

Accept-Ranges: bytes

Content-Length: 2572

Content-Type: text/html

Via: 1.1 www.example.org

Vary: Accept-Encoding

...

18

18

Request Headers

- **Accept:** for content negotiation
 - Content-Type: response header e.g.
ATOM (application/atom+xml)
RDF (application/rdf+xml)
XHTML (application/xhtml+xml)
Form-encoded key-value pairs (application/x-www-form-urlencoded)
- **Authorization:** app-defined auth info
 - WWW-Authenticate: response header with status code of 401 ("Unauthorized")

19

19

Request Headers

- **Cookie:** (non-standard)
 - Save-Cookie: to save cookie on client

20

20

Response Headers

- **Last-modified**: time of last modification
 - If-Last-Modified: request header for caching
- **Etag**: hash of metadata
 - If-None-Match: request header for caching
- **Cache-Control**: how long to cache
- **Upgrade**: upgrade protocol e.g. http to https
- **Location**: URI for newly created resource, redirection, ...

21

21

Response Codes

- **1XX**: for negotiation with Web server
 - E.g. 101 ("Switching protocols") with Upgrade: response header
- **2XX**: to signal success
 - E.g. 200 ("Success"), 201 ("Created"), ...
- **3XX**: redirect clients
 - E.g. 303 ("See other"), 307 ("Temporary redirect")
- **4XX**: client errors
 - E.g. 400 ("Bad request"), 404 ("Not found"), 401 ("Unauthorized"), 403 ("Forbidden")
- **5XX**: server errors
 - E.g. 500 ("Internal server error")

22

22