APIs REST con NodeJS y MongoDB

CONTENIDO

REQUISITOS

1. **Node.js**
   - http://nodejs.org/download/
2. **MongoDB**
   - http://www.mongodb.org/downloads
3. **MongoDB driver**
   - npm install mongodb
4. **ExpressJS**
   - npm install express
5. **Otros**
   - npm install body-parser

1. REST: Representational State Transfer

1. **Qué es?**
   - Un estilo de arquitectura de software
2. **Qué no es?**
   - Un estándar o protocolo
3. **Quién lo propuso?**
   - Roy Fielding en su tesis doctoral
4. **Transferencia de qué?**
   - De la representación del estado de recursos
5. **Por ejemplo?**
   - World Wide Web, Facebook Graph API

1. REST: Principios y restricciones

1. Recursos y representaciones

- Todo en términos de recursos en general, no de archivos específicos

2. Recursos identificados con URIs

- No: facebook.com/profile.php
- Si: facebook.com/username
- Si: twitter.com/username/status/:tweetID

3. Operaciones sobre recursos con métodos HTTP

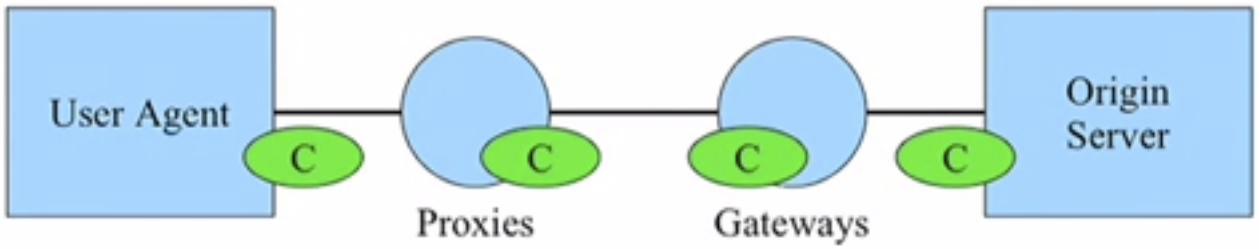
- GET
- POST
- PUT
- DELETE

4. Interfaces uniformes con sustantivos, no verbos

- No: /obtenerSaldoACuenta
- No: /agregarSaldoACuenta
- Si: GET /cuentas/#cuenta/saldo
- Si: POST /cuentas/#cuenta/saldo

1. REST: Beneficios

	Cliente-Servidor	Stateless	Cacheable	Layered
Eficiencia			x	
Escalabilidad		x	x	x
Desempeño percibido	x		x	



2. REST APIs con NodeJS y ExpressJS

1. Servidor HTTP

- api = require('express')()
- http = require('http').createServer(api)

2. Enrutamiento de solicitudes

- api.route(…)
- api.get(…)
- api.post(…)
- api.all(…)
- api.put(…)
- api.delete(…)

3. Procesamiento de las solicitudes

- req.body
- req.params

4. Respuesta a las solicitudes

- res.status(…)
- res.write(…)
- res.setHeader(…)
- res.end(…)

3. REST APIs con NodeJS y MongoDB

1. **Qué es MongoDB?**
   - Una base de datos no relacional (NoSQL)
2. **Cuál es la unidad de almacenamiento**
   - Documentos (vs filas en SQL)
3. **Cuál es la unidad de agrupación?**
   - Colecciones (vs tablas en SQL)
4. **Licencias o regalías?**
   - No, es open source
5. **Ventajas para JavaScript?**
   - Notación muy similar
   - No requiere esquemas (i.e. tablas SQL)
   - Los documentos lucen como objetos

3. REST APIs con NodeJS y MongoDB

1. MongoDB driver

- mongoClient = require('mongodb').MongoClient

2. Conectarse a una base de datos

- mongoClient.connect('mongodb://localhost:27017/databaseName')

3. Acceder a una colección

- db.collection('collectionName')

4. Operaciones CRUD (Create, Read, Update, Delete)

- Create: collection.insert(document)
- Read: collection.find(query)
- Update: collection.update(query, modifier)
- Delete: collection.remote(query)

5. Mongo Shell

- mongod Inicia el proceso de la base de datos MongoDB
- mongo Inicia una interfaz de línea de comandos a MongoDB

Referencias

- **APIGEE**

http://apigee.com/

- **REST API Tutorial**

http://www.restapitutorial.com/

- **ExpressJS**

http://expressjs.com/

- **MongoDB**

http://docs.mongodb.org/manual/core/crud-introduction/
https://github.com/mongodb/node-mongodb-native

/jorgezaccaro

/bogotajs-apis