
RESTful Web Services

Representational State Transfer (REST)

- An architectural style
  - Can be implemented in many different ways
- REST is the underlying architectural principle of the World Wide Web
- Clients (browsers) can operate without knowing anything about:
  - The server
  - The resources it hosts
- Key constraint: server and client must both agree on the media used
  - HTML for the Web
- If something conforms to REST principles: RESTful

Implementing REST with HTTP

- Resources are manipulated with the HTTP verbs
  - POST, GET, PUT, DELETE, …
- HTTP verbs map to CRUD:
  - Create, Read, Update, Delete
- Rely on the fact that some verbs are idempotent
  - GET, PUT, DELETE are free from side-effects
- Provide responses that are self descriptive and link to other resources as necessary
  - Hypermedia As The Engine Of Application State
  - HATEOAS

Features of REST (with Web examples)

- Resources identified by a persistent identifier
  - URI
- The representation retrieved for a resource is dependent on the request and not the identifier
  - Use Accept headers to control whether you want HTML, XML, JSON, …
- Maintain state in the object and display that state in the representation

Features of REST (with Web examples)

- Embed the relationships between resources in the representation of the resource
  - Links between objects are embedded directly in the representation
  - HTML: …

- Resource representations describe how the representation can be used and under what circumstances it should be discarded/re-fetched in a consistent manner
  - HTTP Cache-Control headers

HATEOAS in Spring

- As with most things, Spring does a lot for you
- If you use @Entity and CrudRepository Spring can:
  - Serve your data automatically in JSON format
  - Provide links where needed
- Quick exercise:
  - Run Eventlite. In a terminal try (all one line):
    - `$ curl -H "Accept: application/json" http://localhost:8080/api`
  - Follow the links
    - The code has an EventsControllerApi class
    - Have you implemented an API class for venues?
      - So how does it work?