RESTful APIs with Node.js

Marian Edu, Acorn IT
Why Node.js?
And most recently….

Nobody Use It... really.

And most recently....

#1 Retailer
Why Not Node.js?
• Different technology (JS, really?)
• Not from Progress world (Nativescript, Modulus)
• Single threaded (OE anyone?)
Open source framework designed for APIs

Rapid creation of Models and REST APIs

Model Driven Architecture

Simple ORM system for CRUD

Built on Express
• Models
• Relations
• Methods
• Hooks
Models

- Models define your API
- You describe properties - LoopBack does the REST
- Inheritance + Relationships
- Out of the box - full CRUD + search
Datasources

- Where LoopBack stores crap
- Referred to as connectors as well
- Cloudant, DB2, Memory, MongoDB, MySQL, Oracle, PostgreSQL, Redis, SQL Server
- Roll your own
### Customers

<table>
<thead>
<tr>
<th>Method</th>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/Customers</td>
<td>Create a new instance of the model and persist it into the data source.</td>
</tr>
<tr>
<td>PUT</td>
<td>/Customers</td>
<td>Update an existing model instance or insert a new one into the data source.</td>
</tr>
<tr>
<td>GET</td>
<td>/Customers</td>
<td>Find all instances of the model matched by filter from the data source.</td>
</tr>
<tr>
<td>HEAD</td>
<td>/Customers/{id}</td>
<td>Check whether a model instance exists in the data source.</td>
</tr>
<tr>
<td>GET</td>
<td>/Customers/{id}</td>
<td>Find a model instance by id from the data source.</td>
</tr>
<tr>
<td>DELETE</td>
<td>/Customers/{id}</td>
<td>Delete a model instance by id from the data source.</td>
</tr>
<tr>
<td>PUT</td>
<td>/Customers/{id}</td>
<td>Update attributes for a model instance and persist it into the data source.</td>
</tr>
<tr>
<td>GET</td>
<td>/Customers/{id}/exists</td>
<td>Check whether a model instance exists in the data source.</td>
</tr>
<tr>
<td>GET</td>
<td>/Customers/{id}/invoices</td>
<td>Queries invoices of Customer.</td>
</tr>
<tr>
<td>POST</td>
<td>/Customers/{id}/invoices</td>
<td>Creates a new instance in invoices of this model.</td>
</tr>
<tr>
<td>DELETE</td>
<td>/Customers/{id}/invoices</td>
<td>Deletes all invoices of this model.</td>
</tr>
<tr>
<td>GET</td>
<td>/Customers/{id}/invoices/{fk}</td>
<td>Find a related item by id for invoices.</td>
</tr>
</tbody>
</table>
StrongLoop Arc

A visual suite for the StrongLoop API Platform
**loopback-discovery**

`loopback-discovery` is a command prompt discovery script for the [LoopBack](https://github.com/strongloop/loopback/) framework.

**Installation**

`npm install -g loopback-discovery`

**Basic use**

The script can discover models from data sources defined on a StrongLoop application from command line. This is mainly for overcoming missing support for "non-supported" connectors in StrongLoop's [Arc](https://github.com/strongloop/strong-arc) GUI interface.

- To run model discovery for **sports** data source inside a StrongLoop application available in current folder.

  `$ loopback-discovery`
Exports a Swagger API definition for a LoopBack application.

```
slc loopback:export-api-def
```

Creates a fully-functional application with an API defined using the Swagger 2.0 specification.

```
slc loopback:swagger
```
```json
{
    "swagger": "2.0",
    "info": {
        "version": "1.0.0",
        "title": "akera-sports"
    },
    "basePath": "/api",
    "paths": {
        "/Customers/{id}/invoices/{fk}": {
            "get": {
                "tags": ["Customer"],
                "summary": "Find a related item by id for invoices.",
                "operationId": "Customer.prototype._findById_invoices",
                "parameters": [
                    {
                        "name": "fk",
                        "in": "path",
                        "description": "Foreign key for invoices",
                        "required": true,
                        "type": "string",
                        "format": "JSON"
                    }
                ]
            }
        }
    }
}
```
Fishing Lure

Price: $3.99
IN STOCK

You'll be catching more fish than ever before with this deluxe green and yellow fishing lure.

Frequently Bought Together

Fishing Lure added to cart
Odata

- Open Data Protocol
- REST-based
- content-type
  - AtomPub
  - JSON
  - XML
- www.odata.org
- www.oasis-open.org
OData Providers

Relational Data Resources

WCF Data Services

WCF RIA Services

SharePoint 2010

Connect Any Data Source to Any Application
The Metadata Service provides convenient access to the entity model of a service

```json
{
  "@odata.context":"~/$metadata/$metadata",
  "value": [
    { "name":"References" , "url":"References" },
    { "name":"Schemata" , "url":"Schemata" },
    { "name":"Types" , "url":"Types" },
    { "name":"Properties" , "url":"Properties" },
    { "name":"NavigationProperties" , "url":"NavigationProperties" },
    { "name":"EnumTypeMembers" , "url":"EnumTypeMembers" },
    { "name":"Actions" , "url":"Actions" },
    { "name":"Functions" , "url":"Functions" },
    { "name":"Terms" , "url":"Terms" },
    { "name":"Annotations" , "url":"Annotations" },
    { "name":"EntityContainer" , "url":"EntityContainer",
      "kind":"Singleton" },
    { "name":"EntitySets" , "url":"EntitySets" },
    { "name":"Singletons" , "url":"Singletons" },
    { "name":"NavigationPropertyBindings" , "url":"NavigationPropertyBindings" },
    { "name":"ActionImports" , "url":"ActionImports" },
    { "name":"FunctionImports" , "url":"FunctionImports" }
  ]
}
```
Hypertext As The Engine Of Application State

“People are trying to wrangle their APIs into hypermedia formats or (worse) inventing new ones, all because someone decided that everything must be self-documenting, despite the fact that the consumers of that implicit documentation (machines) have no hope of understanding it.”

Mixing Meta-Data with Data.
"catdescription": "Use what the Pros use. Specify Graphite, Non-Rainforest Wood, or Aluminum. ",
"category1": "Tennis",
"category2": "Equipment",
"catpage": 2,
"itemname": "Tennis Racquet",
"itemnum": 2,
"minqty": 5,
"onhand": 12987,
"onorder": 483,
"price": 119.5,
"reorder": 7,
"special": "2",
"weight": 3,
"orderlines": {
    "__deferred": {
        "uri": "http://localhost:8080/odata/Items(2)/orderlines"
    }
},
"__metadata": {
    "uri": "http://localhost:8080/odata/Items(2)",
    "type": "NODATASERVER.Item"}
Assistants, talk about REST 😊

Amazon Alexa Echo

Amazon Lambda

Google Home

Firebase Functions

REST APIs