REST

1. REST
   1. Representational state transfer
      1. The term comes from Roy Feilding’s PHD work
      2. More practical than theoretical
      3. It is an architectural style, not a technology, nor an architecture
   2. Software architecture = {Element, Constraints, Rationale}
      1. An architecture style abstracts elements and constraints from various specific architectures
      2. Analogy: designs and design pattern
2. Uniform interface
   1. Resources are identified by ONE resource identifier mechanism
   2. Access methods are the same for all resources
   3. Resources are manipulated by exchanging representations
   4. Representation are in self-descriptive message
   5. Hypermedia acts as the engine of application state
3. REST rationale
   1. Maximize reuse
   2. Minimize coupling
   3. Eliminate partial failure condition
   4. Scale without bound
   5. Simplify
4. Generic process to design RESTful service
   1. Identify resources
   2. Design URIs
   3. Expose a subset of the uniform interface
   4. Design the representations from and to the client
   5. Integrate this resource into existing resources, using hypermedia links and forms
   6. Consider typical flows
   7. Consider error condition
5. Web maturity model
   1. 0: the swamp of pox
   2. 1: resources
   3. 2: http verbs
   4. 3: hypermedia controls
   5. Glory of REST
6. CRUD
   1. CREATE
   2. READ
   3. UPDATE
   4. DELETE