Lecture # IFN666 Web and [REDACTED]Week 03 - Server Applications 2

<Ifn666 />

Dr [REDACTED]

[REDACTED] | [REDACTED]

TEQSA Provider ID PRV12079 Australian University CRICOS No. 00213J [REDACTED]

[REDACTED]

Lecture # IFN666 Web and [REDACTED]Agenda

- Notices
- Revisiting reverse proxies
- Third-party APIs
- Advanced HTTP
- Data modelling
- Document databases
- More about RESTful APIs
- Middleware

Lecture # IFN666 Web and [REDACTED]

- Reminder: assessment 01 is on-going
- Due Friday week 4
- Assessment 02 will be released this week
- Delayed due to [REDACTED] date remains as Friday week 8
- Essay has been removed to account for loss of 1 week
- Curious
- How are you going with the web servers? Any issues?
- How is the pace of the unit?

Lecture # IFN666 Web and [REDACTED]

/><Reverse Proxies

Lecture # IFN666 Web and [REDACTED]

InternetWeb Server: ifn666.com

Rev

proxy

GET //ifn666.com/locallib/api

GET //ifn666.com/carhire/api

GET //ifn666.com/cafe/api

Local lib service

/apps/locallib/server.js

Listening on :3000

Car hire service

/apps/carhire/server.js

Listening on :3005

Cafe service

/apps/cafe/server.js

Listening on :3010

:80 HTTP:443 HTTPS

```
ifn666.com {
  handle /locallib/* {
    uri strip_prefix /locallib
    reverse_proxy localhost:3000
  }
  handle /carhire/* {
    uri strip_prefix /carhire
    reverse_proxy localhost:3005
  }
  handle /cafe/* {
    uri strip_prefix /cafe
```

```
    reverse_proxy localhost:3010
  }
}
```

Lecture # IFN666 Web and [REDACTED]

Web Server: n123.ifn666.com

186.94.58.23

Rev

proxy GET //n123.ifn666.com/locallib/api

GET //n123. ifn666.com/carhire/api

GET //n123. ifn666.com/cafe/api

Local lib service

/apps/locallib/server.js

Listening on :3000

Car hire service

/apps/carhire/server.js

Listening on :3005

Cafe service

/apps/cafe/server.js

Listening on :3010

:80 HTTP

:443 [REDACTED]

Web Server: n987.ifn666.com

186.94.58.46

Rev

proxy GET //n987.ifn666.com/locallib/api

GET //n987. ifn666.com/carhire/api

GET //n987. ifn666.com/cafe/api

Local lib service

/apps/locallib/server.js

Listening on :3000

Car hire service

/apps/carhire/server.js

Listening on :3005

Cafe service

/apps/cafe/server.js

Listening on :3010

:80 HTTP

:443 HTTPS

Domain Name [REDACTED]

Domain IP address

n123.ifn666.com 186.94.58.23

n987.ifn666.com 186.94.58.46

This architecture is useful in IFN666

because you each get only one web

server but need to run many

applications on that server.

Lecture # IFN666 Web and [REDACTED]

Web Server: locallib.n123.ifn666.com

186.94.58.23

GET //locallib.n123.ifn666.com/api

Local lib service

/apps/locallib/server.js

Listening on :80, :443

:80 HTTP

:443 HTTPSDomain Name [REDACTED]

Domain IP address

locallib.n123.ifn666.com 186.94.58.23

carhire.n123.ifn666.com 186.94.58.24

cafe.n123.ifn666.com 186.94.58.25

locallib.n987.ifn666.com 186.94.58.46

carhire.n987.ifn666.com 186.94.58.47

cafe.n987.ifn666.com 186.94.58.48

Web Server: carhire.n123.ifn666.com

186.94.58.24

GET //carhire.n123.ifn666.com/api

Local lib service

/apps/carhire/server.js

Listening on :80, :443

:80 HTTP

:443 [REDACTED]

Web Server: cafe.n123.ifn666.com

186.94.58.25

GET //cafe.n123.ifn666.com/api

Local lib service

/apps/cafe/server.js

Listening on :80, :443

:80 HTTP

:443 [REDACTED]

Web Server: locallib.n987.ifn666.com

186.94.58.46

GET // locallib.n987.ifn666.com/api

Local lib service

/apps/locallib/server.js

Listening on :80, :443

:80 HTTP

:443 [REDACTED]

...

This architecture does not use a

reverse proxy. Instead, each

application is running on a server of

its own. [REDACTED] does not use this

architecture because you each get

only one web server.

Lecture # IFN666 Web and [REDACTED] provides [REDACTED]

- Caddy's default [REDACTED] settings are secure
- You could manually specify the settings, but avoid that
- Local/internal hostnames are secured with a self-signed cert
- Public DNS names are secured from a public ACME CA
- Caddy uses Let's Encrypt and ZeroSSL to obtain certificates
- Both are trusted, large organisations that issue free certificates
- For the CA to issue a certificate, [REDACTED] publishes a temporary

cryptographic resource on port 80. The CA sees the expected

resource and issues the certificate.

ACME CA is [REDACTED]

Lecture # IFN666 Web and [REDACTED]

Lecture # IFN666 Web and [REDACTED]

[REDACTED] without Caddy

- You would need to manually configure a certificate

- Either request a free cert from Let's Encrypt or alike

- Or, purchase one from another CA

Lecture # IFN666 Web and [REDACTED]

/><Third-party APIs

CAB432

[REDACTED]Lecture # IFN666 Web and [REDACTED] APIs in IFN666

- We are encouraging you to use third-party, public APIs

- Learn through exposure to real APIs

- The quality of APIs widely varies

- Some APIs are not free to use

- Be mindful of limits

- Your budget

- Rate limiting (x requests per minute, for example)

- Use reputable vendors

- If an [REDACTED] becomes unavailable before key milestones in

assessment, we cannot help you

Lecture # IFN666 Web and [REDACTED]

Why use third-party APIs?

- To avoid reinventing the wheel

- Why forecast the weather when others are already doing it?

- To combine your proprietary data with external data for novel use

- Out of interest, for commercial purposes, etc.

- To enable rich user experience

- Login to my app with [REDACTED], [REDACTED], [REDACTED], etc.

- Share in-game what you are listening to on [REDACTED] weather predictions or historical data

[REDACTED]

Some key examples

[REDACTED]

- Spotify

- Weather

- Google (has hundreds)

- OpenAI

- QUT Canvas [REDACTED] use the [REDACTED] within the T&Cs

- Never share your API credentials

Lecture # IFN666 Web and [REDACTED]

- As with any software, documentation is important for you and other

developers

- Some APIs are not documented because they are not intended to

be used by applications other than the vendors'

- For example, the [REDACTED] mobile applications use an undocumented API

- I doubt you would find documentation for the [REDACTED] powering the [REDACTED] mobile

application

Lecture # IFN666 Web and [REDACTED]

/><Advanced HTTP

Lecture # IFN666 Web and [REDACTED]

Media types / MIME types

- A [REDACTED]

type describes the format of the document

- Generally, the file extension should not be

authoritative

- Browsers determine file content based on

the MIME type specified in the response

- Some server configurations may use the

associated MIME type to perform

optimizations, such as file concatenation,

compression, or caching.

- Key examples

- text/plain

- text/css

- text/html

- image/jpeg

- multipart/form-data

Some material on this slide was created by [REDACTED] and is licensed under [REDACTED]-BY-SA 2.5: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types

Lecture # IFN666 Web and [REDACTED]

MIME types for REST APIs

- application/json - Most common for REST APIs

- application/xml - Used in older APIs or when XML is required.

- Less so or not accepted by some REST APIs:

- text/plain - For simple text responses.

- multipart/form-data - Used for file uploads.

- application/x-www-form-urlencoded - Used in form

submissions (e.g., [REDACTED] with URL-encoded data).

Lecture # IFN666 Web and [REDACTED]

Cookies

- Small piece of data a server sends to a user's web browser

- The browser may create, modify, delete cookies

- The browser can send cookies to the server

- Cookies can expire

- Useful for session management, personalisation, tracking

- Servers can specify if cookies can be sent to other servers

- More modern APIs are available: Storage API, IndexedDB

- Caveat: "cookies" are not available on other platforms

Lecture # IFN666 Web and [REDACTED] for user authentication

- Next week, we will discuss more detail about how to securely implement user authentication

- The process involves HTTP cookies and headers

1. The client sends a request to login using the provided user/pwd

2. The server checks the credentials are correct and sends back a cookie that represents that user

3. In subsequent requests, the client sends that cookie to the server to prove who they are

Lecture # IFN666 Web and [REDACTED]

1) Client requests to login

```
POST /login HTTP/1.1
Host: localhost:3000
Content-Type: application/json
Content-Length: 33
Connection: keep-alive
Accept: */*
User-Agent: curl/7.64.1
Accept-Encoding: gzip, deflate
{
 "username": "john_doe",
 "password": "secret_password"
```