

Week 5

COMP1531

half way there!





You're doing COMP1531 this term?
Say no more

Updates

Iteration 1

- Auto marking should be released soonTM
- Project Demo this week!!!
- Manual marking feedback will be given during the lab + on teams

Iteration 2

- Iteration 2 is out
- Implementing web server
- In typescript
- Routes are wrapped around iteration 1 functions
- Persistence
- Tokens



Topics

1

Express / HTTP

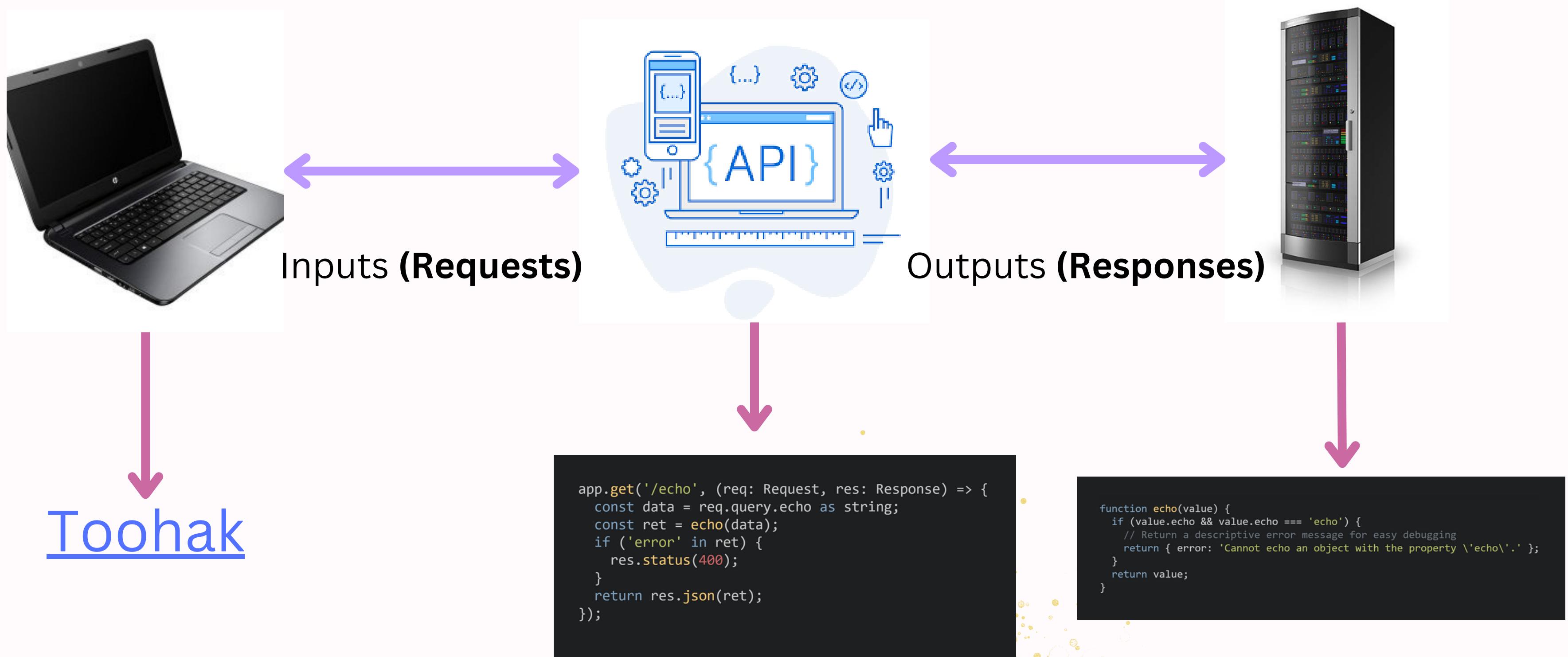
2

First-class functions (hopefully)

3

JSON and YAML (maybe)

Web Servers





APIs

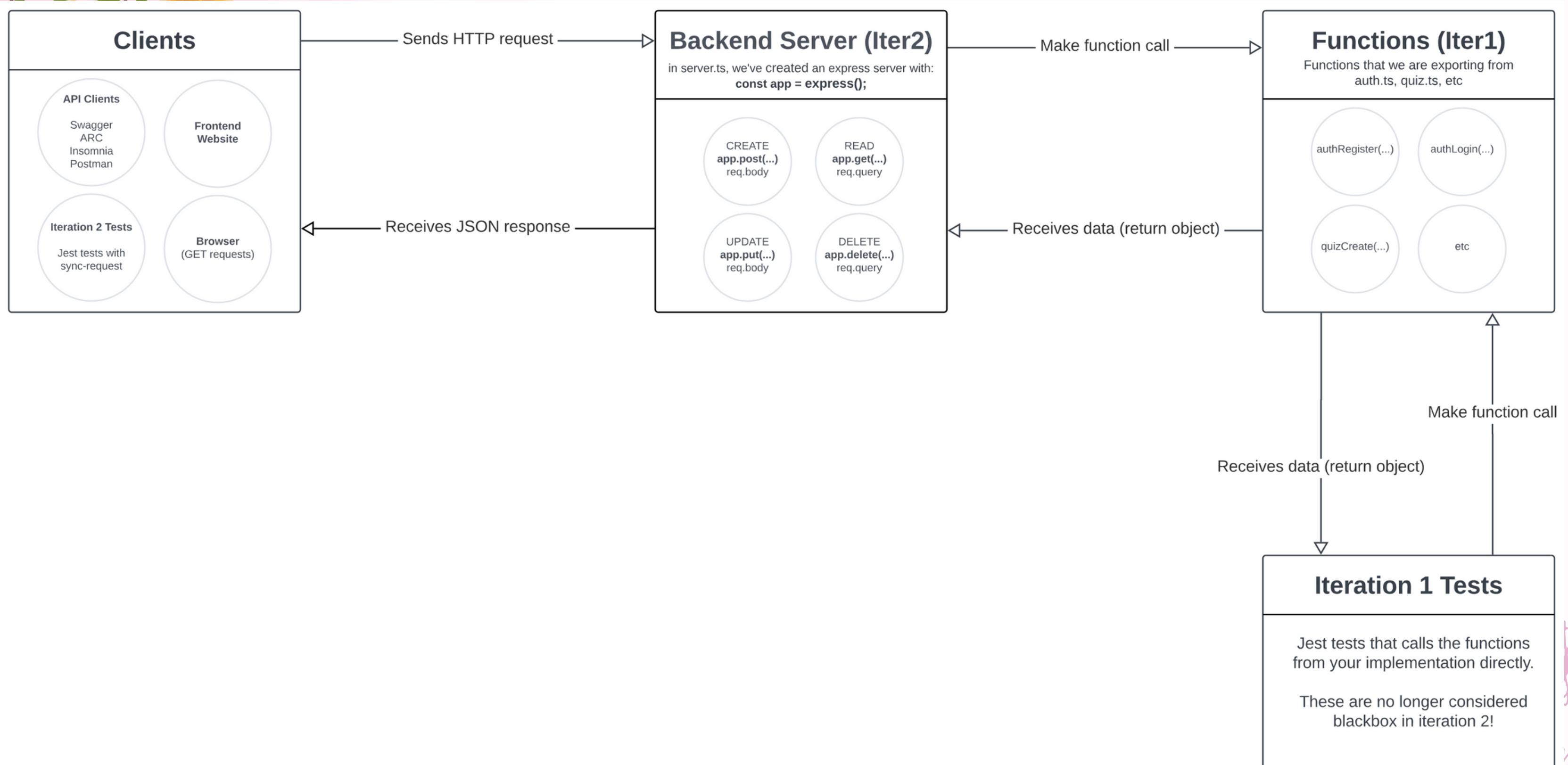
What is an API?

-  [Spotify](#)
 - and a cool [example](#) of how the Spotify API can be used in a project
-  [IMDb](#)
-  [Stocks/Finance](#)
-  [Pokemon](#)

For a more comprehensive list of free APIs for use in software and web development, see:

<https://github.com/public-apis/public-apis>

Picturing Iteration 2

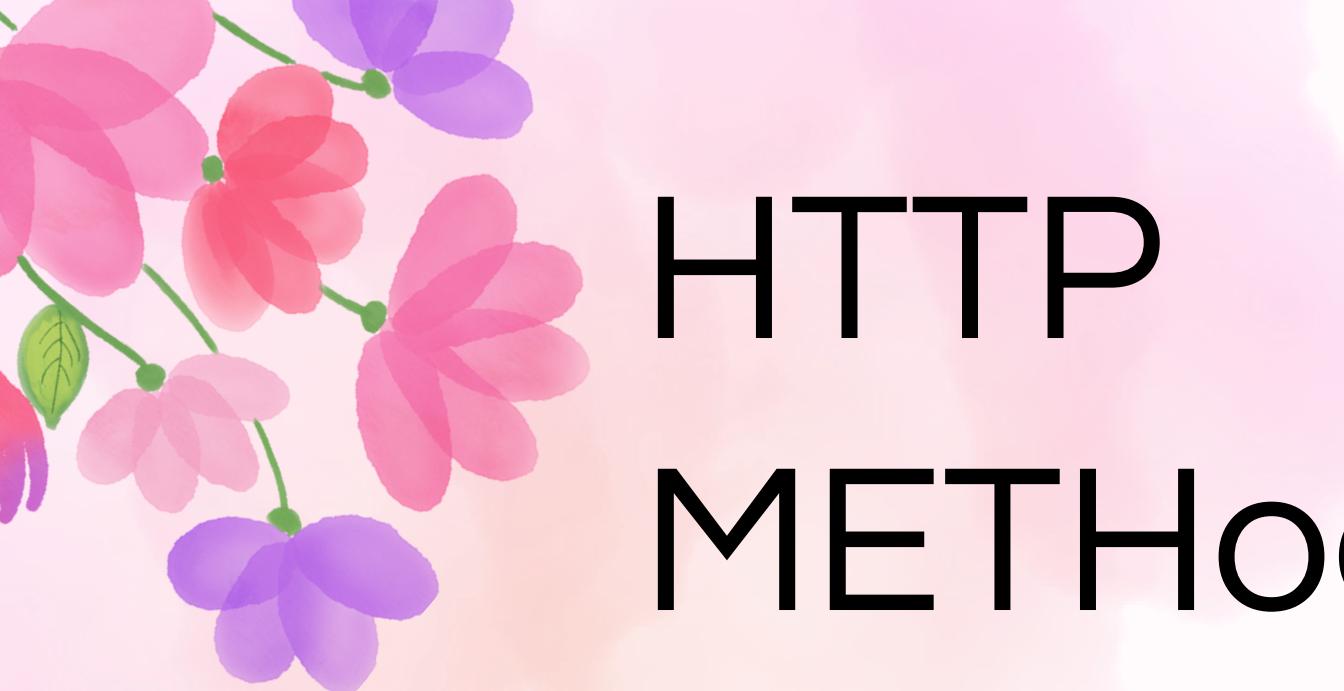


Express Server Build Demo



node.js™

express



HTTP METHODS

GET	Reading <u>data</u> from the dataStore
PUT	Editing <u>data</u> in the dataStore
POST	Adding <u>data</u> to the dataStore
DELETE	Deleting <u>data</u> from the dataStore

Interface

[Swagger.yaml](#)

Interface	Structure
Person	<pre>{ name: string, age: number }</pre>
AgeStats	<pre>{ minAge: number, maxAge: number, averageAge: number }</pre>

Name & Description	HTTP Method	Data Types	Errors
/clear Remove all entries from the data store.	???	??? Parameters {} Return Object {}	N/A
/people/add Given a name and an age, add the entry into the data store if it is valid.	???	??? Parameters {name: string, age: number} Return Object {}	Return { error } with status code 400 when: <ul style="list-style-type: none">The given name is an empty string, "".The given age is not strictly positive.The given name already exists in the data store.
/people/list Return all people that are equal to or older than the given minAge. Names should be returned in descending age order (e.g. eldest at index 0), or in lexicographical case-insensitive order if the ages are equal.	???	??? Parameters {minAge: number} Return Object { people: Person[] }	Return { error } with status code 400 when: <ul style="list-style-type: none">minAge is strictly negative.
/person/:peopleid Return the name and age of a person that corresponds with the given personid	???	??? Parameters {} Return Object { person: Person }	Return { error } with status code 400 when: <ul style="list-style-type: none">peopleid does not refer to an existing person
/person/:peopleid Edit the age for the given name entry.	???	??? Parameters {name: string, age: number} Return Object {}	Return { error } with status code 400 when: <ul style="list-style-type: none">peopleid does not refer to an existing personThe new age is not strictly positive.The new name is an empty string, ""The given name already exists in the data store (no error if updating name to the same thing!).
/person/:peopleid Remove the given person entry.	???	??? Parameters {} Return Object {}	Return { error } with status code 400 when: <ul style="list-style-type: none">The given name does not exist in the data store.
/people/stats Return an object containing stats about the age of all people	???	??? Parameters {} Return Object {stats: AgeStats}	Return { error } with status code 400 when: <ul style="list-style-type: none">There are no entries in the data store.



GET PUT POST DELETE ??

clear





GET PUT POST DELETE ??

peopleAdd





GET PUT POST DELETE ??

peopleList





GET PUT POST DELETE ??

personNameEdit





GET PUT POST DELETE ??

peopleStats





Getting data

GET	Reading <u>data</u> from the dataStore	req.query (string) req.parmas (string)
PUT	Editing <u>data</u> in the dataStore	req.body (object) req.parmas (string)
POST	Adding <u>data</u> to the dataStore	req.body (object) req.parmas (string)
DELETE	Deleting <u>data</u> from the dataStore	req.query (string) req.parmas (string)



**LAB + DEMO
TIME!**

wow!