Introducing REST APIs
What is REST?
What is REST?
REpresentational State Transfer
What is REST?
REpresentational State Transfer
Six Basic Characteristics
What is REST?

REpresentational State Transfer

Six Basic Characteristics

- Separation between client and server
- Self contained requests (stateless)
- Cacheable
- Layered architecture to allow intermediaries
- Uniform method of communication between client and server
- (Optional) Code on Demand
What is REST?

REpresentational State Transfer

Six Basic Characteristics

- Separation between client and server
- Self contained requests (stateless)
- Cacheable
- Layered architecture to allow intermediaries
- Uniform method of communication between client and server
- (Optional) Code on Demand

Core Ideas

- Communicate with Application Resources
- Generally uses HTTP Requests
HTTP Requests and REST
HTTP Requests and REST

Ways to Modify Resources in REST

You may recognize these as operations from relational databases...

- Create
- Retrieve
- Update
- Delete
HTTP Requests and REST

Ways to Modify Resources in REST

You may recognize these as operations from relational databases...

- Create
- Retrieve
- Update
- Delete

HTTP Requests

<table>
<thead>
<tr>
<th>HTTP Request</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>Create an item/send information</td>
</tr>
<tr>
<td>GET</td>
<td>Retrieve items from the server</td>
</tr>
<tr>
<td>PUT</td>
<td>Update a resource</td>
</tr>
<tr>
<td>DELETE</td>
<td>Remove a resource</td>
</tr>
</tbody>
</table>
Cool. Why should I care?
Cool. Why should I care?

Flexible

- Unlike SOAP, not just contained to XML
- Allows easy integration with external tools
Cool. Why should I care?

Flexible
- Unlike SOAP, not just contained to XML
- Allows easy integration with external tools

Easy to Build/Use
- Pretty uniform once you know the basics
- Lots of frameworks to help you out (we're going to use Flask)
- Extremely scalable and lightweight
Cool. Why should I care?

Flexible

- Unlike SOAP, not just contained to XML
- Allows easy integration with external tools

Easy to Build/Use

- Pretty uniform once you know the basics
- Lots of frameworks to help you out (we're going to use Flask)
- Extremely scalable and lightweight

It's Fun!

Most of the services you use have an API that can be used to do cool things!
Enough talking.
Let's do something cool.

Grab a copy of the starter code.