**Large-Scale Programming and Testing**

Fall 2017 – CSCI 4963/6963 – Week 11  
David Goldschmidt – goldschmidt@gmail.com  
Office: Amos Eaton 115  
Office hours: Mon/Thu 1:00-1:50PM; Wed 1:00-2:50PM

---

**RESTful interfaces**

- REpresentational State Transfer (REST) is an architectural approach to distributed hypermedia systems
- REST was first described in 2000 in Roy Thomas Fielding’s PhD dissertation:

> “The complexity of modern software systems has necessitated a greater emphasis on componentized systems, where the implementation is partitioned into independent components that communicate to perform a desired task.”
Distributed or networked systems?

- What are the differences between distributed and network-based systems?

  - A distributed system treats its users as being part of an “ordinary” centralized system that just happens to be running on multiple CPUs
  - Such distribution is transparent to users

  - A network-based system performs operations across a network, often with specialized components defined within the network
  - Such distribution is not necessarily transparent to users

REST defines a uniform interface

- By adhering to a uniform interface, the system architecture is simplified and flexible to support a wide variety of applications

- Implementation can proceed (and change) in a manner that is largely independent of the defined API(s)

- The key disadvantage is reduced efficiency

“Implementations are decoupled from the services they provide, which encourages independent evolvability.”
REST is stateless

- In a RESTful interface, each client request must contain all of the necessary information for that request
- On the server side, no session information is maintained
  - Therefore, it is up to each client to maintain its own session state, if necessary
- Advantages of being stateless include reliability and scalability
- The key disadvantage is (again) reduced efficiency

REST responses may be cacheable

- Given the disadvantage of reduced efficiency, server responses identify whether results can be cached
- If so, clients can reuse response data for future requests
- In doing so, the number of client requests made should decrease and network traffic is reduced
Representing information

- The key abstraction of information is a resource, which is essentially a named “bit” of information
  - e.g., a student, a document, a query term, an image, a collection, etc.
- Resource methods are defined in the RESTful interface to provide operations on resources
  - While HTTP and JSON are often used, REST does not require this
  - HTTP is useful in that it is text-based, flexible, and provides both read and write functionality between clients and servers
  - JSON is useful in that it is text-based, flexible, and human-readable
- Helpful URL: https://jsonplaceholder.typicode.com/