REST Testing

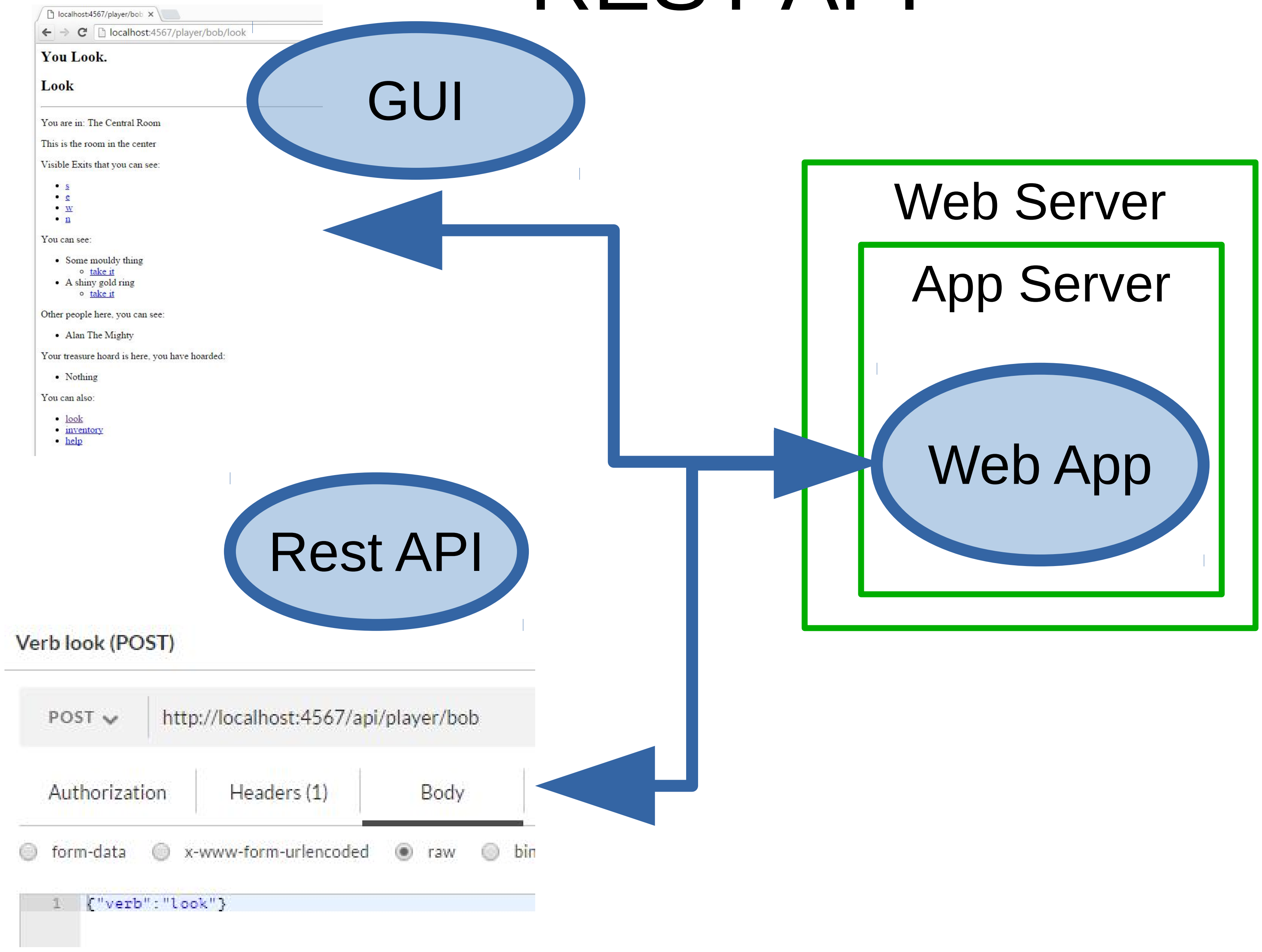
What is “REST”

“REpresentational State Transfer”

What is REST?

- API (Application Programming Interface) using HTTP calls where the VERBs dictate the action and the URI the thing to act on
- In theory each call is stateless and cacheable, but not always
- Most people use it to mean an API accessible via HTTP on URLs using JSON or XML as the message and response data

REST API



localhost:4567/player/bob
localhost:4567/player/bob/look

You Look.

Look

You are in: The Central Room

This is the room in the center

Visible Exits that you can see:

- s
- e
- w
- n

You can see:

- Some mouldy thing
  - take it
- A shiny gold ring
  - take it

Other people here, you can see:

- Alan The Mighty

Your treasure hoard is here, you have hoarded:

- Nothing

You can also:

- look
- inventory
- help

Verb look (POST)

POST http://localhost:4567/api/player/bob

Authorization | Headers (1) | Body

form-data | x-www-form-urlencoded | [✓ raw] | bin

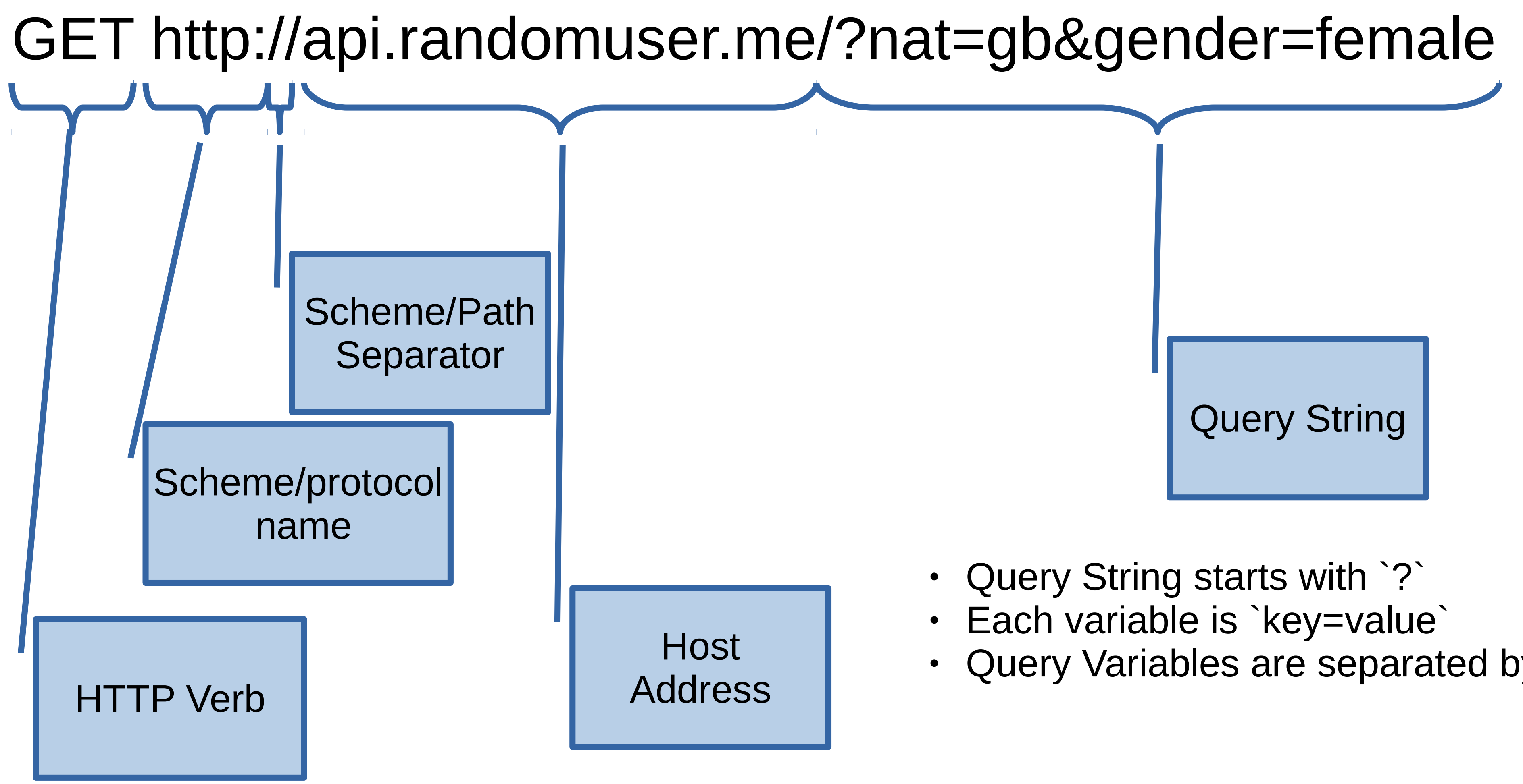
```
{"verb":"look"}
```

RandomUser.me Example

GET http://api.randomuser.me/?nat=gb&gender=female

- A GET request on randomuser.me API
  - We can issue API GET Requests through a browser
- Query parameters specify
  - Nationality Great Britain
  - Gender Female

URI Example



- Query String starts with `?`
- Each variable is `key=value`
- Query Variables are separated by `&`

See also https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

Example JSON Response

GET http://api.randomuser.me/?nat=gb&gender=female

- JSON – JavaScript Object Notation
- http://www.json.org/

```
{"results":[{"gender":"female","name":
{"title":"miss","first":"phoebe","last":"graham"},"location":
{"street":"1257 kings road","city":"leicester","state":"county
down","postcode":"UX31
2LQ"},"email":"phoebe.graham@example.com","login":
{"username":"beautifulbird315","password":"thong","salt":"17RIVsok","md5
":"ac41371d469204fac73584a7c39ef7e2","sha1":"18612f6e2e8f2d4d2b94a5adc44
6e7d854e391f6","sha256":"7821627d118bfe7de53fba4a81766f4ff05cafc8b11d97a
fa5cbafa8b24f77c0"},"registered":1345700664,"dob":769204756,"phone":"011
6164 012 5787","cell":"0776-993-939","id":{"name":"NINO","value":"RJ 04
74 93 I"},"picture":
{"large":"https://randomuser.me/api/portraits/women/92.jpg","medium":"ht
tps://randomuser.me/api/portraits/med/women/92.jpg","thumbnail":"https:/
/randomuser.me/api/portraits/thumb/women/92.jpg"},"nat":"GB"}],"info":
{"seed":"38d3dc86583143fc","results":1,"page":1,"version":"1.0"}}
```

HTTP References

- URI
  - en.wikipedia.org/wiki/Uniform_Resource_Identifier
- HTTP
  - en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

REST References

- Microsoft API Guidelines
  - https://github.com/Microsoft/api-guidelines
    - https://github.com/Microsoft/api-guidelines/blob/master/Guidelines.md
- Rest API Tutorial
  - http://www.restapitutorial.com/
- Original Dissertation that defines REST
  - http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- JSON
  - http://www.json.org/

Exercise

Explore a REST API using a Browser

- https://randomuser.me/documentation
- Make some default API calls by typing http://api.randomuser.me/ into a browser
- Use the documentation and:
  - GET 10 random people from Germany
  - GET 2 random males from France
  - GET name, email and phone number for 25 people
  - GET results as a csv list

Sample Answers

- GET 10 random people from Germany
  - http://api.randomuser.me/?results=10&nat=DE
- GET 2 random males from France
  - http://api.randomuser.me/?results=2&nat=FR&gender=male
- GET name, email and phone number for 25 people
  - http://api.randomuser.me/?results=25&inc=name,email,phone
- GET results as a csv list
  - http://api.randomuser.me/?results=25&inc=name,email,phone&format=csv

REST Client

What is a REST Client

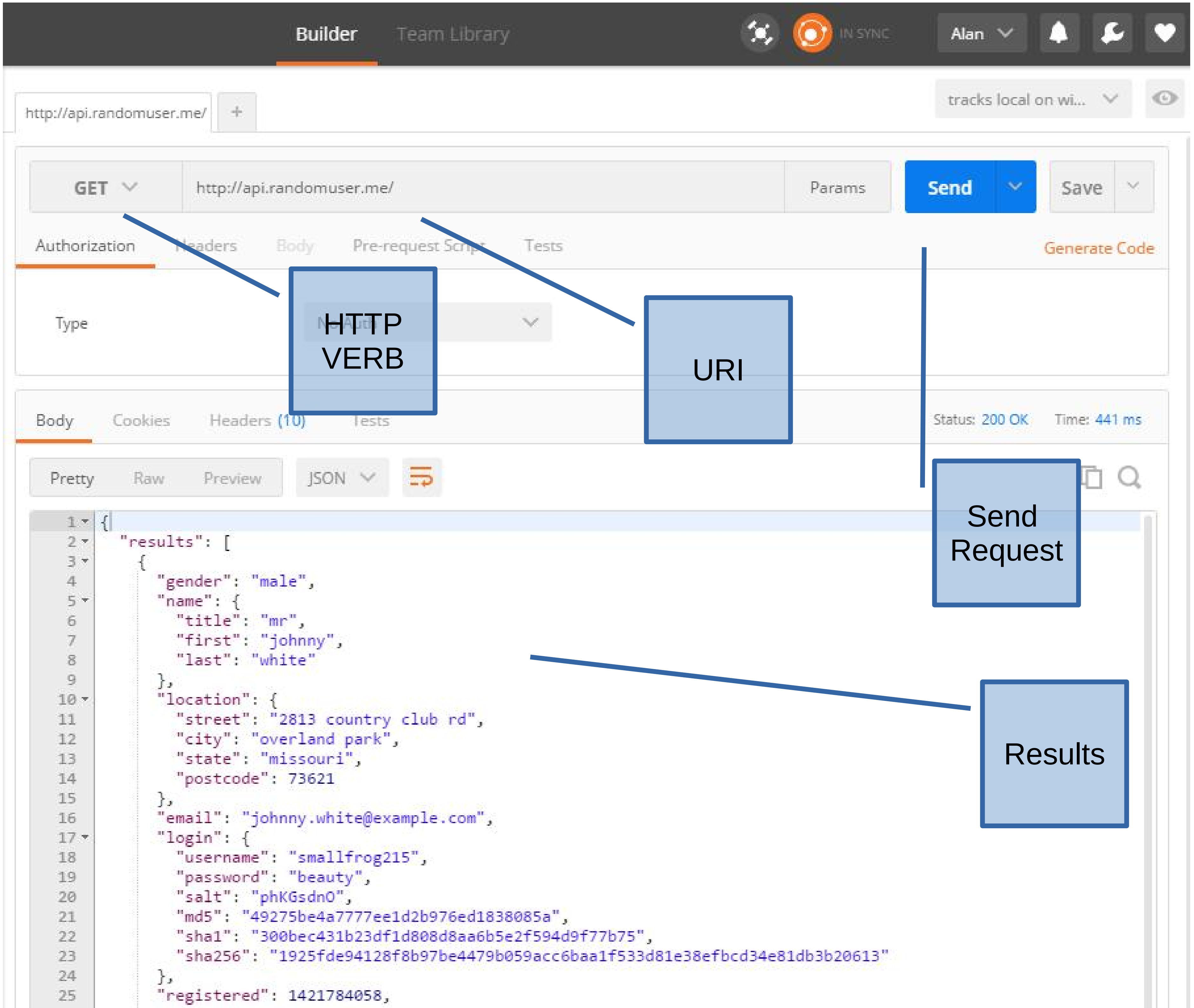
- A GUI that will allow us to interact with a REST API
- Easily issue REST API calls with different API Verbs
- There are a lot of RestClients Available

PostMan REST Client

- Built on Chrome
- Runs on Windows, Linux, Mac
- Free
- Visit https://www.getpostman.com/ using Chrome Browser
  - Install “Chrome App”
  - Visit “chrome://apps”
  - Run Postman

Using Postman

- You can create an account on Postman or use without an account
- Account is Free and allows you to share Postman collections
  - Lists of requests and templates for APIs



```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "johnny",
        "last": "white"
      },
      "location": {
        "street": "2813 country club rd",
        "city": "overland park",
        "state": "missouri",
        "postcode": 73621
      },
      "email": "johnny.white@example.com",
      "login": {
        "username": "smallfrog215",
        "password": "beauty",
        "salt": "phKGsdnO",
        "md5": "49275be4a7777ee1d2b976ed1838085a",
        "sha1": "300bec431b23df1d808d8aa6b5e2f594d9f77b75",
        "sha256": "1925fde94128f8b97be4479b059acc6baa1f533d81e38efbcd34e81db3b20613"
      },
      "registered": 1421784058,
```

Exercise: Use Postman

- Repeat the previous exercises with Postman using GET requests on api.randomdata.com
- GET the results in XML
- GET the results in YAML
- GET the results in CSV

Answers: Using Postman

- GET 10 random people from Germany
  - http://api.randomuser.me/?results=10&nat=DE
- GET 2 random males from France
  - http://api.randomuser.me/?results=2&nat=FR&gender=male
- GET name, email and phone number for 25 people
  - http://api.randomuser.me/?results=25&inc=name,email,phone
- GET results as a csv list
  - http://api.randomuser.me/?results=25&inc=name,email,phone&format=csv

Answers: Using Postman

- Postman tries to format the results for you
- XML and JSON
  - Pretty Printed
  - Expand/Collapse Sections
    - http://api.randomuser.me/?results=25&inc=name,email,phone&format=yaml
    - http://api.randomuser.me/?results=25&inc=name,email,phone&format=xml
    - http://api.randomuser.me/?results=25&inc=name,email,phone&format=json

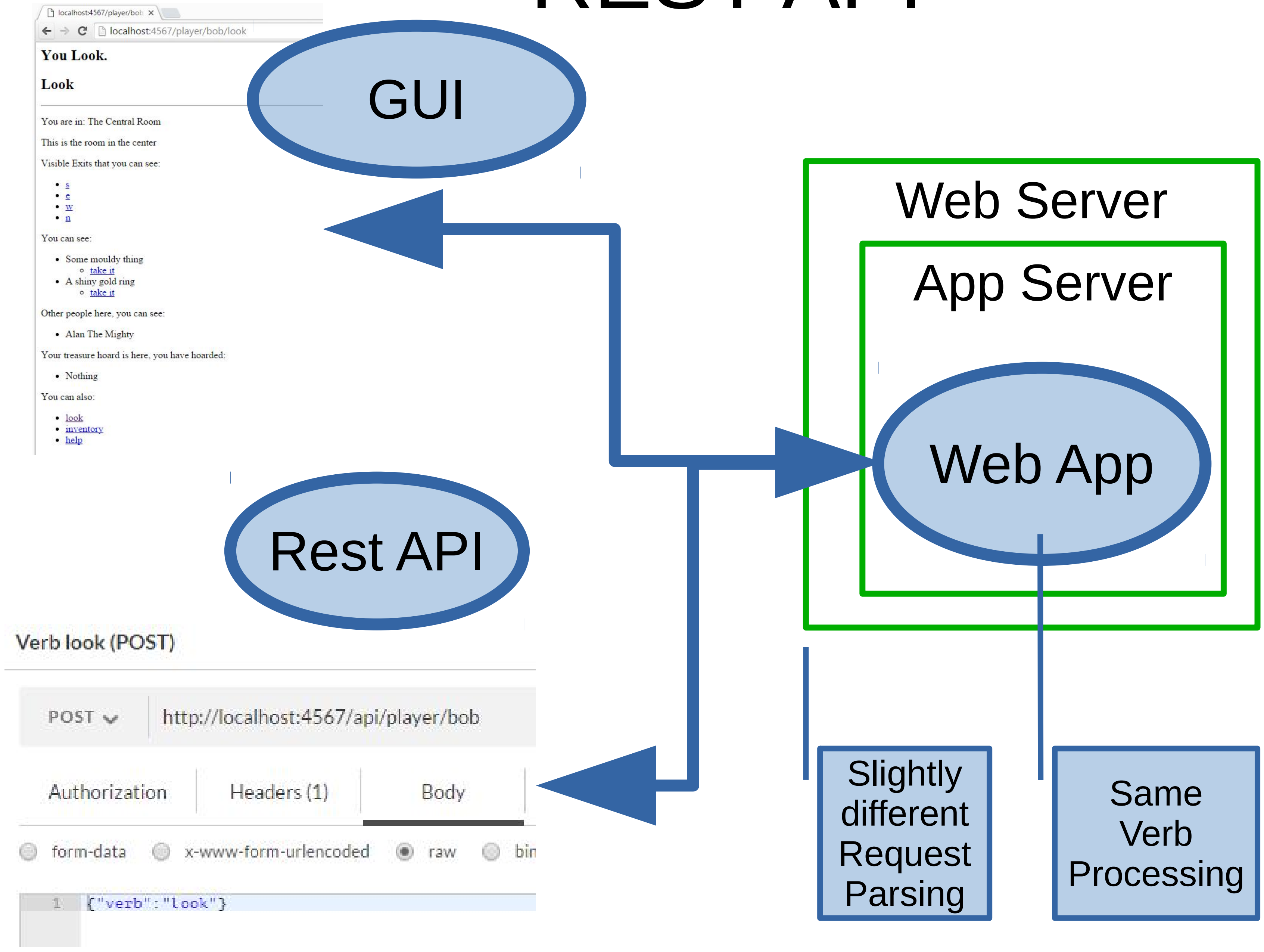
Postman Lessons Learned

- Gotchas
  - Shared Cookies between Chrome sessions
  - Shared Proxy Settings between Chrome sessions
  - Postman shows you what it 'thinks' it sent – use a proxy to be sure
- Lessons Learned
  - I tend to always use Postman through a proxy
    - Sometimes easier to repeat requests that way, fuzz, etc.
    - Easier to see actuals
  - Use Postman and collections as request 'templates'

Playing RestMud with Postman

- If you access RestMud through Chrome then Postman will use the same user session details
- GET requests, could be same as the browser issues
- GET requests, use `/api/` path
- POST requests use JSON payloads
  - e.g. {"verb":"look"}

REST API



localhost:4567/player/bob/look

You Look.

Look

You are in: The Central Room

This is the room in the center

Visible Exits that you can see:

- s
- e
- w
- n

You can see:

- Some mouldy thing
  - take it
- A shiny gold ring
  - take it

Other people here, you can see:

- Alan The Mighty

Your treasure hoard is here, you have hoarded:

- Nothing

You can also:

- look
- inventory
- help

Verb look (POST)

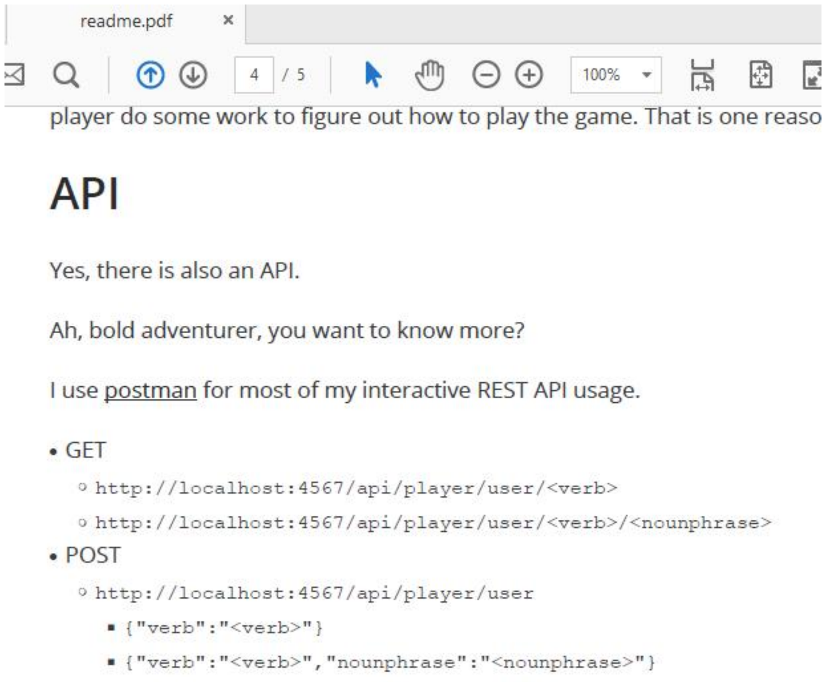
POST http://localhost:4567/api/player/bob

Authorization | Headers (1) | Body

form-data x-www-form-urlencoded [✓ raw] bin

```
{"verb":"look"}
```

RestMud API


readme.pdf

4 / 5 100%

player do some work to figure out how to play the game. That is one reaso

API

Yes, there is also an API.

Ah, bold adventurer, you want to know more?

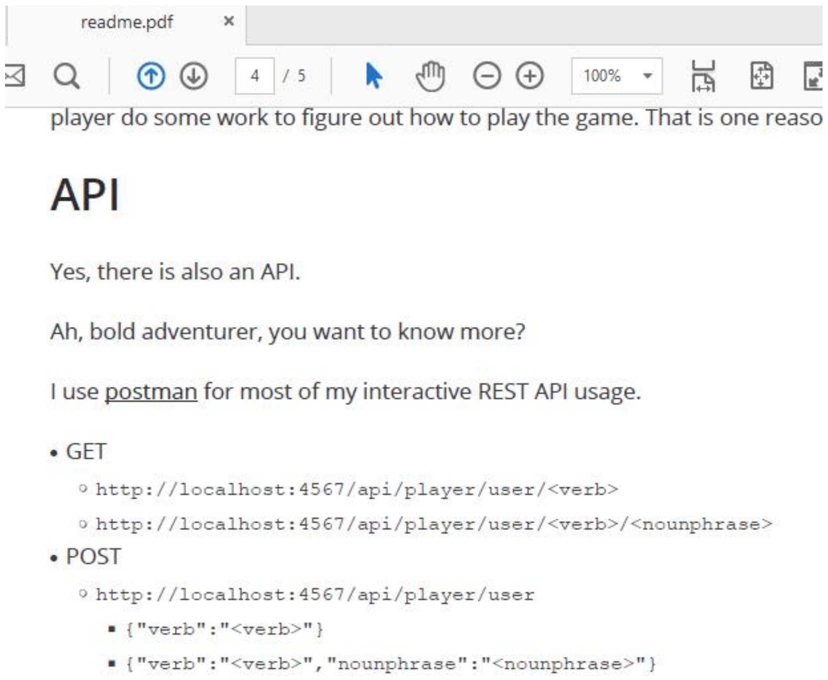
I use postman for most of my interactive REST API usage.

- GET
  - `http://localhost:4567/api/player/user/<verb>`
  - `http://localhost:4567/api/player/user/<verb>/<nounphrase>`
- POST
  - `http://localhost:4567/api/player/user`
    - `{"verb":"<verb>"}`
    - `{"verb":"<verb>","nounphrase":"<nounphrase>"}`

- GET
  - http://localhost:4567/api/player/user/<verb>
  - http://localhost:4567/api/player/user/<verb>/<noun>
- POST
  - http://localhost:4567/api/player/user
    - {"verb":"<verb>"}
    - {"verb":"<verb>","nounphrase":"<noun>"}

Using RestMud API example

- Demo playing
- Demo building a collection



player do some work to figure out how to play the game. That is one reaso

API

Yes, there is also an API.

Ah, bold adventurer, you want to know more?

I use postman for most of my interactive REST API usage.

- GET
  - `http://localhost:4567/api/player/user/<verb>`
  - `http://localhost:4567/api/player/user/<verb>/<nounphrase>`
- POST
  - `http://localhost:4567/api/player/user`
    - `{"verb":"<verb>"}`
    - `{"verb":"<verb>","nounphrase":"<nounphrase>"}`

Postman Collections

- Save As
  - To create a new collection and save as new request
- Save to overwrite a request
- Import other people's collections
  - My api.randomuser.me collection
  - <insert URL here>

Exercise: Use RestMud with Postman

- Use RestMud with Postman
- Try some get requests
- Try using Post requests
- Use the pretty print view for JSON
- Build a RestMud collection to help you play the game

Using Postman through a proxy

- Setup Chrome to go through proxy
- Use Postman

A Little More REST Theory

- HTTP Verbs are used to dictate the type of action in a CRUD model

CRUD	**HTTP VERB**	**Description**
Create	POST	Insert Data
Read	GET	Retrieve Data
Update	PUT	Amend Data
Delete	DELETE	Delete Data

A Little More REST Theory

- HTTP Return codes let you know if your call was successful or not

Code	Description
1xx	Information Message
2xx	Successful Action
3xx	Redirection Response
4xx	Client Error – error in request
5xx	Server Error – error processing request

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

A Little More REST Theory

- HTTP Return codes let you know if your call was successful or not e.g.

Code	Description
200	OK
201	Created e.g. after a POST
204	No content e.g. after a PUT or DELETE
etc.	

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

A Little More REST Theory

- HTTP Return codes let you know if your call was successful or not e.g.

Code	Description
400	Bad Request
401	Unauthorised
403	Forbidden
404	Not Found
etc.	

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

A Little More REST Theory

- HTTP Return codes let you know if your call was successful or not e.g.

Code	Description
500	Internal Server Error
501	Not Implemented
504	Gateway Timeout
etc.	

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Authorisation

- May use an `Authorization` header
- `username:password` Base64 encoded
- e.g.
  - username:password
  - dXNlcm5hbWU6cGFzc3dvcmQ=

```
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
```

https://en.wikipedia.org/wiki/Basic_access_authentication

Example Apps with APIs for practice

- Get On Tracks
- Redmine

Example: Redmine

- http://www.redmine.org/
- Vms
  - https://www.turnkeylinux.org/redmine
  - https://bitnami.com/stack/redmine

Example: GetOnTracks

- http://www.getontracks.org/
- Recommend use in a VM
  - https://bitnami.com/stack/tracks
  - https://www.turnkeylinux.org/tracks