Django: REST in Peace
A RESTful approach to Django Web

Shreyank Gupta/sgupta@redhat.com

September 18, 2011
REST Architecture
- Using resource identifiers (URLs) to represent resources
- Transferring representations of the state of that resource between system components.
Using resource identifiers (URLs) to represent resources
Transferring representations of the state of that resource between system components.
Using resource identifiers (URLs) to represent resources

Transferring representations of the state of that resource between system components.
HTTP verbs
5.1.1 Method

The Method token indicates the method to be performed on the resource identified by the Request-URI. The method is case-sensitive.

```
Method = "OPTIONS" ; Section 9.2
| "GET" ; Section 9.3
| "HEAD" ; Section 9.4
| "POST" ; Section 9.5
| "PUT" ; Section 9.6
| "DELETE" ; Section 9.7
| "TRACE" ; Section 9.8
| "CONNECT" ; Section 9.9
| extension-method
```

The list of methods allowed by a resource can be specified in an Allow header field (section 14.7). The return code of the response always notifies the client whether a method is currently allowed on a resource, since the set of allowed methods can change dynamically. An origin server SHOULD return the status code 405 (Method Not Allowed) if the method is known by the origin server but not allowed for the requested resource, and 501 (Not Implemented) if the method is unrecognized or not implemented by the origin server. The methods GET and HEAD MUST be supported by all general-purpose servers. All other methods are OPTIONAL; however, if the above methods are implemented, they MUST be implemented with the same semantics as those specified in section 9.
HTTP verbs
HTTP verbs

- GET
- POST
- PUT?
- DELETE?
HTTP verbs

- GET
- POST
- PUT?
- DELETE?
HTTP verbs

- GET
- POST
- PUT?
- DELETE?
Dirty URLs
Resource centric architecture
Conforming to the REST constraints is referred to as being "RESTful"
Advantages of RESTful

- Clean, self documenting URLs
- Allows you make web-services out of your views more easily
RESTful URLs
GET

- GET /examples/ - List examples
- GET /examples/new/ - New example
- GET /example/1/ - Show example #1
- GET /example/1/edit/ - Edit example #1
POST /examples/ - Create new example
PUT /example/1/ - Update example #1
DELETE /example/1/ - Destroy example #1
Does Django REST?
GET and POST
PUT and DELETE
class HttpMethodsMiddleware
Unobstusive JavaScript
Django: REST in Peace
!?!