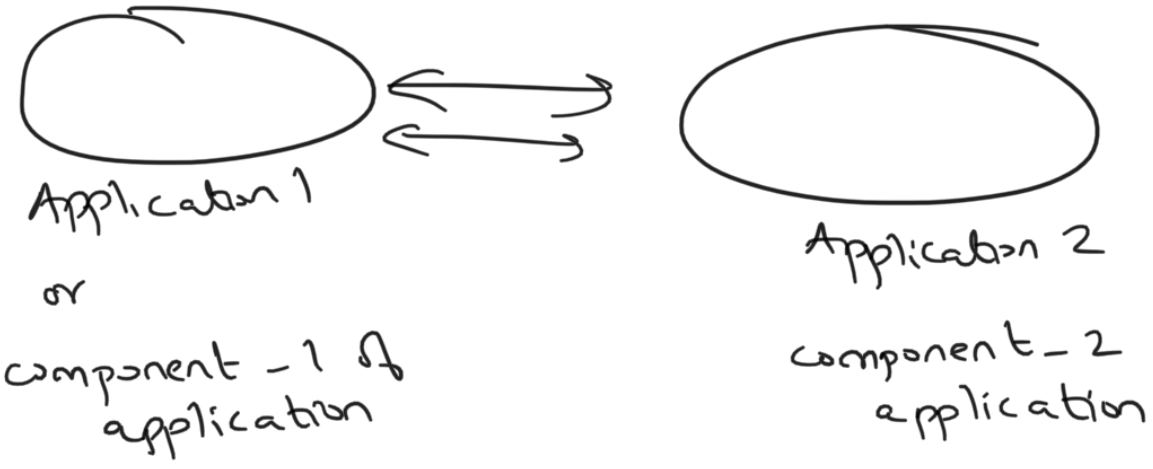
Lecture : API

Application Programming interfaces

protocols for communication bet



Info rep in

JSON : Java script object notation

(eg)
```
{ "origin" : "Tokyo"
  "destination" : "Shanghai"
  "duration" : 185
}
```

Dictionary like

↑ easy for humans and computers to read.

eg) ["Alice", "Bob"]

List like

different URLs --- to get to diff

(nested)

end points . .

/flights ← get API response ∩ all flights

/flights/28 ← get API response for flight 28 . .

APIs used to → get info

→ add new info

→ update info

HTTP methods

any web browser search is a get req by default

GET (get info)

POST (add new info)

PUT (replace info)

PATCH (update info)

DELETE (delete info)

requests (Python library)

```
import requests

def main():
    res = requests.get("_url_")
    print(res.text)

if __name__ == "__main__":
```

response object

main ()

---

example fixer.io (exuby API on the internet)

Foreign exchange rate info

→ JSON response....

dictionary?

```
data = res.json()
```

get JSON data from it

response object

```
print (data)
```

Status codes

```
if res.status_code != 200:
    raise Exception("error in API")
```

200 ← OK
201 ← Created

400 ← Bad request
403 ← Forbidden

404 ← Not Found

405 ← method not allowed

422 ← Unprocessable entity

we are able to understand req but still not able to process it---

res = requests.get("url",

params = { "parameter-1" : ... ,

"parameter-2" : ... }

the API should tell you what these keys ... parameter-1 parameter-2 are---

writing your API

```
from flask import jsonify

@app.route("/api/flights/<int:flight_id>")
def flight_api(flight_id):
```

```
return jsonify( python-dict ), status code
```

If not specified, by default it is 200

API keys

→ rate limiting (restrict API access to users---)

→ users register and get an API key

→ servers will know when getting a request, what the user's API key

→ so it can keep a tally of how many requests an API key is making