Urban Big Data Centre

Data and expertise for improving cities

www.ubdc.ac.uk

Practical approaches for web scraping for research – using Airbnb as an example data provider

Transcript from webinar video recording

1

00:00:00,018 --> 00:00:01,526

First of all, welcome everybody

2

00:00:01,626 --> 00:00:05,074

to the latest in a series of webinars

3

00:00:05,174 --> 00:00:09,241

being presented by the Urban Big Data Centre.

4

00:00:09,739 --> 00:00:14,606

This session, hopefully you're not surprised to see, is entitled

5

00:00:14,706 --> 00:00:16,885

Practical Approaches for Web Scraping

6

00:00:17,303 --> 00:00:21,962

using Airbnb data and the UBDC Airbnb scraping platform.

7

00:00:22,582 --> 00:00:24,209

My name is Andrew McHugh.

8

00:00:24,309 --> 00:00:28,132

I'm the Senior Data Science Manager at the Urban Big Data Centre

9

00:00:28,232 --> 00:00:31,710

and I will describe what we do as a centre

10

00:00:32,748 --> 00:00:34,157

in the following slides.

11

00:00:34,257 --> 00:00:36,572

But I'll ask my colleague, Nick Ves,

12

00:00:36,941 --> 00:00:40,756

who is the architect of the platform we'll showcase today,

13

00:00:41,695 --> 00:00:43,355

to introduce himself just now as well.

14

00:00:43,505 --> 00:00:46,123

- Nick.

- Hi, good morning all.

15

00:00:46,223 --> 00:00:52,307

I am a Data Scientist working with the UBDC and Andrew here.

16

00:00:52,844 --> 00:00:57,732

And it's a pleasure that you guys have come here.

17

00:00:57,832 --> 00:01:00,330

We are going to present to you, later on,

18

00:01:02,164 --> 00:01:06,796

the way that we have been using to capture data from Airbnb.

19

00:01:06,993 --> 00:01:10,550

So, hopefully you will all enjoy it and welcome.

20

00:01:12,293 --> 00:01:14,121

Thanks, Nick. So, we've got

21

00:01:14,221 --> 00:01:16,588

a relatively packed agenda today,

22

00:01:16,688 --> 00:01:18,998

but I want to start with some housekeeping.

23

00:01:20,676 --> 00:01:22,920

This session is being recorded,

24

00:01:23,020 --> 00:01:27,243

as you will no doubt have been notified when you joined the Zoom session.

25

00:01:27,343 --> 00:01:29,490

So, an edited video will be published on

26

00:01:29,590 --> 00:01:31,776

UBDC's YouTube Channel in due course.

27

00:01:31,876 --> 00:01:34,169

That's the YouTube link where you'll find

28

00:01:34,269 --> 00:01:37,056

all Urban Big Data Centre video content.

29

00:01:37,417 --> 00:01:40,884

But just, obviously, to bear in mind that that is happening.

30

00:01:41,318 --> 00:01:44,206

All slides that we will present today will be

31

00:01:44,306 --> 00:01:45,973

made available to course participants,

32

00:01:46,073 --> 00:01:52,493

as well as the code that will form the major part of today's focus.

33

00:01:53,450 --> 00:01:55,067

Also all accessible.

34

00:01:56,336 --> 00:01:59,402

Questions can and we encourage you to

35

00:02:01,586 --> 00:02:03,782

send those in via the Zoom chat.

36

00:02:03,830 --> 00:02:05,464

We will have time at the end

37

00:02:05,564 --> 00:02:08,451

and we'll also take some time in the middle of the session

38

00:02:08,551 --> 00:02:09,998

to cover those.

39

00:02:10,098 --> 00:02:11,785

But they will be monitored throughout,

40

00:02:11,885 --> 00:02:15,302

so if you have some more technical questions

41

00:02:15,402 --> 00:02:21,388

or bigger questions, we'll tackle those at the appropriate times.

42

00:02:21,738 --> 00:02:25,449

And we ask, obviously, that people keep mics muted throughout the session.

43

00:02:26,553 --> 00:02:31,328

And then the principal form of interactivity will be

44

00:02:31,428 --> 00:02:33,376

through the Zoom chat.

45

00:02:34,757 --> 00:02:37,137

So, just to summarise what we'll be covering today.

46

00:02:37,543 --> 00:02:40,734

You'll have seen the intended learning outcomes already, perhaps.

47

00:02:40,834 --> 00:02:43,882

But following today's webinar,

48

00:02:44,216 --> 00:02:46,622

we hope you should be able to accomplish a number of things.

49

00:02:46,722 --> 00:02:49,211

And the main focus, the principle focus today is

50

00:02:49,500 --> 00:02:54,049

UBDC's open-source platform for scraping Airbnb data.

51

00:02:54,149 --> 00:02:55,297

So, you'll be able to

52

00:02:56,135 --> 00:02:58,790

describe the installation, configuration, and deployment of that.

53

00:02:58,890 --> 00:03:01,978

And later on, having grabbed the code for yourself,

54

00:03:02,078 --> 00:03:04,587

you'll be able to recreate that within your own environment.

55

00:03:04,653 --> 00:03:07,906

You'll be able to explain what web scraping means in very broad terms.

56

00:03:08,515 --> 00:03:12,073

Understand what the key legal, ethical, and data governance issues

57

00:03:12,173 --> 00:03:14,362

associated with web scraping are.

58

00:03:15,497 --> 00:03:17,977

Summarise the limits on data use and sharing.

59

00:03:19,490 --> 00:03:23,173

To identify and select appropriate datasets for web scraping.

60

00:03:23,515 --> 00:03:25,503

Understand what an API is.

61

00:03:26,543 --> 00:03:30,077

Learn how to negotiate some of the technical barriers

62

00:03:30,961 --> 00:03:33,180

that we associate with web scraping.

63

00:03:33,689 --> 00:03:37,208

And lastly, to systematise scale and optimise your approaches.

64

00:03:37,927 --> 00:03:40,285

So, as I say, this will be principally through the lens of

65

00:03:40,385 --> 00:03:43,771

our own implementation, our own scraping platform.

66

00:03:45,758 --> 00:03:49,137

But hopefully the lessons

67

00:03:49,237 --> 00:03:51,796

and the learnings

that we describe will be

68

00:03:51,896 --> 00:03:54,935

applicable to other kinds of problems that you may be tackling.

69

00:03:55,774 --> 00:03:58,083

In terms of the programme, we're here for two hours today.

70

00:03:59,642 --> 00:04:05,173

We're in the first of those agenda items at the moment.

71

00:04:06,389 --> 00:04:08,966

We will have a talk that follows.

72

00:04:09,066 --> 00:04:13,179

I will lead you through some of the background to this work

73

00:04:13,351 --> 00:04:17,790

before handing over to Nick for the main part of the agenda,

74

00:04:17,890 --> 00:04:20,725

which will be a technical discussion of

75

00:04:21,203 --> 00:04:22,852
how we've approached the scraping problem

76
00:04:22,952 --> 00:04:26,040
and how we've implemented a platform.

77
00:04:26,238 --> 00:04:29,477
And latterly, of course, how that platform works

78
00:04:29,577 --> 00:04:31,075
and how you can use it yourself.

79
00:04:31,754 --> 00:04:34,253
And, as I said, again,

80
00:04:34,570 --> 00:04:39,909
encouraging as much discussion and questions as possible

81
00:04:40,009 --> 00:04:44,878
to try and ensure that this answers any of those questions that may arise

82
00:04:44,978 --> 00:04:46,875
or are already on your mind.

83

00:04:47,615 --> 00:04:49,893

Before we get started, I wanted to draw your attention to

84

00:04:49,993 --> 00:04:52,300

just one other webinar. I mentioned that this is

85

00:04:52,400 --> 00:04:55,348

the latest in a series of UBDC webinars.

86

00:04:55,448 --> 00:04:58,345

We have another one, later this week, on Thursday,

87

00:04:58,445 --> 00:05:00,770

the 24th June at 10am,

88

00:05:01,737 --> 00:05:05,247

which is something of a companion piece to this one.

89

00:05:06,524 --> 00:05:08,841

Whereas we focus today on how you go about

90

00:05:09,846 --> 00:05:13,394

accumulating collections of

data through web scraping,

91

00:05:13,822 --> 00:05:15,801

we'll have a companion session on Thursday

92

00:05:16,299 --> 00:05:19,556

led by my colleagues Mark Livingston and Yang Wang

93

00:05:19,923 --> 00:05:22,931

and they will be focusing much more on

94

00:05:23,031 --> 00:05:27,220

how you actually use the Airbnb web scraped data

95

00:05:27,498 --> 00:05:30,896

to undertake some pertinent research questions

96

00:05:30,996 --> 00:05:32,335

and research challenges.

97

00:05:32,435 --> 00:05:36,034

So, that takes place on Thursday.

98

00:05:36,134 --> 00:05:38,753

You can visit the UBDC website for more details and register.

99

00:05:38,853 --> 00:05:43,362

And, again, like all of these webinars, that's free to participate in.

100

00:05:45,100 --> 00:05:48,246

So, moving on, I'll offer just a wee summary of

101

00:05:48,346 --> 00:05:49,924

what the Urban Big Data Centre does

102

00:05:50,024 --> 00:05:52,072

because, again, it may give you some context

103

00:05:52,172 --> 00:05:56,510

and it may make some of what follows a bit clearer.

104

00:05:57,047 --> 00:05:59,568

We're based at the University of Glasgow

105

00:06:00,649 --> 00:06:01,959

in the United Kingdom.

106

00:06:02,059 --> 00:06:05,447

We are funded by the Economic and Social Research Council,

107

00:06:05,547 --> 00:06:07,335

which is a UK government funder of

108

00:06:07,831 --> 00:06:11,561

social science research and associated infrastructure.

109

00:06:11,726 --> 00:06:15,798

As well as receiving some funding from the University of Glasgow itself.

110

00:06:16,567 --> 00:06:17,884

We've got several priorities,

111

00:06:20,700 --> 00:06:24,709

which are associated with two principal functions.

112

00:06:25,387 --> 00:06:29,364

So, we operate a traditional research centre

113

00:06:29,464 --> 00:06:31,973

focused on better understanding cities

114

00:06:32,073 --> 00:06:35,603

and relying on the use of new and emerging forms of data

115

00:06:35,703 --> 00:06:37,210

and associated methods

116

00:06:37,709 --> 00:06:39,625

to enhance our understanding of

117

00:06:39,725 --> 00:06:43,293

how cities function and how their residents behave and interact.

118

00:06:43,573 --> 00:06:45,353

And we also offer, on behalf of

119

00:06:45,453 --> 00:06:47,329

the Economic and Social Research Council,

120

00:06:47,429 --> 00:06:52,458

a national data service where we support other, mainly academic, researchers

121

00:06:52,923 --> 00:06:55,662

in exploring similar themes in similar kinds of ways.

122

00:06:55,762 --> 00:06:58,471

But amid that, we have these priorities around

123

00:06:58,571 --> 00:07:00,109

data infrastructure and collections.

124

00:07:00,209 --> 00:07:03,307

So, we're actively developing our data infrastructure

125

00:07:03,407 --> 00:07:05,834

and associated data collections.

126

00:07:06,916 --> 00:07:10,898

We support a number of priority research strands

127

00:07:10,998 --> 00:07:13,206

within the Centre, although not exclusively

128

00:07:13,979 --> 00:07:16,127

for those that we support externally.

129

00:07:16,227 --> 00:07:18,313
But those include things like transport and mobility;

130
00:07:18,413 --> 00:07:20,068
neighbourhood, housing, and environment;

131
00:07:20,269 --> 00:07:22,424
education, skills, and productivity;

132
00:07:22,524 --> 00:07:27,732
and some of the more umbrella questions, if you like,

133
00:07:28,000 --> 00:07:31,868
around the role and value of big data and urban governance.

134
00:07:33,146 --> 00:07:35,866
And our aims are to achieve public policy impact,

135
00:07:36,354 --> 00:07:38,280
to critically evaluate the role and the value of

136
00:07:38,380 --> 00:07:39,737
big data and urban analytics,

137

00:07:39,837 --> 00:07:44,225

and to enhance the data collections that we create and collect

138

00:07:44,393 --> 00:07:47,023

and the associated methods for their use.

139

00:07:48,005 --> 00:07:52,002

And all of these various different functions and priorities

140

00:07:53,190 --> 00:07:58,130

often come back to or rely upon the availability of data.

141

00:07:58,230 --> 00:08:00,909

And getting hold of data is, as you can imagine

142

00:08:01,009 --> 00:08:03,587

and no doubt many of you have experienced,

143

00:08:04,054 --> 00:08:05,935

can be quite a challenge at times.

144

00:08:06,674 --> 00:08:08,842

It takes up a lot of our time

145

00:08:09,580 --> 00:08:12,868

to go out and not only identify data

146

00:08:12,968 --> 00:08:16,906

but to develop the means by which we can get hold of that data.

147

00:08:17,006 --> 00:08:19,385

And that can imply a range of technical challenges

148

00:08:19,485 --> 00:08:22,124

as well as, obviously, legal challenges and so on.

149

00:08:22,735 --> 00:08:25,447

And obviously there are many reasons why organisations don't

150

00:08:26,003 --> 00:08:27,182

make their data available

151

00:08:27,282 --> 00:08:31,881

or do not feel incentivised to do so.

152

00:08:32,290 --> 00:08:34,306

Including a lack of capacity to do so;

153

00:08:34,406 --> 00:08:35,744

perceptions of risk;

154

00:08:35,893 --> 00:08:38,343

perceived conflict with existing business models

155

00:08:38,971 --> 00:08:42,329

if, for example, making data available may undermine

156

00:08:42,557 --> 00:08:47,575

the business models which involve the provision of data

157

00:08:47,883 --> 00:08:49,341

on a commercial basis

158

00:08:49,839 --> 00:08:51,476

or other kinds of conflicts;

159

00:08:52,399 --> 00:08:55,207

and fears too that research results may reflect badly on them

160

00:08:55,307 --> 00:08:58,655

or expose them to criticism or be received negatively

161

00:08:58,755 --> 00:09:00,114

by other stakeholders.

162

00:09:00,601 --> 00:09:02,125

That last point is probably one that

163

00:09:02,225 --> 00:09:05,544

we focus on most in the content that follows

164

00:09:05,644 --> 00:09:07,671

and in our particular case.

165

00:09:08,132 --> 00:09:11,625

And while there are many reasons why we feel we can encourage organisations

166

00:09:11,725 --> 00:09:13,850

to make their data available, they don't always work.

167

00:09:13,950 --> 00:09:17,463

So, sometimes we have to find other means of getting hold of data

168

00:09:18,640 --> 00:09:22,397

that aren't quite as collaborative as,

169

00:09:23,375 --> 00:09:27,801

for example, a licensing agreement or a collaboration agreement.

170

00:09:30,472 --> 00:09:32,843

So, Airbnb is obviously the focus of today's session.

171

00:09:32,943 --> 00:09:37,017

It provides the lens through which we'll look at the technology discussion

172

00:09:37,117 --> 00:09:38,406

that will follow later on.

173

00:09:38,685 --> 00:09:41,950

So, I thought it would be useful, again, to give you a little bit of a summary of

174

00:09:42,246 --> 00:09:44,814

what we were looking for and for what reason.

175

00:09:45,796 --> 00:09:48,164

I'll start with a very, very fundamental point.

176

00:09:48,264 --> 00:09:51,422

So, many of you will no doubt be familiar with Airbnb

177

00:09:51,522 --> 00:09:56,037

and no doubt have used their services in the past yourself.

178

00:09:56,137 --> 00:09:59,715

But it's an online market place for arranging or offering lodgings,

179

00:09:59,815 --> 00:10:02,252

homestays, and tourism experiences.

180

00:10:02,352 --> 00:10:07,937

So, you can onto Airbnb and perform one of several roles.

181

00:10:08,037 --> 00:10:09,304

The most obvious ones being

182

00:10:09,492 --> 00:10:11,961

there are hosts, there are people that own properties

183

00:10:12,061 --> 00:10:15,777

or that manage properties on behalf of others who use Airbnb as a platform

184

00:10:15,877 --> 00:10:22,215

to promote the availability of these homes or lodgings

185

00:10:23,442 --> 00:10:28,579

for tourism or for other kinds of accommodation purposes.

186

00:10:29,120 --> 00:10:32,257

Meanwhile, there is another set of people

187

00:10:32,357 --> 00:10:34,698

who are on Airbnb from a consumer point of view.

188

00:10:34,798 --> 00:10:36,885

People who are interested in finding a holiday home,

189

00:10:36,985 --> 00:10:39,128

finding somewhere to stay for a short-term basis.

190

00:10:40,239 --> 00:10:47,686

And they will discover the available Airbnb homes

191

00:10:47,786 --> 00:10:50,993

or venues or accommodations via that platform.

192

00:10:52,630 --> 00:10:55,930

Our researchers were looking at Airbnb

193

00:10:56,030 --> 00:11:02,575

and were particularly interested in some of the impacts of Airbnb

194

00:11:02,675 --> 00:11:05,400

and the rapidly growing sharing economy

195

00:11:05,500 --> 00:11:07,898

that it's a really important manifestation of.

196

00:11:09,358 --> 00:11:10,584

What are the impacts of that?

197

00:11:10,684 --> 00:11:13,989

What's the importance of that for the private rented sector property market?

198

00:11:14,201 --> 00:11:19,114

How does this impact on more traditional private rental?

199

00:11:20,883 --> 00:11:23,894

We hear of these things, obviously as being very disruptive

200

00:11:23,994 --> 00:11:26,726

and we wanted to be able to characterise and understand

201

00:11:27,267 --> 00:11:29,877

what those disruptive impacts actually are.

202

00:11:30,795 --> 00:11:31,968

Question two,

203

00:11:33,027 --> 00:11:37,145

the extent to which they are positive or potentially damaging.

204

00:11:38,075 --> 00:11:43,039

Some of the questions that are foremost in our researchers' minds.

205

00:11:44,518 --> 00:11:45,787

The impact that Airbnb has on

206

00:11:45,887 --> 00:11:47,856

the availability of private rented stock.

207

00:11:47,956 --> 00:11:55,094

So, given the opportunity to market and sell properties

208

00:11:55,194 --> 00:11:57,470

or rather sell access to properties on Airbnb,

209

00:11:57,570 --> 00:12:01,278

does that limit the amount of private rental stock that's available

210

00:12:01,378 --> 00:12:02,915

on a more traditional basis?

211

00:12:03,624 --> 00:12:05,682

Is there a spatial aspect?

212

00:12:05,782 --> 00:12:08,061

Are there particular locations or areas that

213

00:12:09,117 --> 00:12:14,667

see a greater or lesser impact of these changes?

214

00:12:15,655 --> 00:12:18,339

What, if any, is the relationship with areas of social deprivation?

215

00:12:18,439 --> 00:12:20,294

And does the rise of the sharing economy

216

00:12:20,394 --> 00:12:23,672

increase things like inner city gentrification

217

00:12:23,772 --> 00:12:26,900

and the suburbanisation of poverty?

218

00:12:27,000 --> 00:12:31,085

So, things we would often associate with negative outcomes.

219

00:12:31,695 --> 00:12:33,839

Are Airbnb properties falling below the standards for

220

00:12:33,939 --> 00:12:35,956

the private rented sector?

221

00:12:36,056 --> 00:12:37,525

E.g. in terms of occupancy levels.

222

00:12:37,625 --> 00:12:40,929

So, where there is a regulated environment,

223

00:12:41,389 --> 00:12:44,696

are there standards that are not being met?

224

00:12:44,796 --> 00:12:49,244

Has this been providing a loophole for substandard service provision?

225

00:12:50,704 --> 00:12:51,704

And what impact, if any,

226

00:12:51,804 --> 00:12:54,174

does Airbnb have on the existing hospitality industry,

227

00:12:54,274 --> 00:12:56,093

as well as private rented sectors?

228

00:12:56,193 --> 00:12:58,180

So, as you can no doubt imagine...

229

00:12:58,280 --> 00:13:00,659

And I won't go into these in any more detail than that.

230

00:13:00,759 --> 00:13:03,097

you'll see perhaps some more coverage of some of these

231

00:13:03,197 --> 00:13:06,125

specific research questions and how they are tackled and resolved

232

00:13:06,225 --> 00:13:09,343

in the later webinar on Thursday.

233

00:13:10,233 --> 00:13:13,581

But these are really just presented to show that these are not necessarily

234

00:13:13,681 --> 00:13:18,990

research interests or goals that will resonate with Airbnb

235

00:13:19,090 --> 00:13:21,958

as being something they are keen to get involved in.

236

00:13:22,599 --> 00:13:24,267

They may not be enthusiastic about

237

00:13:24,367 --> 00:13:27,340

facilitating such research, which may have the potential to

238

00:13:28,608 --> 00:13:32,925

lead to outcomes which may be critical of their business models

239

00:13:33,025 --> 00:13:34,102

and so on.

240

00:13:35,572 --> 00:13:42,200

But given their possible lack of enthusiasm about making data available,

241

00:13:42,300 --> 00:13:44,778

we have to look to sources elsewhere.

242

00:13:44,878 --> 00:13:46,906

Unfortunately, as an unregulated sector,

243

00:13:47,006 --> 00:13:49,444

there is little reliable data available

244

00:13:49,544 --> 00:13:52,220

so we have to think creatively about how we can get the data

245

00:13:52,320 --> 00:13:56,315

that will give us the insights into how this model works

246

00:13:56,415 --> 00:13:58,033

and what its implications are.

247

00:13:59,424 --> 00:14:02,341

So, we wanted to automate the scraping of Airbnb.

248

00:14:03,061 --> 00:14:07,127

As an online marketplace, the data is available on the web,

249

00:14:07,336 --> 00:14:09,795

it's accessible freely.

250

00:14:10,882 --> 00:14:15,181

And we wanted to effectively industrialise the collection of data

251

00:14:15,281 --> 00:14:18,000

from that platform on a very systematic basis

252

00:14:18,917 --> 00:14:21,218

in order to facilitate our research.

253

00:14:23,618 --> 00:14:25,686

An obvious initial starting point was

254

00:14:25,786 --> 00:14:28,044

the legality of our proposed data collection.

255

00:14:28,144 --> 00:14:33,051

To what extent are we legally entitled or permitted to

256

00:14:33,151 --> 00:14:36,369

go and collect data at scale from Airbnb

257

00:14:36,649 --> 00:14:38,107

in that kind of way?

258

00:14:39,356 --> 00:14:42,386

I'll cover that in some slides in a moment or so.

259

00:14:43,054 --> 00:14:46,052

But then, if we're able to reassure ourselves that we are

260

00:14:47,400 --> 00:14:49,707

legally permitted, we have a sound legal basis

261

00:14:49,901 --> 00:14:51,882

for proceeding in that particular way,

262

00:14:54,507 --> 00:14:59,436

can we then develop methods and can we design our data collection

263

00:14:59,632 --> 00:15:04,085

to support questions around the scale and growth of Airbnb?

264

00:15:05,190 --> 00:15:06,469

Spatial change?

265

00:15:06,617 --> 00:15:08,431

Focus on short-term lets?

266

00:15:08,531 --> 00:15:12,899

So, can we see the changes manifest on a spatial basis?

267

00:15:13,082 --> 00:15:14,790

Can we understand how occupancy works?

268

00:15:14,890 --> 00:15:21,339

How many people are occupying particular Airbnbs at a given time?

269

00:15:21,439 --> 00:15:23,726

Can we understand price and the changes in price?

270

00:15:23,826 --> 00:15:27,475

And what some of the determinants of price

271

00:15:27,575 --> 00:15:29,194

and the things that influence price are?

272

00:15:29,348 --> 00:15:30,659

And availability as well.

273

00:15:30,759 --> 00:15:34,808

So, these are the things that we may feel are

274

00:15:34,908 --> 00:15:36,906

self-evident within the data, to some extent,

275

00:15:37,006 --> 00:15:38,654

and when we go onto Airbnb,

276

00:15:38,754 --> 00:15:43,782

we feel we can sort of infer some of these things on a limited basis.

277

00:15:45,087 --> 00:15:47,005

But can we capture these at scale

278

00:15:47,105 --> 00:15:53,062

and achieve a robust and a credible understanding of some of these things?

279

00:15:54,603 --> 00:15:58,231

So, that will frame how we actually go about

280

00:15:58,331 --> 00:16:00,559

doing the scraping, which we'll see later on.

281

00:16:00,659 --> 00:16:02,812

But, again, before going there,

282

00:16:02,912 --> 00:16:08,220

I think it's important just to cover some of the fundamentals.

283

00:16:08,766 --> 00:16:12,384

And the first is just to offer a definition of scraping itself.

284

00:16:12,484 --> 00:16:17,217

So, I've already mentioned in the past that we'd go out and get data from

285

00:16:17,317 --> 00:16:19,643

a variety of organisations in support of our research

286

00:16:19,743 --> 00:16:22,461

and very often that can be a very collaborative thing.

287

00:16:22,561 --> 00:16:25,688

It can be a very explicitly consensual thing.

288

00:16:25,788 --> 00:16:27,768

Things like license agreements and so on.

289

00:16:28,394 --> 00:16:32,964

Scraping is, comparatively, a one-sided activity.

290

00:16:33,153 --> 00:16:36,972

It's about taking data that exists, for example, on a website,

291

00:16:37,965 --> 00:16:40,311

and employing any one or a combination of

292

00:16:40,660 --> 00:16:42,599

a range of different technical means

293

00:16:43,038 --> 00:16:47,315

to automate the transfer of data from that location or resource.

294

00:16:47,415 --> 00:16:51,007

And there's a whole range of methods that can be done to facilitate that.

295

00:16:51,107 --> 00:16:56,285

So, with online data, we can employ techniques like text pattern matching

296

00:16:56,385 --> 00:17:00,473

to grab particular types of data

297

00:17:00,573 --> 00:17:02,582

and to then impose some kind of

298

00:17:02,861 --> 00:17:05,634

structure on those at the collection end.

299

00:17:06,094 --> 00:17:08,089

We can use things like HTTP programming.

300

00:17:08,189 --> 00:17:10,818

And that will be the primary focus of what we do,

301

00:17:11,215 --> 00:17:15,381

relying on the Airbnb API, as Nick will outline later on.

302

00:17:16,250 --> 00:17:21,177

We can rely on the existing structure that is implicit within web content

303

00:17:21,277 --> 00:17:24,257

through HTML and the Document Object Models

304

00:17:25,755 --> 00:17:30,622

and effectively parse through marked up content to extract

305

00:17:30,722 --> 00:17:33,168

particular data,

306

00:17:33,268 --> 00:17:35,745

and then obviously impose structure

307

00:17:35,845 --> 00:17:40,217

or transpose from the structure already evident.

308

00:17:40,996 --> 00:17:44,597

And we can use a whole range of other more sophisticated techniques too,

309

00:17:44,697 --> 00:17:46,167

like computer vision, for instance,

310

00:17:46,267 --> 00:17:50,326

where we're actually analysing the visual representation of

311

00:17:50,726 --> 00:17:53,504

web content in order to extract particular parts.

312

00:17:54,464 --> 00:17:57,813

What's, I think, true of all of these approaches

313

00:17:57,913 --> 00:18:02,061

and is core to the technical philosophy of this approach is that

314

00:18:02,161 --> 00:18:04,478

it's unsupported by a web content owner.

315

00:18:04,568 --> 00:18:06,876

The very fact that you're doing this is probably because

316

00:18:07,144 --> 00:18:10,159

the web content owner is resistant

317

00:18:10,259 --> 00:18:14,006

or unable to make the data available

318

00:18:15,343 --> 00:18:18,873

under a cooperative model.

319

00:18:19,756 --> 00:18:23,304

That means that while it may work very well one day,

320

00:18:23,404 --> 00:18:25,730

it might be prone to break in the event of things like

321

00:18:26,178 --> 00:18:27,276

changes to websites.

322

00:18:27,376 --> 00:18:29,513

Web scraping technologies can fall down.

323

00:18:30,110 --> 00:18:31,910

They are quite comparatively brittle

324

00:18:32,010 --> 00:18:33,328

in that respect.

325

00:18:33,898 --> 00:18:37,476

And there can be a whole range of different website controls in place

326

00:18:37,576 --> 00:18:41,743

to limit one's ability to do things like scraping.

327

00:18:43,570 --> 00:18:48,377

Those can include common controls that are implicit

328

00:18:48,477 --> 00:18:51,493

in many websites that are designed to

329

00:18:51,751 --> 00:18:53,098

block nuisance hosts.

330

00:18:53,198 --> 00:18:55,202

So, when you start doing things at scale,

331

00:18:55,302 --> 00:18:57,659

when you're accumulating a lot of data,

332

00:18:57,759 --> 00:19:00,431

then the behaviours, as far as the websites are concerned,

333

00:19:00,531 --> 00:19:03,097

may look like a nuisance host,

334

00:19:03,197 --> 00:19:07,611

like behaviours akin to denial of service attacks.

335

00:19:07,711 --> 00:19:10,054

You need a lot of requests in a very short space of time

336

00:19:10,154 --> 00:19:16,702

and there can be controls in place to blacklist nuisance clients

337

00:19:16,802 --> 00:19:19,018

or those identified as nuisance clients.

338

00:19:19,648 --> 00:19:23,168

And, obviously, as well as technical measures like those,

339

00:19:23,268 --> 00:19:26,535

there can also be contractual measures.

340

00:19:26,635 --> 00:19:32,119

The terms and conditions of websites that explicitly prohibit people from

341

00:19:32,219 --> 00:19:33,805

doing web scraping activities.

342

00:19:33,905 --> 00:19:35,062

And as we'll see,

343

00:19:36,059 --> 00:19:40,136

there may be those explicit contractual measures

344

00:19:40,236 --> 00:19:44,703

but, likewise, there are other parts of the law

345

00:19:44,803 --> 00:19:47,599

that offer us some encouragement and some reassurance

346

00:19:48,187 --> 00:19:52,747

that those contractual measures may not be applicable.

347

00:19:52,926 --> 00:19:56,935

We'll cover how we resolve all of these types of issues

348

00:19:57,142 --> 00:19:58,582

in the slides that follow.

349

00:20:00,022 --> 00:20:02,141

There are some alternatives to scraping, of course,

350

00:20:02,241 --> 00:20:05,292

and even without the active participation of

351

00:20:05,392 --> 00:20:06,841

Airbnb themselves.

352

00:20:07,041 --> 00:20:08,949

There are some existing online resources

353

00:20:09,049 --> 00:20:11,979

that we have reviewed and assessed.

354

00:20:13,078 --> 00:20:19,575

Two prominent data providers are Inside Airbnb and AirDNA.

355

00:20:21,441 --> 00:20:25,080

The former is

356

00:20:25,667 --> 00:20:28,525

mostly the fruits of data collection efforts

357

00:20:28,625 --> 00:20:30,882

undertaken by a small number of people

358

00:20:30,982 --> 00:20:34,499

who focused on particular geographic areas

359

00:20:34,599 --> 00:20:37,276

and particular prominent global cities

360

00:20:37,713 --> 00:20:42,521

and sought to build a picture using a range of different scraping tools.

361

00:20:43,959 --> 00:20:47,176

AirDNA is a commercial platform

362

00:20:47,276 --> 00:20:50,951

where they will actually provide access to data at commercial cost.

363

00:20:53,008 --> 00:20:54,619

Our view, having reviewed these,

364

00:20:54,719 --> 00:20:57,925

is that they do tend to suffer from a number of shortcomings,

365

00:20:58,302 --> 00:21:00,328

principally around things like limited sampling.

366

00:21:00,428 --> 00:21:03,214

So, for instance, whilst Inside Airbnb offers

367

00:21:03,711 --> 00:21:06,160

interesting data and it's been a useful means of

368

00:21:06,260 --> 00:21:10,017

calibrating and validating some of our data collection efforts,

369

00:21:10,710 --> 00:21:14,767

the sampling is quite limited to particular specific global cities

370

00:21:14,867 --> 00:21:19,086

which don't necessarily intersect with those areas of interest that we have

371

00:21:19,186 --> 00:21:20,903

within our research agenda.

372

00:21:22,953 --> 00:21:25,419

There are data quality concerns at times

373

00:21:25,519 --> 00:21:28,450

and also, I think, with AirDNA specifically,

374

00:21:28,550 --> 00:21:31,673

there was uncertainty at times of

375

00:21:31,773 --> 00:21:34,516

exactly how data was being collected

376

00:21:34,616 --> 00:21:38,722

and an issue of black box processing,

377

00:21:39,498 --> 00:21:43,159

which, from a research perspective, is particularly unattractive because

378

00:21:43,259 --> 00:21:46,883

we want to have a degree of reassurance that the data we're getting

379

00:21:46,983 --> 00:21:51,949

and the calculated variables or the inferred variables that are being

380

00:21:52,049 --> 00:21:53,555

presented within data sets,

381

00:21:53,655 --> 00:21:57,160

we have a good understanding of how those are arrived at.

382

00:21:59,225 --> 00:22:03,057

In particular, our understanding of occupancy, price, and availability,

383

00:22:03,157 --> 00:22:04,834

which are very key to our interests,

384

00:22:05,941 --> 00:22:11,095

it was not clear how the data related to those types of features was

385

00:22:11,195 --> 00:22:13,217

arrived at in the AirDNA platform.

386

00:22:13,317 --> 00:22:21,096

So, our view was that we'd be much more comfortable with

387

00:22:21,309 --> 00:22:24,355

developing our own data collection approach that

388

00:22:24,455 --> 00:22:28,901

much more directly relates to and aligns with

389

00:22:29,257 --> 00:22:32,212

the research needs that we had identified.

390

00:22:32,954 --> 00:22:37,217

So, how do you begin to develop a scraping approach?

391

00:22:37,317 --> 00:22:40,253

Well, there are various open-source projects out there

392

00:22:40,353 --> 00:22:41,527

that we initially evaluated.

393

00:22:41,627 --> 00:22:43,873

And, for example, with the Inside Airbnb platform,

394

00:22:43,973 --> 00:22:48,159

there are links to corresponding code bases that,

395

00:22:48,259 --> 00:22:51,605

where effectively you can deploy

396

00:22:51,808 --> 00:22:56,051

their tool sets in different locations on different schedules.

397

00:22:57,434 --> 00:22:59,899

But we've reviewed a number of...

398

00:22:59,999 --> 00:23:03,865

And that's the first link here, the Airbnb data collection project.

399

00:23:04,514 --> 00:23:08,821

But for each of those that we reviewed, we determined that there were problems

400

00:23:08,921 --> 00:23:14,494

and challenges with, for example, code bases that had not been updated,

401

00:23:14,594 --> 00:23:18,341

that we were struggling to deploy successfully,

402

00:23:18,441 --> 00:23:20,761

and we became increasingly aware of

403

00:23:20,861 --> 00:23:26,847

a different approach that we were keen to adopt.

404

00:23:26,947 --> 00:23:31,263

So, these projects here...

405

00:23:32,121 --> 00:23:35,661

Well, the first link here, Airbnb data collection,

406

00:23:36,320 --> 00:23:42,195

it's grabbing data directly using the Document Object Model, I think.

407

00:23:43,225 --> 00:23:46,129

The latter one here, HTTPs requests randomiser is

408

00:23:47,579 --> 00:23:52,454

a project that allows you to effectively distribute requests

409

00:23:52,554 --> 00:23:54,532

across a number of different proxy hosts

410

00:23:54,632 --> 00:23:57,121

in order to not fall foul of some of those website controls

411

00:23:57,221 --> 00:23:58,729

I described earlier on.

412

00:23:59,671 --> 00:24:03,247

We've devised our own alternative approach

413

00:24:03,347 --> 00:24:05,704

that covers both of these things that Nick will be able to explain

414

00:24:05,804 --> 00:24:08,882

much more comprehensively than I can.

415

00:24:10,468 --> 00:24:14,912

But, again, we mentioned the issue of legal and ethical issues.

416

00:24:15,012 --> 00:24:18,206

And, again, it's important to resolve these

417

00:24:18,306 --> 00:24:22,214

and to feel reassured that

418

00:24:22,679 --> 00:24:26,157

we're not falling foul of ethical or legal expectations

419

00:24:26,916 --> 00:24:29,430

in order to undertake this kind of data collection work.

420

00:24:29,530 --> 00:24:35,064

Unfortunately, despite being hopeful there would be a very clear

421

00:24:35,164 --> 00:24:40,573

and unambiguous legal basis or lawful basis for us to proceed,

422

00:24:41,766 --> 00:24:44,347

we're still very much involved in

423

00:24:45,956 --> 00:24:49,503

a broader effort to achieve clarity there

424

00:24:49,612 --> 00:24:53,793

that's probably going to prove elusive for a little bit longer.

425

00:24:55,363 --> 00:24:57,311

So, we're based at the University of Glasgow

426

00:24:57,411 --> 00:25:00,300

and we worked with the CREATe centre,

427

00:25:00,842 --> 00:25:02,939

which is also based at the University of Glasgow,

428

00:25:03,039 --> 00:25:07,317

to better understand the legal landscape that relates to online scraping.

429

00:25:07,547 --> 00:25:10,014

As I said, unfortunately the picture is not wholly clear.

430

00:25:10,114 --> 00:25:16,610

So, we were initially alerted to something within copyright law

431

00:25:16,710 --> 00:25:19,429

called the "text and data mining" exception,

432

00:25:19,529 --> 00:25:24,498

which effectively, from a lay person's perspective,

433

00:25:26,965 --> 00:25:30,206

provides a provision whereby for academic research

434

00:25:30,704 --> 00:25:36,796

one is permitted to collect at scale

435

00:25:38,185 --> 00:25:40,584

data that is made publicly available,

436

00:25:40,684 --> 00:25:44,079

that is covered by copyright but is nonetheless made publicly available.

437

00:25:44,179 --> 00:25:45,707

So, you have legal access to that.

438

00:25:46,751 --> 00:25:49,209

And under this particular exception,

439

00:25:49,309 --> 00:25:52,488

you are lawfully permitted to collect that data

440

00:25:52,588 --> 00:25:56,374

and to use it as the basis of academic research work.

441

00:25:56,863 --> 00:26:02,642

Now, that is a right that cannot simply be

442

00:26:05,080 --> 00:26:09,796

ruled out through the use of contracts and terms and conditions.

443

00:26:09,896 --> 00:26:15,252

So, it's not possible to contract your way out of that situation

444

00:26:15,352 --> 00:26:17,818

from Airbnb's point of view.

445

00:26:18,800 --> 00:26:22,067

So, we learned of this and took this to CREATe

446

00:26:22,167 --> 00:26:24,705

and felt relatively reassured that

447

00:26:25,163 --> 00:26:28,623

this would provide us our lawful basis to proceed.

448

00:26:28,723 --> 00:26:31,341

But unfortunately, as I said, the issue is not quite as clear

449

00:26:31,441 --> 00:26:35,074

as we'd hoped because, in addition to copyright law,

450

00:26:35,174 --> 00:26:37,833

there are a range of other different legal regimes

451

00:26:37,933 --> 00:26:40,761

which each intersect here.

452

00:26:41,614 --> 00:26:44,805

Including privacy, including contracts,

453

00:26:44,905 --> 00:26:47,737

and confidentiality and other intellectual property rights too

454

00:26:47,837 --> 00:26:52,015

which themselves appear to be

455

00:26:52,115 --> 00:26:55,571

operating in a somewhat contradictory fashion.

456

00:26:55,671 --> 00:26:57,635

So, although we can look at copyright law

457

00:26:57,735 --> 00:26:59,503

and feel relatively reassured,

458

00:26:59,792 --> 00:27:04,591

there may be other relevant legal precedents

459

00:27:04,860 --> 00:27:08,227

from other legal regimes that would cast some doubt

460

00:27:08,427 --> 00:27:11,095

in terms of the lawfulness of what we wish to do.

461

00:27:14,446 --> 00:27:16,669

Ultimately, until case law emerges,

462

00:27:16,769 --> 00:27:19,872

this therefore remains a risk-based policy decision.

463

00:27:19,972 --> 00:27:23,475

Do you do this on the basis of a risk-based decision.

464

00:27:23,575 --> 00:27:25,651

And some of the factors that we've considered

465

00:27:25,751 --> 00:27:28,015

and felt relatively reassured about,

466

00:27:28,823 --> 00:27:33,387

having reviewed in great detail the case law that already exists,

467

00:27:33,487 --> 00:27:35,153

there's very little if anything

468

00:27:35,253 --> 00:27:38,011

that feels relevant to our particular setting.

469

00:27:38,111 --> 00:27:40,556

It's obviously from a lot of different commercial environments

470

00:27:40,656 --> 00:27:45,581

and clearly our agenda is not to achieve some kind of commercial outcome,

471

00:27:45,681 --> 00:27:49,999

rather we're seeking to deliver academic research that will

472

00:27:50,099 --> 00:27:55,099

lead to, that will be influential to policy, perhaps,

473

00:27:55,199 --> 00:27:59,216

but without any kind of commercial expectations.

474

00:27:59,556 --> 00:28:01,144

Similarly, we looked at other academics

475

00:28:01,244 --> 00:28:05,143

who were doing similar work and feel relatively reassured that

476

00:28:05,243 --> 00:28:07,932

there is an emerging code of practice

477

00:28:08,032 --> 00:28:11,370

around how one approaches this stuff responsibly.

478

00:28:12,729 --> 00:28:16,228

The work that we did with CREATe has been

479

00:28:16,328 --> 00:28:18,114

written up into an initial working paper,

480

00:28:18,214 --> 00:28:20,073

which is linked here and you can see a DOI,

481

00:28:20,173 --> 00:28:22,833

and I would encourage people to familiarise themselves with

482

00:28:22,933 --> 00:28:25,372

the law of data scraping. And it's work that we'll

483

00:28:25,472 --> 00:28:29,068

very much continue to develop

484

00:28:29,297 --> 00:28:33,085

because there is a really important part here.

485

00:28:33,185 --> 00:28:34,900

We'll talk about some of the ways that

486

00:28:35,000 --> 00:28:38,317

Nick has innovated in very technical terms

487

00:28:39,056 --> 00:28:42,629

to achieve a means of data collection

488

00:28:42,729 --> 00:28:46,835

which will hopefully be the basis of some quite transformative research.

489

00:28:47,470 --> 00:28:48,917

But the innovation space here is

490

00:28:49,017 --> 00:28:51,700

not limited just to the technology, of course.

491

00:28:51,800 --> 00:28:54,117

And within the legal environment,

492

00:28:54,374 --> 00:28:58,414

there is a great deal of opportunity for us to leverage things like

493

00:28:59,081 --> 00:29:00,818

the "text and mining data" exception,

494

00:29:00,918 --> 00:29:03,561

but, more importantly, to leverage our understanding of the law

495

00:29:03,661 --> 00:29:05,167

and where we can innovate,

496

00:29:05,267 --> 00:29:09,924

where we can remain, at all times, on the right side of the law,

497

00:29:10,601 --> 00:29:12,281

and with a community effort

498

00:29:12,381 --> 00:29:17,298

and developing a community wide coherent approach

499

00:29:17,646 --> 00:29:19,440

where we may seek to influence the law too.

500

00:29:19,540 --> 00:29:22,019

So, the law is ultimately playing catch up a lot

501

00:29:22,119 --> 00:29:23,697

in these technical issues

502

00:29:24,634 --> 00:29:26,594

and therefore there's a real opportunity for us

503

00:29:26,694 --> 00:29:30,830

to assert how the law should behave

504

00:29:30,930 --> 00:29:33,117

and how the law should be,

505

00:29:33,434 --> 00:29:37,302

should reflect what is happening technically.

506

00:29:38,192 --> 00:29:40,771

And that's very much what we're in the midst of at the moment.

507

00:29:40,871 --> 00:29:43,259

But we felt reassured, having gone through this process,

508

00:29:43,359 --> 00:29:48,079

that the risks were within our risk tolerance

509

00:29:48,854 --> 00:29:52,361

and therefore we were comfortable proceeding.

510

00:29:52,461 --> 00:29:56,683

Ultimately, this is a decision that should be taken

511

00:29:57,572 --> 00:30:01,577

with a high degree of consideration within your own institution.

512

00:30:03,471 --> 00:30:04,798

I wanted to touch on too some of

513

00:30:04,898 --> 00:30:07,395

the other legal and ethical issues associated with these data.

514

00:30:07,495 --> 00:30:11,722

As you'll see later on when Nick showcases our platform

515

00:30:12,110 --> 00:30:14,965

and showcases what data we are focusing on,

516

00:30:15,065 --> 00:30:18,853

there are a range of things that you can collect from Airbnb.

517

00:30:19,791 --> 00:30:22,845

Those can include pictures of hosts.

518

00:30:22,945 --> 00:30:25,682

So, I mean, these are not necessarily pictures of individuals

519

00:30:25,782 --> 00:30:30,760

but rather the thumbnail image that they will have included.

520

00:30:31,516 --> 00:30:33,443

Potentially further information about hosts,

521

00:30:33,543 --> 00:30:36,292

what they've included in their "About Me" type statements

522

00:30:36,392 --> 00:30:42,840

which may or may not be personally disclosive information.

523

00:30:43,877 --> 00:30:46,875

Physical locations of where the properties themselves are.

524

00:30:48,305 --> 00:30:52,929

So, the data, while we focus on the benign data,

525

00:30:53,029 --> 00:30:56,087

the data that we don't consider to be disclosive,

526

00:30:56,326 --> 00:30:59,225

there are things in there that one can potentially collect

527

00:30:59,325 --> 00:31:01,532

that may be disclosive,

528

00:31:01,632 --> 00:31:04,919

that may draw questions around

529

00:31:05,525 --> 00:31:08,605

the association with privacy legislation, for example.

530

00:31:10,225 --> 00:31:12,784

There are also ethical questions around

531

00:31:13,002 --> 00:31:17,518

how we approach the practical aspect of

532

00:31:17,618 --> 00:31:21,366

scraping as well, and specifically how we negotiate some of these controls

533

00:31:21,610 --> 00:31:24,584

that are intended to limit the effects of nuisance hosts.

534

00:31:26,541 --> 00:31:32,477

We circumvent those by distributing requests across a proxy host network.

535

00:31:33,845 --> 00:31:39,393

And, again, within the copyright law "text and data mining" exception,

536

00:31:39,493 --> 00:31:44,472

there are important distinctions drawn between controls intended to

537

00:31:45,311 --> 00:31:49,630

protect the copyright of any content that is hosted

538

00:31:50,526 --> 00:31:53,747

and any controls or protections that are intended to

539

00:31:54,024 --> 00:31:57,070

safeguard the stability and functionality of a given website.

540

00:31:57,170 --> 00:32:04,886

And we effectively interpret the controls that Airbnb have in place

541

00:32:04,986 --> 00:32:06,405

as being for the latter.

542

00:32:08,431 --> 00:32:11,279

Our circumvention methods are therefore consistent with

543

00:32:11,955 --> 00:32:14,245

the "text and data mining" exception.

544

00:32:16,656 --> 00:32:19,244

But there are legal, well, if not legal questions,

545

00:32:19,344 --> 00:32:23,163

there are legitimate questions that may be posed.

546

00:32:24,170 --> 00:32:26,436

Clearly, when you're doing scraping at scale,

547

00:32:26,674 --> 00:32:33,213

there are costs that are being incurred by the data owner themselves

548

00:32:33,313 --> 00:32:37,313

on their platform, bandwidth costs the most obvious one.

549

00:32:38,988 --> 00:32:41,499

And these are questions that are,

550

00:32:42,907 --> 00:32:45,885

at present, not completely resolved.

551

00:32:48,216 --> 00:32:53,156

I'll close in a moment and invite any initial questions

552

00:32:53,256 --> 00:32:54,705

that we may have.

553

00:32:56,862 --> 00:33:02,751

But I'll just present this slide here around challenges,

554

00:33:04,439 --> 00:33:06,952

that Nick will pick up later on as well.

555

00:33:07,052 --> 00:33:10,079

But these are some of the things that we had to tackle

556

00:33:10,179 --> 00:33:14,322

when approaching this particular project.

557

00:33:14,422 --> 00:33:19,587

So, obviously, with a scraping approach,

558

00:33:19,687 --> 00:33:23,261

I mentioned that one of the most overwhelming characteristics of

559

00:33:23,361 --> 00:33:26,877

this data collection model is that it's a very one-sided approach.

560

00:33:26,977 --> 00:33:30,735

It's not, in any sense, a cooperation.

561

00:33:30,835 --> 00:33:34,473

In that sense, there is a lot of guessing that goes on.

562

00:33:34,573 --> 00:33:39,701

We have no straightforward means of

563

00:33:39,938 --> 00:33:43,431

looking up what data means

564

00:33:43,531 --> 00:33:45,909

and what they represent, the data that we capture.

565

00:33:46,339 --> 00:33:50,334

It's difficult for us to understand the ways that data are published

566

00:33:50,434 --> 00:33:54,702

and managed within Airbnb's online environment.

567

00:33:56,923 --> 00:34:00,886

We also have challenges that we need to manage at our own end as well

568

00:34:00,986 --> 00:34:03,820

as we get more understanding of the data that's available

569

00:34:03,920 --> 00:34:07,783

and more understanding of what the data may or may not represent.

570

00:34:07,883 --> 00:34:12,300

It evolves our specification of our needs.

571

00:34:12,400 --> 00:34:15,377

So, well, more understanding might let us

572

00:34:15,477 --> 00:34:19,405

refine our statement of what we actually need.

573

00:34:19,505 --> 00:34:24,058

and then we may have to therefore revise our methods

574

00:34:24,158 --> 00:34:29,139

to ensure that we collect more data if and when that need emerges.

575

00:34:31,487 --> 00:34:35,378

Airbnb is obviously not a static resource either.

576

00:34:35,478 --> 00:34:37,201

They are just as prone to change

577

00:34:37,301 --> 00:34:39,679

and, obviously, technical development and innovations

578

00:34:39,779 --> 00:34:41,385

that may take place on their end.

579

00:34:41,684 --> 00:34:44,573

Or, indeed, additional controls

580

00:34:44,673 --> 00:34:47,902

to limit the kind of activity that we're involved in

581

00:34:48,002 --> 00:34:50,021

that they may consider unattractive.

582

00:34:52,452 --> 00:34:54,873

So, we need to make sure that we are

583

00:34:55,073 --> 00:34:58,091

dynamic and aware and keeping up to date with

584

00:34:58,295 --> 00:34:59,712

some of these changes as well as,

585

00:35:01,169 --> 00:35:03,915

from an expectations management point of view,

586

00:35:04,015 --> 00:35:06,901

acknowledging that there may be ultimately changes that follow

587

00:35:07,089 --> 00:35:08,943

that break everything we've done

588

00:35:09,161 --> 00:35:14,233

or that render our efforts completely undone.

589

00:35:15,268 --> 00:35:16,706

There's also, obviously,

590

00:35:16,996 --> 00:35:21,106

service levels and service limitations

591

00:35:21,206 --> 00:35:24,924

that Airbnb have in place that we have to think about

592

00:35:25,024 --> 00:35:26,582

how we negotiate.

593

00:35:27,613 --> 00:35:29,874

There are limits to the number of results, for example,

594

00:35:29,974 --> 00:35:32,691

that may come out of searches that we rely upon.

595

00:35:33,224 --> 00:35:35,956

So, that obviously impacts

596

00:35:37,011 --> 00:35:39,022

how we do things at scale.

597

00:35:39,870 --> 00:35:42,520

And on that same issue of scalability,

598

00:35:42,620 --> 00:35:45,705

it's very important that if, for example, our target is

599

00:35:45,805 --> 00:35:50,461

to be able to collect data on a daily basis,

600

00:35:50,789 --> 00:35:53,994

our approach that's designed to do so has to be

601

00:35:54,094 --> 00:35:57,631

completed within a single day

602

00:35:57,731 --> 00:36:00,177

in order to ensure that it's ready to go the following day.

603

00:36:01,316 --> 00:36:03,517

And the last point is around historical data as well.

604

00:36:03,617 --> 00:36:07,145

We're grabbing data that represents the status of the Airbnb platform

605

00:36:07,541 --> 00:36:09,794

at a particular time on a particular day.

606

00:36:10,213 --> 00:36:14,684

So, there is no obvious means by which we can capture historical data.

607

00:36:14,784 --> 00:36:19,947

So, we'll look at, for example, how we approach the capture of data

608

00:36:20,047 --> 00:36:21,715

from Airbnb calendars,

609

00:36:21,815 --> 00:36:23,831

we want to see on every single individual day

610

00:36:24,132 --> 00:36:29,093

what is the status of the calendar for the forthcoming period of time

611

00:36:29,355 --> 00:36:31,016

in order to then get a better sense of

612

00:36:31,116 --> 00:36:34,706

when things were booked and when those calendars changed and so on.

613

00:36:34,800 --> 00:36:37,791

But we can't get a picture of what that calendar looked like two weeks ago

614

00:36:37,891 --> 00:36:41,474

unless we actually did the scraping activity two weeks ago.

615

00:36:41,574 --> 00:36:45,634

So, in order to build up a historical data set,

616

00:36:45,734 --> 00:36:48,113

which is essential for many of

617

00:36:48,213 --> 00:36:51,442

the research applications we would have in mind,

618

00:36:51,581 --> 00:36:53,650

you have to be doing the data collection

619

00:36:53,750 --> 00:36:54,877

historically as well.

620

00:36:54,977 --> 00:36:59,097

But Nick will summarise how we tackle some of these in due course.

621

00:37:02,816 --> 00:37:06,587

The second part of today is very much around the UBDC platform.

622

00:37:07,376 --> 00:37:10,295

I wanted to mention just a note about how you can get hold of that.

623

00:37:10,397 --> 00:37:11,533

So, it's on GitHub.

624

00:37:12,316 --> 00:37:15,776

It is available under the Apache License,

625

00:37:16,964 --> 00:37:19,314

which allows you to distribute, modify, and use,

626

00:37:19,414 --> 00:37:24,333

and we just require the original attribution notices to be preserved.

627

00:37:24,433 --> 00:37:28,531

So, we do require people to obviously acknowledge

628

00:37:29,270 --> 00:37:33,445

the source of this particular platform.

629

00:37:33,545 --> 00:37:36,561

But we're really comfortable with people using it and adapting it

630

00:37:36,661 --> 00:37:42,673

to their own data collection needs

631

00:37:43,639 --> 00:37:44,819

going forward.

632

00:37:45,999 --> 00:37:52,437

So, that was all I wanted to say for the first section.

633

00:37:52,537 --> 00:37:54,553

And I will break now for a moment

634

00:37:54,653 --> 00:38:00,242

just to invite people to supply questions.

635

00:38:01,358 --> 00:38:05,194

I see that there are a few questions

636

00:38:05,294 --> 00:38:08,423

that have already been submitted.

637

00:38:12,358 --> 00:38:14,318

I'll have a wee read through each of these

638

00:38:14,418 --> 00:38:16,957

and try and offer a response.

639

00:38:17,066 --> 00:38:18,481

So, the first I have here is,

640

00:38:18,581 --> 00:38:21,740

what are the rationale for stakeholders being fearful

641

00:38:21,840 --> 00:38:24,179

or not being collaborative in sharing data sets?

642

00:38:24,261 --> 00:38:26,270

As this would be useful if trying to gain access to

643

00:38:26,370 --> 00:38:27,928

other potential data sets.

644

00:38:28,646 --> 00:38:32,477

Well, I mean, that is very much dependent on

645

00:38:32,577 --> 00:38:35,627

the individual organisation in question.

646

00:38:36,541 --> 00:38:40,161

I think with this specific example,

647

00:38:40,635 --> 00:38:44,995

the researcher colleagues had become aware that Airbnb would be

648

00:38:45,095 --> 00:38:47,611

very resistant to making data available.

649

00:38:47,711 --> 00:38:51,570

And the most obvious assumption in terms of why that would be is that

650

00:38:51,748 --> 00:38:54,424

the type of research that is often being undertaken,

651

00:38:54,619 --> 00:38:58,374

types of associated research, will quite often present

652

00:38:58,813 --> 00:39:03,622

Airbnb and these business models in a relatively negative light

653

00:39:03,722 --> 00:39:07,524

so there is little incentive for them to cooperate with research that

654

00:39:07,624 --> 00:39:10,823

they expect to present them negatively.

655

00:39:10,923 --> 00:39:13,129

I think Airbnb, it's fair to say, are

656

00:39:14,592 --> 00:39:16,839

within a whole range of different locations,

657

00:39:16,939 --> 00:39:18,885

along with other parts of the sharing economy.

658

00:39:19,262 --> 00:39:21,172

Services and companies like Uber.

659

00:39:21,725 --> 00:39:25,696

They do face resistance within particular locations.

660

00:39:28,211 --> 00:39:31,123

The disruptive nature of these business models is such that

661

00:39:32,209 --> 00:39:37,511

they do illicit negative responses from some areas.

662

00:39:37,860 --> 00:39:39,657

And I think that there is an assumption that

663

00:39:39,867 --> 00:39:42,884

at least some of the outcomes of our research would be to be critical of

664

00:39:44,401 --> 00:39:46,239

these business models.

665

00:39:47,379 --> 00:39:49,315

As you can imagine, there is a whole range of

666

00:39:49,415 --> 00:39:54,022

other reasons why companies may be hesitant to make data available.

667

00:39:54,690 --> 00:39:57,959

The other obvious example we see from time to time is

668

00:39:58,059 --> 00:40:00,456

companies that are concerned

669

00:40:01,281 --> 00:40:05,270

when they engage with academic audiences with data provision, for example,

670

00:40:05,370 --> 00:40:06,907

they're concerned that the data may

671

00:40:08,145 --> 00:40:10,153

find its way into the hands of competitors

672

00:40:10,253 --> 00:40:13,639

or it may be used on a commercial basis

673

00:40:14,356 --> 00:40:18,896

that would be to the detriment of themselves.

674

00:40:19,754 --> 00:40:21,768

You can find just as many companies that will say,

675

00:40:21,868 --> 00:40:24,696

well, actually, there's a lot of value in engaging with academia

676

00:40:24,796 --> 00:40:26,142

with data sharing because

677

00:40:26,242 --> 00:40:28,511

it will enhance the data sets,

678

00:40:28,611 --> 00:40:35,730

it will enhance the established reputation

679

00:40:35,830 --> 00:40:39,867

and offer an endorsement of the value of the data that they make available.

680

00:40:39,967 --> 00:40:42,006

But there are many reasons

681

00:40:42,253 --> 00:40:46,404

and they're often unique to the individual business or data owner.

682

00:40:48,684 --> 00:40:49,833

But I think, ultimately,

683

00:40:50,179 --> 00:40:53,055

the ideal goal is to try and offer incentives for sharing.

684

00:40:53,155 --> 00:41:00,640

But where you have research that appears to be

685

00:41:01,947 --> 00:41:07,225

so contradictory to a particular business model,

686

00:41:07,325 --> 00:41:10,814

it's challenging to see how you would achieve a more cooperative approach.

687

00:41:12,900 --> 00:41:14,488

What happens on IP and copyright law

688

00:41:14,588 --> 00:41:17,336

applied to other jurisdictions outside the UK?

689

00:41:17,436 --> 00:41:19,754

Has this been considered and what tips can you provide?

690

00:41:21,621 --> 00:41:26,007

I mean, I should have prefaced all of the coverage of the legal issues.

691

00:41:26,107 --> 00:41:28,041

Obviously, I'm not, myself, a lawyer

692

00:41:28,141 --> 00:41:30,154

and nor is this legal advice.

693

00:41:30,350 --> 00:41:31,719

Our understanding though is that,

694

00:41:31,819 --> 00:41:34,887

certainly within the European setting,

695

00:41:34,987 --> 00:41:40,526

there is a relatively harmonised copyright regime

696

00:41:40,626 --> 00:41:43,165

and that continues even post-Brexit.

697

00:41:46,222 --> 00:41:50,962

Therefore, there are equivalent "text and data mining" exceptions

698

00:41:51,062 --> 00:41:54,513

across individual legal systems throughout Europe, for example,

699

00:41:55,000 --> 00:42:00,146

which, in some cases, offer even more robust opportunities

700

00:42:00,246 --> 00:42:01,994

for academic researchers.

701

00:42:03,143 --> 00:42:07,400

A lot of the questions around the specific legal advice here though are

702

00:42:07,896 --> 00:42:12,397

quite technical and probably not something we should

703

00:42:13,006 --> 00:42:14,151

spend too much time here on,

704

00:42:14,251 --> 00:42:18,290

nor am I sufficiently well placed

705

00:42:18,390 --> 00:42:20,648

to comment authoritatively.

706

00:42:20,748 --> 00:42:26,185

But this issue of the intersection of different legal regimes

707

00:42:26,285 --> 00:42:30,556

and the apparent contradictory nature of what they say does present challenges

708

00:42:30,656 --> 00:42:33,154

and it presents problems and it has the potential to

709

00:42:33,652 --> 00:42:38,879

offer a paralysing effect on researchers who want to do things in good faith

710

00:42:38,979 --> 00:42:41,788

but they want to do things ethically and on the right side of the law.

711

00:42:42,587 --> 00:42:44,335

One of the key questions here is,

712

00:42:44,435 --> 00:42:48,863

are the data we're collecting from Airbnb themselves

713

00:42:49,556 --> 00:42:51,134

covered by copyright law?

714

00:42:56,577 --> 00:42:59,837

If the answer is yes, you may think that means they are

715

00:42:59,937 --> 00:43:02,434

more protected and therefore less accessible to us

716

00:43:02,534 --> 00:43:04,194

but the converse is actually true because

717

00:43:04,294 --> 00:43:08,872

it is only if they are covered by copyright law

718

00:43:08,972 --> 00:43:11,051

then this exception applies.

719

00:43:11,151 --> 00:43:14,037

Whereas in the case that they are not covered by copyright law,

720

00:43:14,137 --> 00:43:16,529

then this "text and data mining" exception would not apply

721

00:43:16,629 --> 00:43:18,040

because it's not a copyright issue.

722

00:43:18,217 --> 00:43:20,572

So, there are complexities.

723

00:43:20,672 --> 00:43:26,133

There's no precedent that really speaks to the specific circumstances

724

00:43:26,315 --> 00:43:30,282

we're doing or, indeed, general questions around the legality of academic research

725

00:43:30,382 --> 00:43:34,787

using "text and data mining" exceptions

726

00:43:34,887 --> 00:43:35,934

as the lawful basis.

727

00:43:36,034 --> 00:43:39,255

So, we'll not quite wait and see.

728

00:43:39,355 --> 00:43:43,475

What we're keen to do is work with others and work with legal scholars

729

00:43:43,575 --> 00:43:45,374

to come up with what we think is

730

00:43:45,474 --> 00:43:50,062

a code of conduct and a code of practice that

731

00:43:50,162 --> 00:43:53,798

we would like to see becoming manifested within the law

732

00:43:53,898 --> 00:43:55,356

and the interpretation of the law.

733

00:43:56,225 --> 00:43:59,758

Next one, can policy really be separated from commercial outcomes?

734

00:44:01,485 --> 00:44:05,980

It's a good question and it's difficult to...

735

00:44:06,080 --> 00:44:08,400

We deal a lot in, obviously, a range of different licenses

736

00:44:08,500 --> 00:44:12,318

and the issue of what is and is not commercial use of

737

00:44:12,418 --> 00:44:17,692

particular data sets is a perennial challenge

738

00:44:17,792 --> 00:44:24,430

and there remains no clear definition that we think is completely water tight.

739

00:44:25,569 --> 00:44:26,914

I think that you're right.

740

00:44:27,014 --> 00:44:29,442

We don't seek to benefit commercially ourselves

741

00:44:29,542 --> 00:44:32,159

and that's perhaps what we would have at the heart of

742

00:44:32,259 --> 00:44:36,658

our own internal justification for this.

743

00:44:36,758 --> 00:44:38,447

Or rather, our own kind of

744

00:44:38,547 --> 00:44:40,436

rationalisation of risk.

745

00:44:42,132 --> 00:44:47,691

I think, in that respect, the goal of

746

00:44:50,549 --> 00:44:52,760

the academic sector here is

747

00:44:53,635 --> 00:44:59,346

not to generate commercial value from these data.

748

00:44:59,446 --> 00:45:02,213

It may no doubt inevitably lead to

749

00:45:02,981 --> 00:45:06,039

commercial impacts on others

750

00:45:06,100 --> 00:45:08,220

who are operating within those kinds of environments.

751

00:45:08,250 --> 00:45:10,229

But, from our own point of view,

752

00:45:10,500 --> 00:45:12,574

I think we feel relatively reassured that

753

00:45:13,658 --> 00:45:18,011

we're not establishing ourselves as a competitor to Airbnb

754

00:45:18,111 --> 00:45:19,740

but rather, we are interested in

755

00:45:21,050 --> 00:45:24,445

providing clarity on some of the policy recommendations

756

00:45:24,545 --> 00:45:29,244

and some of the policy impacts of businesses operating like Airbnb are.

757

00:45:29,675 --> 00:45:31,793

If that isn't too fudgy an answer.

758

00:45:32,792 --> 00:45:38,592

How are you not to be seen as a security threat?

759

00:45:40,986 --> 00:45:43,775

I think that that's...

760

00:45:43,875 --> 00:45:46,943

The importance of some of the things Nick will touch on briefly I suppose is

761

00:45:47,043 --> 00:45:50,135

how we ensure that

762

00:45:50,802 --> 00:45:56,051

we are not interpreted, at least through their automated controls,

763

00:45:56,151 --> 00:45:57,364

as being a security threat.

764

00:45:57,464 --> 00:45:59,715

You may argue that our methods are

765

00:46:02,283 --> 00:46:05,531

still kind of akin to what they may expect to see from

766

00:46:05,631 --> 00:46:08,030

people who would be security threats.

767

00:46:08,340 --> 00:46:10,868

However, again, we are comfortable

768

00:46:10,968 --> 00:46:13,556

with our understanding of what we wish to do with these data

769

00:46:13,656 --> 00:46:16,716

and the value that we want to deliver.

770

00:46:17,974 --> 00:46:20,464

So, in that respect, our main priority is just to

771

00:46:20,793 --> 00:46:25,031

ensure that we don't set off the technical measures

772

00:46:25,131 --> 00:46:27,909

that would impede our efforts to collect these data.

773

00:46:29,105 --> 00:46:31,542

Even if legally there is no restriction on scraping the content, are there

774

00:46:31,642 --> 00:46:32,941

any legal restrictions on reporting findings?

775

00:46:33,041 --> 00:46:34,404

Well, this is an important point.

776

00:46:35,070 --> 00:46:37,811

And, again, this is something we're

777

00:46:37,911 --> 00:46:41,579

really quite in the dark about.

778

00:46:41,722 --> 00:46:43,729

So, it's a very good point.

779

00:46:43,829 --> 00:46:48,607

So, the "text and data mining" exception is

780

00:46:48,707 --> 00:46:51,959

explicitly around data collection and the use of that data for research.

781

00:46:52,059 --> 00:46:56,734

But it says nothing at all of any...

782

00:46:57,339 --> 00:46:58,766

There's no real clarity about

783

00:46:58,866 --> 00:47:01,244

what you can actually do with the results of that research,

784

00:47:01,344 --> 00:47:03,622

what you may be able to republish, for example.

785

00:47:04,394 --> 00:47:05,482

So, we can't...

786

00:47:06,194 --> 00:47:10,222

We feel fairly clear that we can't circulate the data that we've collected.

787

00:47:10,322 --> 00:47:12,531

So, instead what we're doing as a data service here is

788

00:47:12,631 --> 00:47:14,787

we're saying let's promote our code, let's promote our methods,

789

00:47:14,887 --> 00:47:16,917

let's encourage other people to do the same kind of thing

790

00:47:17,017 --> 00:47:20,831

or at least give them the technical means

791

00:47:20,931 --> 00:47:23,645

and some sense of the legal understanding

792

00:47:23,745 --> 00:47:25,271

to make a decision to do so.

793

00:47:25,371 --> 00:47:26,948

We can't share the data sets though.

794

00:47:27,048 --> 00:47:28,407

That feels very clear.

795

00:47:28,900 --> 00:47:32,389

What we can even publish within the context of a research article is,

796

00:47:32,489 --> 00:47:34,188

again, not spelled out in clear terms

797

00:47:34,288 --> 00:47:36,906

so we'll ultimately be basing it on

798

00:47:37,006 --> 00:47:39,424

some kind of an informed risk assessment.

799

00:47:39,524 --> 00:47:41,819

And maybe you can make an argument, for instance,

800

00:47:41,919 --> 00:47:45,347

to say that, implicitly, to do academic research work

801

00:47:45,447 --> 00:47:48,392

you're going to publish and implicitly, if you're doing

802

00:47:48,738 --> 00:47:50,646

academic publications within a certain setting,

803

00:47:50,746 --> 00:47:52,931

there are norms, there are expectations around

804

00:47:53,031 --> 00:47:57,940

what kinds of data should accompany the results within an academic paper.

805

00:47:59,030 --> 00:48:00,837

That's something we're still wrestling with.

806

00:48:00,937 --> 00:48:04,072

But, again, it's an unfortunate lack of clarity

807

00:48:04,172 --> 00:48:08,829

around exactly what one can do

808

00:48:08,929 --> 00:48:11,195

with the data

809

00:48:11,500 --> 00:48:15,007

in terms of resharing, in terms of circulating,

810

00:48:16,204 --> 00:48:19,041

even within the context of

811

00:48:21,216 --> 00:48:23,574

a credible academic publication.

812

00:48:25,094 --> 00:48:29,233

So, this area is mired in some uncertainty

813

00:48:29,333 --> 00:48:32,051

and certainly in comparison with a very neat and elegant

814

00:48:32,799 --> 00:48:35,815

data licensing agreement where your permitted purpose is spelled out

815

00:48:37,032 --> 00:48:40,270

and signed up to,

816

00:48:40,927 --> 00:48:45,044

where expectations are explicitly managed and mutually agreed.

817

00:48:45,144 --> 00:48:52,262

This is comparatively untidy,

818

00:48:52,362 --> 00:48:54,640

I suppose you could say, this method of data collection.

819

00:48:54,740 --> 00:48:56,077

But it's a means to an end

820

00:48:56,177 --> 00:48:59,916

that would otherwise be very difficult to achieve.

821

00:49:02,163 --> 00:49:06,295

I'm not exactly sure what the programme for next week,

822

00:49:06,484 --> 00:49:08,022

for, beg your pardon, for Thursday's

823

00:49:08,742 --> 00:49:11,309

much more research focused coverage is

824

00:49:11,409 --> 00:49:13,877

but that is a very pertinent question

825

00:49:14,554 --> 00:49:17,009

within that presentation as well.

826

00:49:17,109 --> 00:49:20,367

Not just how do we get this stuff

827

00:49:20,467 --> 00:49:22,895

but, ultimately, we need to be reassured that

828

00:49:22,995 --> 00:49:27,123

you can actually do with that data what you wish.

829

00:49:28,157 --> 00:49:31,664

So, this is the main part of the agenda now

830

00:49:31,764 --> 00:49:36,351

where we'll see the platform that Nick has developed.

831

00:49:36,451 --> 00:49:39,251

And he will take it away.

832

00:49:39,351 --> 00:49:40,430

So, thanks, Nick.

833

00:49:41,489 --> 00:49:44,097

Hi, all. Good morning.

834

00:49:47,366 --> 00:49:48,968

We are going to demonstrate

835

00:49:49,068 --> 00:49:54,483

this new platform that we've developed here at UBDC

836

00:49:54,723 --> 00:50:00,039

to scrape or capture the data from Airbnb.

837

00:50:00,796 --> 00:50:04,293

The objective of my part here is

838

00:50:04,393 --> 00:50:09,859

basically to demonstrate how we set up a data collection service for Airbnb

839

00:50:09,959 --> 00:50:14,007

and hopefully to kickstart anyone who wishes to do

840

00:50:15,400 --> 00:50:16,518

something similar.

841

00:50:16,918 --> 00:50:19,407

Now, apologies over here because

842

00:50:19,507 --> 00:50:22,875

I want to share the presentation

843

00:50:22,975 --> 00:50:26,222

and also the desktop that shows these steps.

844

00:50:26,322 --> 00:50:31,737

So, bear with me for a little bit until I figure out how to share screens.

845

00:50:38,302 --> 00:50:40,538

Okay. You guys can see my presentation.

846

00:50:42,387 --> 00:50:43,514

Hopefully you can.

847

00:50:45,815 --> 00:50:46,821

What I was trying to say.

848

00:50:47,138 --> 00:50:51,003

It's a little bit of history over here.

849

00:50:51,103 --> 00:50:54,251

It's that we've been trying to

850

00:50:56,261 --> 00:50:58,662

capture the data from Airbnb.

851

00:50:58,943 --> 00:51:00,553

And there were several problems.

852

00:51:01,110 --> 00:51:04,228

We initially tried a more direct approach.

853

00:51:04,328 --> 00:51:08,616

And no doubt you're familiar with how the Airbnb website operates.

854

00:51:08,716 --> 00:51:13,593

It's that you get a search page

855

00:51:13,693 --> 00:51:17,282

where you put your query and where you get the data.

856

00:51:17,382 --> 00:51:20,239

Well, something similar which I like to do initially is that

857

00:51:20,339 --> 00:51:22,928

for each search result is go for its listings

858

00:51:23,028 --> 00:51:26,437

and then for its listing to get the data from it

859

00:51:26,617 --> 00:51:29,415

and extract the data from the listing pages

860

00:51:29,515 --> 00:51:31,663

and then check in the database

861

00:51:31,763 --> 00:51:33,660

and hopefully repeat.

862

00:51:34,801 --> 00:51:36,942

There's nothing wrong about this approach initially

863

00:51:37,042 --> 00:51:38,650

but there is, as mentioned before,

864

00:51:39,778 --> 00:51:41,637

Airbnb does have some limitations.

865

00:51:42,585 --> 00:51:44,689

Now really the biggest limitations we could find.

866

00:51:44,789 --> 00:51:48,357

It's for quality reasons results,

867

00:51:49,142 --> 00:51:53,292

Airbnb only returns 300 results per query

868

00:51:53,982 --> 00:51:56,050

per search. So, they think,

869

00:51:56,347 --> 00:52:02,115

and rightfully so, that no one will go over 300 results

870

00:52:02,422 --> 00:52:06,429

to find their perfect match for their holidays.

871

00:52:06,749 --> 00:52:09,667

Usually, users are finding whatever they want to find

872

00:52:09,767 --> 00:52:12,425

within 50 or 60 results

873

00:52:12,603 --> 00:52:14,891

so they don't bother going any further because

874

00:52:14,991 --> 00:52:16,425

it would take too many resources to

875

00:52:16,525 --> 00:52:20,291

to find all the listings that would fit into their context.

876

00:52:20,596 --> 00:52:24,594

So, high end results.

877

00:52:24,744 --> 00:52:28,061

So, high end queries, it's out of the question.

878

00:52:28,219 --> 00:52:30,077

For example, Edinburgh, if you put Edinburgh,

879

00:52:30,177 --> 00:52:32,973

it has around 10,000 listings or somewhere around there.

880

00:52:33,073 --> 00:52:34,684

So, in the first iteration,

881

00:52:34,784 --> 00:52:36,542

you cannot find through these methods,

882

00:52:36,642 --> 00:52:37,848

you can't find everything.

883

00:52:38,316 --> 00:52:43,385

And if you are familiar with programming,

884

00:52:43,485 --> 00:52:46,773

it's basically one big linear approach over here.

885

00:52:46,873 --> 00:52:48,502

So, if something breaks

886

00:52:48,602 --> 00:52:51,451

and it's within the search results,

887

00:52:51,551 --> 00:52:54,979

then you can't really restart where you picked it up because

888

00:52:55,079 --> 00:52:58,177

every search result is unique every time

889

00:52:58,277 --> 00:52:59,906

that you're searching something,

890

00:53:00,006 --> 00:53:02,404

their engine behind creates a ticket,

891

00:53:02,504 --> 00:53:04,561

and the results are unique for you

892

00:53:04,661 --> 00:53:06,720

based on the day or based on where you are,

893

00:53:06,820 --> 00:53:10,169

based on the many different parameters that they put down.

894

00:53:11,038 --> 00:53:15,216

And even if you were able to go around that,

895

00:53:17,244 --> 00:53:20,732

Airbnb is constantly evolving their model.

896

00:53:20,832 --> 00:53:23,210

They're constantly putting in new features

897

00:53:24,460 --> 00:53:26,307

or they're taking out other features.

898

00:53:26,696 --> 00:53:28,963

Like, for example, we do have...

899

00:53:29,840 --> 00:53:31,618

If you've recently been to their website,

900

00:53:31,718 --> 00:53:33,937

they reacted to COVID.

901

00:53:34,037 --> 00:53:40,295

Is the listings available for booking during this COVID situation?

902

00:53:40,395 --> 00:53:46,002

Has the UK government rolled out any regulations about COVID?

903

00:53:46,102 --> 00:53:50,400

So, all of them, they need to

904

00:53:51,305 --> 00:53:53,195
be reflected in their website.

905
00:53:53,295 --> 00:53:57,908
So, it's a constant game of

906
00:53:58,008 --> 00:53:59,154
mouse and cat here,

907
00:53:59,254 --> 00:54:01,954
where one's changing and the other has to react.

908
00:54:03,474 --> 00:54:05,601
If going for traditional methods,

909
00:54:05,701 --> 00:54:08,709
the biggest problem is that,

910
00:54:08,809 --> 00:54:11,149
not the biggest problem, one of the interesting problems is

911
00:54:11,226 --> 00:54:14,201
that after you get the data from the HTML,

912

00:54:14,500 --> 00:54:17,335

you are discarding it, you are throwing it away,

913

00:54:17,435 --> 00:54:18,814

you don't keep it.

914

00:54:18,914 --> 00:54:21,486

So, you can't really compare

915

00:54:21,586 --> 00:54:24,776

what you have in the past to what we have in the future,

916

00:54:24,900 --> 00:54:26,298

what you will have in the future.

917

00:54:26,398 --> 00:54:28,865

Or maybe you forgot to

918

00:54:28,965 --> 00:54:32,173

pick up something that you should have

919

00:54:32,573 --> 00:54:35,621

and later on you figure out that, okay, maybe I needed that.

920

00:54:35,721 --> 00:54:36,990

But it's too late.

921

00:54:37,880 --> 00:54:41,536

So, well, having these tried things in mind,

922

00:54:41,636 --> 00:54:46,768

I've similarly looked very carefully at the website.

923

00:54:47,006 --> 00:54:49,334

It's not a static website.

924

00:54:49,434 --> 00:54:51,572

It's basically a responsive web application,

925

00:54:51,672 --> 00:54:54,073

which is a combination itself of

926

00:54:54,173 --> 00:54:58,652

two elements, the HTML element and the JavaScript element.

927

00:54:58,741 --> 00:55:02,144

The HTML code, basically, it's a placeholder, a template,

928

00:55:02,327 --> 00:55:04,594

where its an initial starting place,

929

00:55:04,694 --> 00:55:09,402

it provides the cards, provides places where you put

930

00:55:09,750 --> 00:55:11,347

the information.

931

00:55:11,666 --> 00:55:13,064

And after it's loaded,

932

00:55:13,164 --> 00:55:16,220

it has a JavaScript code in the background

933

00:55:16,320 --> 00:55:19,997

where it kicks in and then starts grabbing extra information

934

00:55:20,217 --> 00:55:24,586

and it sort of hydrates the template

935

00:55:24,686 --> 00:55:26,072

to populate it.

936

00:55:26,571 --> 00:55:29,850

For example, when you load the website

937

00:55:29,950 --> 00:55:32,200

and, afterwards, the JavaScript,

938

00:55:32,300 --> 00:55:35,816

it's requesting extra data from an API endpoint.

939

00:55:35,916 --> 00:55:40,159

For example, what listings there are within this context.

940

00:55:40,259 --> 00:55:42,025

And they get those listings

941

00:55:42,125 --> 00:55:46,047

and then they make sure to create extra HTML elements too.

942

00:55:46,147 --> 00:55:48,523

So, what's been happening.

943

00:55:49,528 --> 00:55:51,166

That's a very interesting concept.

944

00:55:51,266 --> 00:55:53,642

So, basically, one could do...

945

00:55:54,065 --> 00:55:56,753

So, instead of going directly for HTML code,

946

00:55:56,853 --> 00:55:58,311

which is like the end result,

947

00:55:59,639 --> 00:56:04,566

hijack or do the same thing as what the JavaScript does.

948

00:56:04,855 --> 00:56:06,596

Hit these API points

949

00:56:10,114 --> 00:56:13,112

and instead of letting them make the requests for data,

950

00:56:13,212 --> 00:56:15,079

you are making them.

951

00:56:16,390 --> 00:56:20,578

So, if someone is not very familiar with what an API is,

952

00:56:20,678 --> 00:56:24,196

basically, it's a well-known application on the web

953

00:56:24,384 --> 00:56:26,700

where you're requesting, not requesting,

954

00:56:26,800 --> 00:56:28,458

you're hitting them with your web browser

955

00:56:28,558 --> 00:56:33,595

and then instead of coming back with an HTML document,

956

00:56:34,200 --> 00:56:37,479

they come back with a string

957

00:56:37,579 --> 00:56:42,251

which follows a JSON, which is basically a key and a value.

958

00:56:42,351 --> 00:56:46,154

So, you got the ID equals the listing ID,

959

00:56:46,254 --> 00:56:47,959

the number, a listing ID includes a number.

960

00:56:48,482 --> 00:56:51,161

Or is it available? Yes, no.

961

00:56:51,413 --> 00:56:54,465

Is it for what is the price for the day?

962

00:56:54,585 --> 00:56:56,180

That number. And etc.

963

00:56:56,280 --> 00:57:01,757

And imagine that you've got a very, very big stream of data.

964

00:57:01,964 --> 00:57:08,405

It's representing all kinds of information in this matter.

965

00:57:10,954 --> 00:57:15,022

So, the easy way to

966

00:57:15,122 --> 00:57:21,346

talk with Airbnb is basically to use the API.

967

00:57:26,415 --> 00:57:30,288

As mentioned before, you are hitting this endpoint

968

00:57:30,388 --> 00:57:31,626

with your browser.

969

00:57:32,299 --> 00:57:34,006

Alternatively, you could do that manually

970

00:57:34,106 --> 00:57:37,933

if you put some URLs in your web browser,

971

00:57:38,033 --> 00:57:41,121

you could do that and you would get the appropriate response.

972

00:57:41,371 --> 00:57:43,580

But, alternatively, what you're usually doing

973

00:57:43,680 --> 00:57:45,229

within a programming environment is

974

00:57:45,329 --> 00:57:50,574

that you are letting your favourite language construct these requests.

975

00:57:50,674 --> 00:57:54,923

And then your language or your code is

976

00:57:55,965 --> 00:57:57,563

creating and generating these requests

977

00:57:57,663 --> 00:57:59,212

and they're grabbing that data for you

978

00:57:59,312 --> 00:58:06,011

in a programmatical environment.

979

00:58:08,449 --> 00:58:10,726

We have developed, not developed.

980

00:58:10,895 --> 00:58:12,729

We started with...

981

00:58:13,976 --> 00:58:16,863

If you do a little bit of searching on GitHub,

982

00:58:17,147 --> 00:58:20,755

there are multiple libraries

983

00:58:20,855 --> 00:58:22,912

that do that for Airbnb.

984

00:58:23,489 --> 00:58:26,467

They've not been updated for a long time

985

00:58:26,567 --> 00:58:30,027

or they've been abandoned or basically they've been so changed that

986

00:58:30,127 --> 00:58:36,725

there is constant evolution or interest of people

987

00:58:36,825 --> 00:58:38,815

trying to collect data from Airbnb.

988

00:58:39,273 --> 00:58:41,102

Many of them, they have their own limitations,

989

00:58:41,202 --> 00:58:42,600

are a little more current,

990

00:58:42,700 --> 00:58:44,238

but they paint a picture.

991

00:58:44,338 --> 00:58:47,726

And if you are interested in some history,

992

00:58:48,235 --> 00:58:49,323

it's quite interesting.

993

00:58:49,423 --> 00:58:52,475

It shows that, back in the day,

994

00:58:52,575 --> 00:58:56,619

Airbnb had a public API where it was sharing the data.

995

00:58:56,719 --> 00:58:58,803

They decided, at some point, to stop doing that.

996

00:58:58,903 --> 00:59:02,393

But it had published a white paper

997

00:59:02,493 --> 00:59:07,061

that was cataloguing and explaining all their APIs,

998

00:59:07,161 --> 00:59:08,470

the endpoints that they had,

999

00:59:08,570 --> 00:59:12,427

and how you could request for access,

1000

00:59:12,527 --> 00:59:13,980

and how they were controlling you.

1001

00:59:14,071 --> 00:59:16,550

That think have been deprecated since then.

1002

00:59:16,650 --> 00:59:21,994

But people use this information to collect all the data,

1003

00:59:22,094 --> 00:59:26,387

all this knowledge into sql databases.

1004

00:59:26,668 --> 00:59:29,505

And the one that we started with was

1005

00:59:31,443 --> 00:59:36,321

from, this is wrong, it was from another one.

1006

00:59:38,226 --> 00:59:40,904

Someone else's effort.

1007

00:59:41,004 --> 00:59:42,980

And we extended the model, the library,

1008

00:59:43,080 --> 00:59:44,929

to do some of the stuff that we needed.

1009

00:59:45,587 --> 00:59:48,286

And, basically, this is...

1010

00:59:49,245 --> 00:59:51,513

So, basically, after you implement

1011

00:59:51,613 --> 00:59:54,161

this kind of library,

1012

00:59:55,400 --> 00:59:58,659

you can talk directly to Airbnb.

1013

00:59:58,759 --> 01:00:00,637

For example, if you're familiar with Python,

1014

01:00:00,737 --> 01:00:04,202

you import the name of the library called Airbnb,

1015

01:00:04,396 --> 01:00:07,765

you are creating an API object, and then you get the data.

1016

01:00:07,865 --> 01:00:10,471

For example, here's the new thing that we added.

1017

01:00:10,727 --> 01:00:14,903

It's that you can get the data from this bounding box.

1018

01:00:15,182 --> 01:00:16,716

It gives me all the data.

1019

01:00:17,137 --> 01:00:21,508

And basically what comes back is a JSON.

1020

01:00:21,608 --> 01:00:25,128

This JSON has then been translated to a python dictionary.

1021

01:00:25,375 --> 01:00:27,481

And then you take it over with your code

1022

01:00:27,581 --> 01:00:31,428

to do whatever you need to do.

1023

01:00:31,720 --> 01:00:33,177

And what comes back,

1024

01:00:33,277 --> 01:00:34,765

it's something very structured.

1025

01:00:34,865 --> 01:00:37,193

This is another endpoint

1026

01:00:37,293 --> 01:00:39,430

which gives back information about

1027

01:00:39,530 --> 01:00:41,708

the details of the listing IDs.

1028

01:00:41,909 --> 01:00:44,507

So, this ID, it has these coordinates,

1029

01:00:44,607 --> 01:00:46,247

it's in this place,

1030

01:00:46,347 --> 01:00:50,995

and has all the extra stuff that one might need,

1031

01:00:51,216 --> 01:00:52,226

or not.

1032

01:00:53,550 --> 01:00:57,181

So, that is at the heart of what we're trying to do.

1033

01:00:57,281 --> 01:01:02,430

It's that we now have the means of talking to Airbnb.

1034

01:01:02,530 --> 01:01:07,148

So, we built a framework, a web, no, not a framework.

1035

01:01:07,248 --> 01:01:11,616

A system around that that manages and collects

1036

01:01:11,716 --> 01:01:16,009

and scales and puts everything

1037

01:01:16,109 --> 01:01:23,304

into a more organised way to help collect the data.

1038

01:01:24,131 --> 01:01:25,537

So, for our solution over here,

1039

01:01:25,637 --> 01:01:27,406

obviously you can...

1040

01:01:27,965 --> 01:01:30,075

I'll demonstrate what has been happening.

1041

01:01:30,484 --> 01:01:32,193

But if you wanted to do it by yourself,

1042

01:01:32,293 --> 01:01:35,332

obviously you will need some kind of prerequirement.

1043

01:01:35,521 --> 01:01:38,728

And the biggest one would be Docker.

1044

01:01:38,828 --> 01:01:41,292

And since everything, as I like to put it, makes

1045

01:01:41,392 --> 01:01:45,710

the solution as pain free as possible.

1046

01:01:45,979 --> 01:01:48,067

So, if you're not familiar with Docker,

1047

01:01:48,167 --> 01:01:51,820

basically what it does is it's like a mini virtual machines

1048

01:01:51,920 --> 01:01:56,809

where you are opening them as you see fit, as needed

1049

01:01:56,909 --> 01:02:00,787

and all the code, including the operating system,

1050

01:02:00,887 --> 01:02:03,494

the parameters, the configuration or whatnot,

1051

01:02:03,594 --> 01:02:04,663

the network.

1052

01:02:04,953 --> 01:02:09,377

Everything is resting inside this little bit of information.

1053

01:02:10,329 --> 01:02:12,244

QGIS, we do need QGIS.

1054

01:02:12,766 --> 01:02:17,954

I'm using it at the moment as a means to communicate with the framework

1055

01:02:18,054 --> 01:02:19,172

from outside.

1056

01:02:19,772 --> 01:02:21,528

Git obviously to...

1057

01:02:22,499 --> 01:02:25,485

QGIS, if you are not familiar with it, is a GIS client,

1058

01:02:25,694 --> 01:02:30,693

basically allows you to create geospatial features

1059

01:02:31,961 --> 01:02:34,567

and do some geospatial analysis if you need it.

1060

01:02:34,856 --> 01:02:36,075

But we're not going into that.

1061

01:02:36,624 --> 01:02:39,034

And Git, obviously, to download the code.

1062

01:02:39,294 --> 01:02:42,022

And some other optional stuff

1063

01:02:42,122 --> 01:02:43,530

that you might find useful.

1064

01:02:43,630 --> 01:02:45,038

It is for me.

1065

01:02:45,419 --> 01:02:47,847

[DBeaver] It's a database client.

1066

01:02:49,683 --> 01:02:54,343

All the data that comes from Airbnb is stored in the database.

1067

01:02:54,443 --> 01:02:58,785

So, it's nice that you have something that can visualise and store,

1068

01:02:59,213 --> 01:03:01,920

and extract the data later on

1069

01:03:02,020 --> 01:03:05,058

and do some kind of transformation.

1070

01:03:05,207 --> 01:03:07,174

And a fancy text editor

1071

01:03:07,274 --> 01:03:11,154

where, as I said, the response is coming back as JSON.

1072

01:03:11,605 --> 01:03:15,674

So, a new text editor allows you

1073

01:03:15,774 --> 01:03:17,942

to properly format these strings

1074

01:03:18,042 --> 01:03:22,231

so you can more more easily understand what is happening here.

1075

01:03:22,880 --> 01:03:27,412

Alright. Now, a little bit of architecture.

1076

01:03:27,512 --> 01:03:28,721

Or what's been happening.

1077

01:03:28,821 --> 01:03:31,281

The whole concept has been...

1078

01:03:33,149 --> 01:03:35,507

One of the problems that we had...

1079

01:03:36,883 --> 01:03:40,281

Not problems. One of the requirements of the project

1080

01:03:40,381 --> 01:03:45,817

that we need to finish everything within a certain partner set

1081

01:03:45,917 --> 01:03:47,095

and amount of time.

1082

01:03:47,177 --> 01:03:52,796

So, we are going for the producer-consumer architecture,

1083

01:03:53,161 --> 01:03:56,925

which is basically the producer creating tasks

1084

01:03:57,025 --> 01:04:00,419

and the consumer is consuming the tasks.

1085

01:04:00,979 --> 01:04:03,527

A task, it's basically a message which says

1086

01:04:05,106 --> 01:04:08,075

that, in very simple terms, that, okay:

1087

01:04:09,484 --> 01:04:11,440

Here is a message with a header which says

1088

01:04:11,540 --> 01:04:14,760

you're going to run this command by name

1089

01:04:14,860 --> 01:04:17,239

and you are going to use these parameters by name.

1090

01:04:17,720 --> 01:04:21,756

And this message is then sent to a broker,

1091

01:04:21,856 --> 01:04:25,074

which basically you can parallelize as a post office,

1092

01:04:25,513 --> 01:04:28,719

and it's built like a queue.

1093

01:04:29,797 --> 01:04:32,455

That queue could contain like a million messages

1094

01:04:32,733 --> 01:04:35,082

and they're waiting for someone to pick them up.

1095

01:04:35,183 --> 01:04:39,531

So, these queues are going to be consumed by workers

1096

01:04:39,995 --> 01:04:42,053

and they're picking up the first message

1097

01:04:42,153 --> 01:04:45,043

and they're reading, okay, which function do I have to write, to run?

1098

01:04:45,143 --> 01:04:47,580

I have to run this one so which parameter is this parameter?

1099

01:04:47,908 --> 01:04:49,023

Okay, leave it to me.

1100

01:04:49,432 --> 01:04:52,601

So, the worker will now know

1101

01:04:52,701 --> 01:04:56,091

what kind of function to run

1102

01:04:56,191 --> 01:04:58,849

and which parameters will go and fetch the data.

1103

01:04:59,200 --> 01:05:01,389

And they will consume the data

1104

01:05:01,489 --> 01:05:06,239

and then will throw the result information

1105

01:05:06,339 --> 01:05:07,350

back to the database,

1106

01:05:07,450 --> 01:05:12,469

which then will be available to the framework that we use.

1107

01:05:12,569 --> 01:05:16,420

And it shows us a framework, Django, here.

1108

01:05:16,520 --> 01:05:21,158

And if you're not familiar with it, it's a web application framework.

1109

01:05:22,823 --> 01:05:25,350

It's made, primarily, to create websites.

1110

01:05:25,450 --> 01:05:28,227

It's a fully managed solution.

1111

01:05:28,327 --> 01:05:30,221

And one of the neat features that it does is

1112

01:05:30,321 --> 01:05:35,878

it has its own built-in database management system.

1113

01:05:35,978 --> 01:05:39,660

So, it creates tables, it creates databases,

1114

01:05:39,760 --> 01:05:44,009

it creates everything that queries the data.

1115

01:05:44,109 --> 01:05:48,737

So, everything that gives you all the tools to go ahead,

1116

01:05:50,596 --> 01:05:54,055

manage to shuffle through the information that's happening.

1117

01:05:54,335 --> 01:05:59,515

So, me, through that framework,

1118

01:05:59,615 --> 01:06:01,483

I'm generating the tasks.

1119

01:06:02,119 --> 01:06:04,759

So, for example, I'm telling it to give me

1120

01:06:04,859 --> 01:06:08,567

all the listings that have not been updated within one day of

1121

01:06:08,667 --> 01:06:11,026

the framework queries database.

1122

01:06:11,704 --> 01:06:15,504

The database, tells it to gather these 3000 listings that

1123

01:06:15,604 --> 01:06:18,623

they are stalled when creating the tasks.

1124

01:06:18,782 --> 01:06:21,010

And I am giving them to the post office.

1125

01:06:21,110 --> 01:06:25,034

The post office then has a whole query queue of

1126

01:06:28,201 --> 01:06:31,569

3000 or X number of tasks that

1127

01:06:31,669 --> 01:06:33,136

they are still there pending.

1128

01:06:33,269 --> 01:06:35,160

And the workers will come one by one,

1129

01:06:35,260 --> 01:06:37,508

picking it up, and telling them, "Okay, I'm on it",

1130

01:06:37,608 --> 01:06:38,995

and they finish it or not.

1131

01:06:41,019 --> 01:06:45,407

So, it might sound a little bit complicated

1132

01:06:45,507 --> 01:06:48,274

to people who have not done something similar to this.

1133

01:06:49,613 --> 01:06:54,237

So, as I said, it's...

1134

01:06:54,684 --> 01:06:57,721

I will go into something more hands-on.

1135

01:06:58,260 --> 01:07:00,336

So, throughout the slides,

1136

01:07:00,436 --> 01:07:03,220

we will really demonstrate the following steps on how to install,

1137

01:07:03,320 --> 01:07:06,037

configuring UBDC, that platform.

1138

01:07:06,818 --> 01:07:08,942

How we can create an area of interest

1139

01:07:09,140 --> 01:07:11,679

that we can use for data collection.

1140

01:07:12,028 --> 01:07:16,315

And discover listings within that area of interest.

1141

01:07:16,903 --> 01:07:19,192

The details for the listings.

1142

01:07:19,292 --> 01:07:20,391

Calendar information.

1143

01:07:20,491 --> 01:07:23,279

Reviews and maybe booking quotes.

1144

01:07:23,595 --> 01:07:28,467

And also, how to configure for scale and production.

1145

01:07:29,948 --> 01:07:31,577

Well, without further ado,

1146

01:07:31,677 --> 01:07:33,815

let's go and do that.

1147

01:07:36,184 --> 01:07:39,259

Obviously, we have...

1148

01:07:39,359 --> 01:07:45,058

Now, I'm just going to scroll down that.

1149

01:07:45,358 --> 01:07:49,322

And we have set up a virtual machine with

1150

01:07:49,422 --> 01:07:51,028

already we have Docker here running.

1151

01:07:51,128 --> 01:07:54,219

We've got Git and we've got all the functions entered

1152

01:07:54,613 --> 01:07:56,418

and a database connection in postgres.

1153

01:07:57,015 --> 01:08:00,933

Obviously, the first step is to...

1154

01:08:02,080 --> 01:08:04,202

The first step is always to download the code.

1155

01:08:06,225 --> 01:08:09,436

Maybe if you're familiar with GitHub.

1156

01:08:09,536 --> 01:08:14,404

If you've done clone, the code is something like that.

1157

01:08:15,556 --> 01:08:19,604

So, git clone, the URL that we spoke about before.

1158

01:08:19,786 --> 01:08:22,684

And that will clone it to this folder.

1159

01:08:22,790 --> 01:08:26,421

I have already done these steps so I will not do it.

1160

01:08:28,549 --> 01:08:31,059

In this folder.

1161

01:08:31,255 --> 01:08:36,765

And here, we now have all the code that runs,

1162

01:08:36,993 --> 01:08:39,144

that is on our GitHub.

1163

01:08:39,633 --> 01:08:42,182

An exact carbon copy of what we have.

1164

01:08:42,471 --> 01:08:45,290

So, the first step, obviously...

1165

01:08:45,678 --> 01:08:50,666

as I said before, everything is run through Docker,

1166

01:08:51,072 --> 01:08:52,073

through containers.

1167

01:08:52,173 --> 01:08:58,389

So, the first step, and going with this diagram,

1168

01:08:58,834 --> 01:09:01,763

the first step is that I'm going to create a database

1169

01:09:02,082 --> 01:09:04,541

and the brokers, the post office over here,

1170

01:09:04,641 --> 01:09:09,770

which basically are not optional.

1171

01:09:09,870 --> 01:09:12,808

It's embedded but it doesn't mean that

1172

01:09:12,908 --> 01:09:14,014

you have to use this one.

1173

01:09:14,114 --> 01:09:18,013

Maybe you want to use your own database

1174

01:09:18,211 --> 01:09:22,504

or a more bespoke data manager you have over there

1175

01:09:22,604 --> 01:09:25,363

instead of creating a temporary one like me.

1176

01:09:25,751 --> 01:09:29,447

And I'll say at the end how you can do that.

1177

01:09:29,796 --> 01:09:32,801

But basically you need to change the configuration of

1178

01:09:32,901 --> 01:09:36,319

where the database is but that's not often of concern.

1179

01:09:36,742 --> 01:09:40,127

So, the first step is obviously to create the broker

1180

01:09:40,227 --> 01:09:42,464

and the database.

1181

01:09:42,576 --> 01:09:44,827

And you do that very easily by

1182

01:09:45,127 --> 01:09:49,415

issuing this easy command.

1183

01:09:49,515 --> 01:09:50,863

So, Docker compose.

1184

01:09:50,963 --> 01:09:52,395

We have the compose files

1185

01:09:52,495 --> 01:09:55,384

which basically contain all the instructions for

1186

01:09:55,484 --> 01:09:57,453

what this service should be.

1187

01:09:57,601 --> 01:10:02,709

And up or generate these two services.

1188

01:10:02,809 --> 01:10:06,605

These two services are the database and the rabbit,

1189

01:10:06,705 --> 01:10:07,783

which is the broker.

1190

01:10:07,883 --> 01:10:09,071

And we press enter.

1191

01:10:09,579 --> 01:10:10,977

We are creating it.

1192

01:10:11,077 --> 01:10:14,095

The database, it's made.

1193

01:10:14,195 --> 01:10:17,054

It creates a spatially enabled database.

1194

01:10:17,574 --> 01:10:21,243

And then you can also see the rabbit here,

1195

01:10:23,357 --> 01:10:25,566

the broker that has also been activated.

1196

01:10:25,876 --> 01:10:27,225

After a little while,

1197

01:10:28,849 --> 01:10:32,278

after a little while, it will come down - these are just the logs.

1198

01:10:32,536 --> 01:10:36,854

And that is when your code, database, everything is up and running.

1199

01:10:38,153 --> 01:10:40,512

I'm opening a new tab

1200

01:10:40,612 --> 01:10:43,556

and I'm, cd-ing to the new folders.

1201

01:10:43,656 --> 01:10:47,334

cd src. cd Airbnb; test.

1202

01:10:48,264 --> 01:10:51,504

And if we go to the database here,

1203

01:10:53,846 --> 01:11:00,012

the rabbit, no, DBeaver a database client

1204

01:11:00,112 --> 01:11:03,280

that was able to connect to our local

1205

01:11:03,380 --> 01:11:06,968

in the port 5432 in the postgres database.

1206

01:11:07,262 --> 01:11:10,299

And, basically, it's empty.

1207

01:11:10,399 --> 01:11:11,878

It doesn't have any tables.

1208

01:11:12,317 --> 01:11:14,167

So, to remedy that,

1209

01:11:14,685 --> 01:11:16,373

our application needs a number of tables

1210

01:11:16,473 --> 01:11:18,617

that it's going to store the data inside.

1211

01:11:18,946 --> 01:11:20,831

To do that,

1212

01:11:20,931 --> 01:11:23,945

we need to do the next step.

1213

01:11:24,045 --> 01:11:27,930

It's... I'm going to do it here.

1214

01:11:28,937 --> 01:11:32,016

So, let's create the tables.

1215

01:11:34,165 --> 01:11:35,533

To create the tables

1216

01:11:36,591 --> 01:11:38,365

by issuing this command.

1217

01:11:38,500 --> 01:11:39,931

Basically, what it does, again,

1218

01:11:40,031 --> 01:11:44,459

it's issuing; telling Docker

1219

01:11:44,627 --> 01:11:46,637

to run an instance of the worker

1220

01:11:46,737 --> 01:11:50,399

and through that instance to run the migrate command

1221

01:11:50,499 --> 01:11:53,517

that will migrate the state of the database

1222

01:11:53,617 --> 01:11:54,919

from the current state

1223

01:11:55,019 --> 01:11:59,910

to the final one that has all the tables,

1224

01:12:00,141 --> 01:12:02,232

all the end tables.

1225

01:12:02,594 --> 01:12:03,655

Excuse me.

1226

01:12:04,056 --> 01:12:06,087

So, here we're getting... Okay, okay.

1227

01:12:06,408 --> 01:12:10,689

That's basically where we're creating the tables that we need

1228

01:12:10,789 --> 01:12:12,338

for our operation of this.

1229

01:12:12,438 --> 01:12:15,552

If we press this one, we'll see all the tables are here

1230

01:12:15,652 --> 01:12:17,718

All of these tables, they are serving a purpose.

1231

01:12:18,018 --> 01:12:20,134

But we will explore a little bit.

1232

01:12:20,322 --> 01:12:22,622

But, basically, now we have everything.

1233

01:12:22,722 --> 01:12:24,213

We have a fully functioning database

1234

01:12:24,313 --> 01:12:27,039

with all of the tables that we are supposed to have.

1235

01:12:27,179 --> 01:12:29,407

And more importantly, these tables are,

1236

01:12:30,755 --> 01:12:32,971

four of them, they are spatially enabled.

1237

01:12:33,108 --> 01:12:35,267

Which... Again, apologies.

1238

01:12:35,726 --> 01:12:38,371

I need to figure out how to...

1239

01:12:40,054 --> 01:12:42,552

Okay. And to close these ones.

1240

01:12:42,652 --> 01:12:44,900

More importantly, it's that this database,

1241

01:12:45,118 --> 01:12:46,738

is spatially enabled, as I was saying.

1242

01:12:47,015 --> 01:12:50,974

So, you can connect with it with your favourite GIS client.

1243

01:12:51,672 --> 01:12:55,521

And we'll give it a second here

1244

01:12:55,621 --> 01:12:57,279

for QGIS to load.

1245

01:13:02,933 --> 01:13:04,293

In the meantime...

1246

01:13:11,080 --> 01:13:13,568

So, this is the world over here.

1247

01:13:14,157 --> 01:13:17,286

One of the things that I have, that is vital over here is

1248

01:13:17,386 --> 01:13:18,634

that the...

1249

01:13:19,354 --> 01:13:21,582

Yeah, let's go and share them together.

1250

01:13:22,841 --> 01:13:26,543

Here the ....

1251

01:13:28,400 --> 01:13:32,758

So, the client correctly recognised that

1252

01:13:33,616 --> 01:13:36,514

it has four spatial tables here.

1253

01:13:52,939 --> 01:13:53,998

This one.

1254

01:13:55,293 --> 01:13:58,872

So, one of the things that we want is

1255

01:14:00,351 --> 01:14:03,959

the application that is built basically allows you anywhere to

1256

01:14:04,438 --> 01:14:06,737

define an area of interest.

1257

01:14:07,656 --> 01:14:09,613

If you remove this one.

1258

01:14:09,713 --> 01:14:11,835

If you can define the area of interest

1259

01:14:11,935 --> 01:14:14,784

by opening QGIS

1260

01:14:14,884 --> 01:14:19,928

and then loading the AOI shape table.

1261

01:14:20,826 --> 01:14:24,834

And you can draw a polygon

1262

01:14:24,934 --> 01:14:28,042

where you say, okay, I'm interested in this location

1263

01:14:28,142 --> 01:14:32,010

for this demonstration, I will go to Jersey.

1264

01:14:32,249 --> 01:14:35,130

It's a small island with very limited listings,

1265

01:14:35,230 --> 01:14:36,879

it's very suitable for our demonstration.

1266

01:14:36,979 --> 01:14:38,857

And basically what you do is

1267

01:14:39,297 --> 01:14:42,136

you are creating a polygon over here

1268

01:14:42,236 --> 01:14:44,814

and you are adding the items that you need.

1269

01:14:46,456 --> 01:14:48,747

Jersey and time stamps.

1270

01:14:48,932 --> 01:14:51,726

You're telling it when you want this GIS.

1271

01:14:51,826 --> 01:14:54,095

This one should be automatically put but it's not.

1272

01:14:54,195 --> 01:14:57,993

And then you are

1273

01:14:58,491 --> 01:15:00,708

telling it how these polygons should be used.

1274

01:15:00,808 --> 01:15:03,026

So, are you going to use them to call up the calendars?

1275

01:15:03,100 --> 01:15:06,179

Are you going to use the data to collect the listing details?

1276

01:15:06,279 --> 01:15:08,997

Are you going to collect any reviews?

1277

01:15:09,097 --> 01:15:11,994

Are you going to use it to discover new listings?

1278

01:15:12,462 --> 01:15:15,013

And are you going to use it to collect bookings?

1279

01:15:15,113 --> 01:15:16,741

I'm going to use it for that one.

1280

01:15:16,841 --> 01:15:18,640

So, basically, you can do it here.

1281

01:15:18,858 --> 01:15:23,107

And this is the AOI number one.

1282

01:15:23,207 --> 01:15:24,955

We click here. We need this number.

1283

01:15:25,263 --> 01:15:27,582

Cannot, no.

1284

01:15:27,731 --> 01:15:30,089

But one of the things that I wanted to tell you is that

1285

01:15:30,254 --> 01:15:33,288

as the application is made,

1286

01:15:34,405 --> 01:15:36,253

that could work globally,

1287

01:15:37,018 --> 01:15:40,923

the one thing you need is to define which areas are

1288

01:15:41,023 --> 01:15:42,910

land -valid- and what are not.

1289

01:15:43,010 --> 01:15:48,655

So, basically, you need a mask of the land areas

1290

01:15:49,087 --> 01:15:53,406

and what will filter any kind of requests

1291

01:15:53,506 --> 01:15:55,491

that will not happen in the sea.

1292

01:15:55,731 --> 01:15:58,529

So, you're going to be saving some bandwidth.

1293

01:15:58,629 --> 01:16:02,207

You're going to be saving a lot of ...

1294

01:16:02,646 --> 01:16:04,475

And to load the mask...

1295

01:16:04,763 --> 01:16:06,271

So, again, this one.

1296

01:16:06,371 --> 01:16:07,695

To load the mask,

1297

01:16:09,113 --> 01:16:14,139

what we are doing is we are downloading the data from GADM.

1298

01:16:14,460 --> 01:16:15,966

GADM, if you are not familiar with it,

1299

01:16:16,066 --> 01:16:20,274

it's a website from University of California, Berkeley

1300

01:16:20,374 --> 01:16:24,572

where they collect all the world boundaries in certain levels

1301

01:16:24,770 --> 01:16:28,850

and they are creating, and they are giving it to us

1302

01:16:28,950 --> 01:16:32,038

these defined world boundaries

1303

01:16:32,138 --> 01:16:36,776

and that is using that as an input to create the mask.

1304

01:16:36,963 --> 01:16:39,988

And also, seeing as they're kind enough to give data

1305

01:16:40,426 --> 01:16:42,825

to have some information about

1306

01:16:42,925 --> 01:16:45,343

land area you can define for which country.

1307

01:16:47,010 --> 01:16:50,118

If you don't put this parameter, the iso parameter,

1308

01:16:50,218 --> 01:16:52,007

it will do the whole world.

1309

01:16:52,107 --> 01:16:55,245

But if you specifically say that I want it for this country,

1310

01:16:55,604 --> 01:16:57,213

then it will do it for that country.

1311

01:16:58,414 --> 01:17:01,991

If you do it for all the countries, if you do it for the whole world,

1312

01:17:02,091 --> 01:17:03,789

it would take about 10 minutes or so

1313

01:17:03,889 --> 01:17:05,178

to add all the information.

1314

01:17:05,399 --> 01:17:06,765

So, it's not really...

1315

01:17:06,865 --> 01:17:09,503

You could do it but we cannot do it this moment.

1316

01:17:10,078 --> 01:17:11,266

So, just for Jersey,

1317

01:17:11,366 --> 01:17:14,766

I want to add, again, a similar command docker compose,

1318

01:17:15,035 --> 01:17:18,612

and we're going to log these two definition files.

1319

01:17:18,948 --> 01:17:21,293

And I'm going to run an instance for the worker

1320

01:17:21,393 --> 01:17:24,602

and they are going to run the command 'load mask'.

1321

01:17:24,702 --> 01:17:28,921

And I'm going to tell it to load only Jersey.

1322

01:17:29,153 --> 01:17:32,062

So, if I press the enter.

1323

01:17:33,653 --> 01:17:35,691

It's downloading the file.

1324

01:17:37,755 --> 01:17:40,583

Hopefully, thankfully we have a good internet connection.

1325

01:17:41,150 --> 01:17:43,488

And that should only take a couple of minutes.

1326

01:17:44,977 --> 01:17:46,524

Yeah, so, then today...

1327

01:17:48,756 --> 01:17:49,865

Oh, come on.

1328

01:17:53,064 --> 01:17:54,342

Okay.

1329

01:17:58,657 --> 01:18:00,357

Today's internet is not very good.

1330

01:18:02,578 --> 01:18:03,885

It's starting to wake up.

1331

01:18:04,766 --> 01:18:09,203

The file, it's about one gigabyte in size.

1332

01:18:10,606 --> 01:18:12,763

It will download the file

1333

01:18:12,863 --> 01:18:19,021

and then it will import all of the boundaries for the country we said,

1334

01:18:19,121 --> 01:18:22,118

and these boundaries will be used

1335

01:18:22,218 --> 01:18:25,675

as a mask.

1336

01:18:25,775 --> 01:18:27,322

It's quite important

1337

01:18:29,612 --> 01:18:33,849

because otherwise it will just say it cannot understand anything over there

1338

01:18:33,949 --> 01:18:41,357

and will filter out the areas that you're trying to harvest.

1339

01:18:42,663 --> 01:18:43,970

And we've finished all that.

1340

01:18:47,693 --> 01:18:51,107

So, here, we downloaded the file

1341

01:18:51,207 --> 01:18:53,343

and we extracted Jersey

1342

01:18:53,900 --> 01:18:59,873

and we can see it here as an entry in our world shapefile file.

1343

01:19:01,662 --> 01:19:02,711

Excuse me.

1344

01:19:02,861 --> 01:19:06,816

So, that knows that Jersey over here is a valid area

1345

01:19:06,916 --> 01:19:09,070

so whatever you make over here,

1346

01:19:09,170 --> 01:19:11,708

it's going to be processed.

1347

01:19:13,683 --> 01:19:16,271

Now, let's change this a little bit.

1348

01:19:16,371 --> 01:19:19,009

The display.

1349

01:19:19,129 --> 01:19:20,418

Symbology.

1350

01:19:20,518 --> 01:19:25,356

Make the fillings transparent

1351

01:19:25,456 --> 01:19:27,754

so we can see what's happening here.

1352

01:19:27,935 --> 01:19:30,165

So, the next step is...

1353

01:19:31,534 --> 01:19:33,139

Let's go back to ...

1354

01:19:36,797 --> 01:19:38,880

Let's go back to the presentation for a little bit.

1355

01:19:39,808 --> 01:19:42,473

So, we defined an area over here.

1356

01:19:42,573 --> 01:19:45,430

So, one of the things that is not...

1357

01:19:45,530 --> 01:19:47,950

The next step that is happening is that,

1358

01:19:48,510 --> 01:19:51,197

if you remember before, it's that we have...

1359

01:19:53,884 --> 01:19:56,254

Internally, it's going to generate grids.

1360

01:19:56,473 --> 01:20:00,132

And if you remember from before,

1361

01:20:02,101 --> 01:20:04,730

one of the things that we added,

1362

01:20:05,159 --> 01:20:09,447

it's the ability to talk to Airbnb through the API

1363

01:20:09,547 --> 01:20:11,899

using coordinates.

1364

01:20:12,026 --> 01:20:16,415

So, we can tell it to give me all the coordinates.

1365

01:20:16,515 --> 01:20:21,294

Sorry, give me all the homes that are within this bounding box.

1366

01:20:21,623 --> 01:20:26,290

So, the next key thing that we've been doing is that

1367

01:20:26,490 --> 01:20:30,438

we are generating grid boxes or bounding boxes

1368

01:20:30,538 --> 01:20:33,746

inside the boundary of this area of interest.

1369

01:20:34,023 --> 01:20:37,981

But they follow a specific order, specific logic.

1370

01:20:39,147 --> 01:20:41,206

Since we've made for the whole world,

1371

01:20:41,306 --> 01:20:43,673

it's one of the logics that we've been using,

1372

01:20:43,773 --> 01:20:48,269

it's that we're using the quadkeys.

1373

01:20:48,369 --> 01:20:50,047

If you're not familiar with quadkeys,

1374

01:20:50,371 --> 01:20:52,634

it's basically a data structure which,

1375

01:20:53,469 --> 01:20:56,876

if you take the whole world and divide it by four tiles,

1376

01:20:56,976 --> 01:20:58,146

one, two, three, four.

1377

01:20:58,500 --> 01:20:59,500

That's the level one.

1378

01:20:59,600 --> 01:21:02,399

If you take the first one and you divide it again into four,

1379

01:21:03,136 --> 01:21:05,723

you've got 00, 01, 02, 03.

1380

01:21:05,823 --> 01:21:08,926

If you take the first one again and you divide it again by three,

1381

01:21:09,026 --> 01:21:10,082

you've got level three

1382

01:21:10,331 --> 01:21:16,078

and you've got 000, 001, 002, and 003 and etc.

1383

01:21:16,178 --> 01:21:19,258

So, you can go all the way down to level 15

1384

01:21:19,358 --> 01:21:23,077

and where you will have these boxes in this level

1385

01:21:23,177 --> 01:21:27,213

and you can divide it extra,

1386

01:21:27,524 --> 01:21:30,502

you can divide it even more accordingly to your needs.

1387

01:21:31,731 --> 01:21:35,219

So, for... That's basically what we're doing.

1388

01:21:35,319 --> 01:21:38,367

It's the...

1389

01:21:38,811 --> 01:21:40,470

Let's go again with that one.

1390

01:21:41,209 --> 01:21:43,867

So, we're going to look at our initial grid.

1391

01:21:44,256 --> 01:21:46,572

So, the initial grid is basically like

1392

01:21:46,672 --> 01:21:49,693

find me the best grids, quadkeys,

1393

01:21:49,793 --> 01:21:52,431

that they will fit in this bounding box

1394

01:21:52,731 --> 01:21:57,379

and put them into the database.

1395

01:21:58,151 --> 01:22:03,809

So, the code has the logics embedded in that.

1396

01:22:03,909 --> 01:22:06,017

So, basically, you need to kickstart this operation.

1397

01:22:06,117 --> 01:22:12,024

Again, you're telling it around this worker container

1398

01:22:12,048 --> 01:22:16,675

and prepare me a grid for area of interest one.

1399

01:22:16,775 --> 01:22:19,794

That's the number one. That's the area we defined before.

1400

01:22:19,994 --> 01:22:25,627

So, if we press the enter, it will do the things.

1401

01:22:25,727 --> 01:22:29,216

It gives me some of the initial grids and...

1402

01:22:34,633 --> 01:22:39,332

And if we do that there, we will see that initially we have

1403

01:22:41,200 --> 01:22:43,147

only these two grids.

1404

01:22:43,808 --> 01:22:46,854

We'll change this to green.

1405

01:22:47,478 --> 01:22:48,788

Green looks okay.

1406

01:22:48,847 --> 01:22:52,259

So, these two grids could be used to

1407

01:22:52,359 --> 01:22:57,170

identify where Airbnb give all the listings for that.

1408

01:22:57,519 --> 01:22:59,946

But what seems...

1409

01:23:00,046 --> 01:23:01,642

We do have a wee problem.

1410

01:23:01,742 --> 01:23:02,999

Like we talked about before,

1411

01:23:03,099 --> 01:23:04,946

Airbnb is only giving you back

1412

01:23:05,143 --> 01:23:06,689

The listings, not the listings.

1413

01:23:06,789 --> 01:23:10,597

The results are limited to the 300 top results.

1414

01:23:10,697 --> 01:23:13,764

So, you don't have any guarantees over here that...

1415

01:23:14,328 --> 01:23:17,247

These listings, this bounding box will only have

1416

01:23:18,232 --> 01:23:19,338

less than 300

1417

01:23:19,438 --> 01:23:23,333

so it's going to be a little bit more abstract.

1418

01:23:24,869 --> 01:23:28,097

We are not certain that getting everything that we need to get from there.

1419

01:23:28,197 --> 01:23:30,737

So, one of the things we need to do is to divide.

1420

01:23:31,396 --> 01:23:32,875

Empirically, I've found that

1421

01:23:35,052 --> 01:23:37,140

the search results are coming back faster

1422

01:23:37,240 --> 01:23:40,899

if you get like around 50 results, 50 listings.

1423

01:23:40,999 --> 01:23:44,358

And the code over here,

1424

01:23:44,458 --> 01:23:50,414

what it does is it asks Airbnb

1425

01:23:50,514 --> 01:23:54,028

how many and only how many listings we have within this bounding box.

1426

01:23:54,128 --> 01:23:55,556

And we've had, like, more than 50.

1427

01:23:55,717 --> 01:23:56,966

Empirical value.

1428

01:23:57,324 --> 01:23:59,014

Divided to its children.

1429

01:23:59,114 --> 01:24:01,562

If you have less than 50, it's fine.

1430

01:24:01,662 --> 01:24:05,140

You leave it like that. And we can do it now.

1431

01:24:08,221 --> 01:24:10,409

And hopefully this finishes.

1432

01:24:10,509 --> 01:24:12,924

Basically, it's a way of

1433

01:24:13,024 --> 01:24:15,901

dividing or breaking down the queries

1434

01:24:16,001 --> 01:24:17,230

into more manageable...

1435

01:24:17,647 --> 01:24:20,505

Not the queries, the target where we're going to run the queries,

1436

01:24:20,605 --> 01:24:23,514

bring it to a level that is more manageable.

1437

01:24:23,723 --> 01:24:27,562

So, this could also work in very, very big areas.

1438

01:24:27,867 --> 01:24:31,664

So, I do not know if you can see this one but it's really annoying.

1439

01:24:33,015 --> 01:24:35,172

So, for example,

1440

01:24:35,272 --> 01:24:40,479

you could do something in much bigger areas than Jersey.

1441

01:24:40,579 --> 01:24:42,029

This is London.

1442

01:24:43,335 --> 01:24:44,502

That I did like a while back.

1443

01:24:44,602 --> 01:24:47,591

So, it's grid, imagine...

1444

01:24:47,799 --> 01:24:50,269

Not imagine. It's grid, at the time I did it, had

1445

01:24:50,369 --> 01:24:53,797

at least, no, at the most 50 listings inside.

1446

01:24:54,784 --> 01:24:58,155

So, the interesting thing about this data structure is that

1447

01:24:58,255 --> 01:25:00,519

you could use all of these levels as they are

1448

01:25:00,619 --> 01:25:04,305

and you could cover the whole world without having any overlaps.

1449

01:25:04,674 --> 01:25:07,140

This is quite important and interesting.

1450

01:25:07,227 --> 01:25:11,124

So, expand, imagine that you are not looking at London,

1451

01:25:11,224 --> 01:25:12,761

you're not looking at Jersey,

1452

01:25:12,861 --> 01:25:15,709

you're not looking at London, you're looking at, like, Great Britain,

1453

01:25:15,809 --> 01:25:18,374

you're looking at the whole of Europe, world, whatever.

1454

01:25:19,154 --> 01:25:22,013

The approach will be the same.

1455

01:25:23,660 --> 01:25:25,609

If we go back to that.

1456

01:25:26,050 --> 01:25:28,496

And if we go to this one.

1457

01:25:28,596 --> 01:25:31,084

So, we have sent this one.

1458

01:25:31,184 --> 01:25:33,492

Hopefully it's finished. If we move this a little bit.

1459

01:25:34,433 --> 01:25:36,570

It did. No, it didn't do it.

1460

01:25:37,040 --> 01:25:38,259

Oh, yeah. For this command,

1461

01:25:38,359 --> 01:25:40,265

I need to fire up the worker.

1462

01:25:40,526 --> 01:25:42,004

Basically, a worker is...

1463

01:25:43,048 --> 01:25:48,868

If you remember before, so far, I've been doing the commands in line.

1464

01:25:49,335 --> 01:25:53,005

For this one, I sent the work order to the broker

1465

01:25:53,105 --> 01:25:55,846

and it's waiting for a worker to pick it up

1466

01:25:55,946 --> 01:25:58,742

and actually do the same thing.

1467

01:25:59,863 --> 01:26:02,140

To the worker.

1468

01:26:06,865 --> 01:26:09,394

And now... Yeah.

1469

01:26:09,773 --> 01:26:10,780

This one.

1470

01:26:13,499 --> 01:26:17,349

Yeah, cd src, cd Airbnb.

1471

01:26:21,007 --> 01:26:23,130

So, here the worker... It's interesting.

1472

01:26:24,599 --> 01:26:26,477

I opened the worker

1473

01:26:27,275 --> 01:26:31,334

and the worker understood,

1474

01:26:32,642 --> 01:26:34,340

picked out a connection to broker,

1475

01:26:34,440 --> 01:26:38,469

and the broker gave it this function,

1476

01:26:39,038 --> 01:26:40,567

these tasks to do.

1477

01:26:40,667 --> 01:26:43,456

Where is it? Gave them this ID.

1478

01:26:43,556 --> 01:26:47,068

And it started doing it.

1479

01:26:47,168 --> 01:26:49,369

One of the tasks.

1480

01:26:49,767 --> 01:26:52,495

Let's give them like three tasks, as many grids as there are.

1481

01:26:52,595 --> 01:26:53,924

So, for each grid up there,

1482

01:26:54,024 --> 01:26:57,172

the initial grids, sends how many listings there are.

1483

01:26:58,134 --> 01:27:02,367

So, for this listing with this queue.

1484

01:27:02,703 --> 01:27:06,101

And this quadkey ID has 108

1485

01:27:06,201 --> 01:27:07,381

so it's more than 50.

1486

01:27:08,180 --> 01:27:13,828

Divide it by four and sends for these four new ones,

1487

01:27:13,928 --> 01:27:15,988

if there are more than 50 listings.

1488

01:27:16,088 --> 01:27:18,853

If yes, then divide and move on.

1489

01:27:18,953 --> 01:27:21,168

So, if I move a little bit, you can see that, okay,

1490

01:27:21,268 --> 01:27:24,756

all of these listing, all of these grids, they have less than 50.

1491

01:27:25,692 --> 01:27:31,815

And if we wait for a little bit.

1492

01:27:35,104 --> 01:27:38,206

So, everything...

1493

01:27:40,163 --> 01:27:42,406

So, if we open this database,

1494

01:27:42,923 --> 01:27:44,584

I think we don't see that...

1495

01:27:44,684 --> 01:27:46,866

Okay, these two grids are still pending.

1496

01:27:47,697 --> 01:27:50,107

So, for all the ones that we already have,

1497

01:27:50,207 --> 01:27:52,978

we have an estimated number of how many listings there are.

1498

01:27:53,078 --> 01:27:56,420

I'm saying estimate because there is a way for Airbnb to ask it

1499

01:27:56,520 --> 01:27:59,513

and tell you, okay, give me back all the metadata only,

1500

01:27:59,613 --> 01:28:00,734

not the actual data.

1501

01:28:00,834 --> 01:28:03,787

Metadata means how many listings there are in your query

1502

01:28:04,262 --> 01:28:06,801

and when you did the query with the engine,

1503

01:28:06,901 --> 01:28:10,152

which is not randomly querying, etc.

1504

01:28:10,818 --> 01:28:12,964

But not the actual information about that.

1505

01:28:13,064 --> 01:28:14,780

That saves some bandwidth.

1506

01:28:15,599 --> 01:28:19,194

So, after we did all the grids.

1507

01:28:21,489 --> 01:28:23,994

So, now that we've done all the grids.

1508

01:28:28,565 --> 01:28:29,953

Now that we've done all the grids,

1509

01:28:30,053 --> 01:28:33,550

we are certain that these ones are,

1510

01:28:34,561 --> 01:28:37,061

it's clear that they have less than 50 ones

1511

01:28:37,161 --> 01:28:39,508

and now we are ready...

1512

01:28:39,608 --> 01:28:42,361

We have an optimised grid network

1513

01:28:42,461 --> 01:28:44,697

which, again, contains less than 50.

1514

01:28:44,797 --> 01:28:49,492

So, the next step is to find, not find, to tell

1515

01:28:53,282 --> 01:28:57,215

the application to find, discover all the listings that we have.

1516

01:28:57,619 --> 01:28:59,025

Again, we've sent a command.

1517

01:28:59,125 --> 01:29:03,507

We are telling it using Docker compose.

1518

01:29:03,607 --> 01:29:06,190

Sorry, use the worker container,

1519

01:29:06,290 --> 01:29:08,399

send this task to the workers,

1520

01:29:08,499 --> 01:29:10,434

and tell it to discover all the listings.

1521

01:29:10,534 --> 01:29:11,993

What it's going to do for each grid,

1522

01:29:12,093 --> 01:29:13,893

it's going to run the query

1523

01:29:14,692 --> 01:29:17,031

for this bounding box, give me all the listings,

1524

01:29:17,131 --> 01:29:19,159

and then it will extract the information

1525

01:29:19,357 --> 01:29:24,001

and it will do some processing and add them to the database.

1526

01:29:24,999 --> 01:29:27,411

So, we are issuing the command.

1527

01:29:27,511 --> 01:29:30,720

The job is submitted. The ID of the job is this one.

1528

01:29:30,879 --> 01:29:34,597

The worker picks it up and it tells us, okay.

1529

01:29:37,533 --> 01:29:40,993

This is a problem I am having in this virtual machine

1530

01:29:41,592 --> 01:29:43,122

Docker ps.

1531

01:29:44,111 --> 01:29:46,419

Docker logs.

1532

01:29:48,691 --> 01:29:50,651

Okay. There we are.

1533

01:29:50,751 --> 01:29:54,050

So, what has happened over here,

1534

01:29:54,150 --> 01:29:55,520

for all the grids, it says,

1535

01:29:55,620 --> 01:29:58,345

okay, find me all the listings in that grid.

1536

01:29:58,445 --> 01:30:00,524

And here are the commands coming back.

1537

01:30:00,624 --> 01:30:03,993

And for this grid, I found 22.

1538

01:30:04,441 --> 01:30:07,405

For this grid, I found 49, and etc.

1539

01:30:07,835 --> 01:30:10,652

For this one, I have some duplicates,

1540

01:30:10,752 --> 01:30:14,031

I had already seen it zero minutes ago.

1541

01:30:14,641 --> 01:30:17,320

And it keeps adding.

1542

01:30:17,771 --> 01:30:19,840

So, if we do this.

1543

01:30:20,044 --> 01:30:22,123

If we enable the layer with the listings,

1544

01:30:23,137 --> 01:30:25,315

we can see all the listings here,

1545

01:30:25,415 --> 01:30:27,757

which is very interesting.

1546

01:30:28,822 --> 01:30:32,349

Initially, immediately we start to have some kind of sense of

1547

01:30:32,449 --> 01:30:34,558

information from Airbnb what's happening in Jersey.

1548

01:30:34,658 --> 01:30:37,926

In Jersey, most of the listings are over here.

1549

01:30:38,286 --> 01:30:40,626

They are within this area.

1550

01:30:41,925 --> 01:30:44,934

And you can find this route.

1551

01:30:46,531 --> 01:30:47,720

Left and right.

1552

01:30:47,920 --> 01:30:51,397

Now, there's questions for post process,

1553

01:30:51,497 --> 01:30:53,025

how you're going to use the data.

1554

01:30:55,746 --> 01:30:56,753

Pardon me.

1555

01:30:57,968 --> 01:31:02,816

So, this is step one.

1556

01:31:02,916 --> 01:31:05,116

Now we have a feeling of how many listings

1557

01:31:05,216 --> 01:31:06,285

and where the listings are

1558

01:31:06,385 --> 01:31:08,943

and, most importantly, we have the listing IDs.

1559

01:31:09,114 --> 01:31:15,271

And if you go to the code which says,

1560

01:31:15,517 --> 01:31:18,416

demonstrates, describes the API,

1561

01:31:18,516 --> 01:31:20,025

how you're going to develop the API,

1562

01:31:20,125 --> 01:31:24,792

most of the endpoints of the API designed that

1563

01:31:25,000 --> 01:31:27,071

can access the parameter, the listing IDs.

1564

01:31:27,171 --> 01:31:29,339

So, you can ask the API,

1565

01:31:29,439 --> 01:31:31,317

okay, give me the listing details for

1566

01:31:33,577 --> 01:31:36,836

the listing with the listing ID XZ

1567

01:31:36,936 --> 01:31:42,605

or, in our case, 21019036,

1568

01:31:43,351 --> 01:31:46,806

and that will give you

1569

01:31:46,906 --> 01:31:51,213

the listing details, the information about

1570

01:31:51,313 --> 01:31:52,971

what this listing contains.

1571

01:31:53,361 --> 01:31:56,598

And we can do that as well.

1572

01:31:57,948 --> 01:32:02,218

We define an area of interest that we're interested in.

1573

01:32:02,900 --> 01:32:06,159

So, we can tell it, okay, give me all the listing details for

1574

01:32:06,527 --> 01:32:09,235

all the areas data, all the area of interest data

1575

01:32:09,335 --> 01:32:10,952

we have enabled it to look for

1576

01:32:11,800 --> 01:32:15,867

that we're interested in getting all the listing details about.

1577

01:32:16,367 --> 01:32:18,494

Again, we're issuing the command.

1578

01:32:18,594 --> 01:32:23,359

So, the framework over here

1579

01:32:23,459 --> 01:32:25,774

created a number of tasks.

1580

01:32:27,463 --> 01:32:30,240

Here are all the tasks for the listings. We have a number of listings.

1581

01:32:30,340 --> 01:32:31,769

I don't know how many we have.

1582

01:32:34,519 --> 01:32:35,977

Wait a second, let's count.

1583

01:32:36,757 --> 01:32:37,765

So, if you just count.

1584

01:32:37,865 --> 01:32:43,124

So, we have 200 listings in all of Jersey so far.

1585

01:32:43,635 --> 01:32:45,794

So, we make 200 requests

1586

01:32:45,894 --> 01:32:49,572

and these requests for each listing

1587

01:32:49,810 --> 01:32:52,299

go to Airbnb and come back with

1588

01:32:52,399 --> 01:32:56,698

the information about what's happening with each listing.

1589

01:32:58,155 --> 01:33:02,093

Now, if we see here what's coming back.

1590

01:33:02,193 --> 01:33:03,999

And now here is the interesting thing.

1591

01:33:04,494 --> 01:33:06,323

I do know about the listing details

1592

01:33:06,423 --> 01:33:08,349

but what is actually in the...

1593

01:33:09,093 --> 01:33:10,931

So, in technical terms,

1594

01:33:11,247 --> 01:33:14,226

what I get back is response object.

1595

01:33:14,777 --> 01:33:18,787

I am requesting information and the API is responding with

1596

01:33:18,984 --> 01:33:20,320

some information.

1597

01:33:20,501 --> 01:33:23,579

And I'm storing those responses in the database.

1598

01:33:23,986 --> 01:33:26,085

And this table, no.

1599

01:33:27,316 --> 01:33:28,486

It's the big one.

1600

01:33:30,957 --> 01:33:32,003

Response.

1601

01:33:32,103 --> 01:33:37,981

So, I am storing these responses back into this table,

1602

01:33:39,046 --> 01:33:44,803

which has all the interesting things that you guys might be interested in.

1603

01:33:45,235 --> 01:33:47,004

And Sublime.

1604

01:33:47,104 --> 01:33:48,322

So, if we take the response.

1605

01:33:50,087 --> 01:33:52,376

I'm saving it as a JSON, as a document,

1606

01:33:52,476 --> 01:33:56,778

and so if you take response over here and you're going to copy the text

1607

01:33:56,878 --> 01:33:58,442

that has been JSON and put it,

1608

01:33:59,777 --> 01:34:03,824

and I put it here, it's a very, very big string.

1609

01:34:03,924 --> 01:34:05,854

So, that's why I was saying that we need

1610

01:34:08,229 --> 01:34:11,867

a fancy text editor so that we can format it regularly.

1611

01:34:13,371 --> 01:34:16,771

It's a JSON grid format.

1612

01:34:18,292 --> 01:34:20,058

Go back to it.

1613

01:34:21,119 --> 01:34:23,105

Here we have interesting things.

1614

01:34:23,244 --> 01:34:26,615

Here we have the details about this listing ID.

1615

01:34:27,443 --> 01:34:30,629

The listing IDs.

This one has these coordinates,

1616

01:34:30,729 --> 01:34:31,915

the longitude and latitude.

1617

01:34:32,015 --> 01:34:34,917

It's in the city of Jersey in the state of that.

1618

01:34:35,017 --> 01:34:38,246

And these are all the photographs that they have.

1619

01:34:38,346 --> 01:34:39,902

All this data is used, again,

1620

01:34:40,002 --> 01:34:43,364

for Airbnb's website to hydrate their templates

1621

01:34:43,464 --> 01:34:45,850

to create the extra features that you see,

1622

01:34:46,296 --> 01:34:49,417

to create a responsive website.

1623

01:34:51,034 --> 01:34:54,611

Other information that they have.

1624

01:34:55,336 --> 01:34:58,532

The host allows you to book it for 21 nights

1625

01:34:58,632 --> 01:35:04,589

and they are only allowed to book it for the minimum amount of two nights.

1626

01:35:05,341 --> 01:35:08,515

Who the host is, that's their ID, Sarah.

1627

01:35:09,142 --> 01:35:14,406

And that's the URL for a picture that she uploaded.

1628

01:35:14,506 --> 01:35:19,412

And what kind of bedroom label it has.

1629

01:35:19,512 --> 01:35:23,598

A three bedroom place and has these quantities.

1630

01:35:24,658 --> 01:35:25,976

Allows pets, no.

1631

01:35:26,076 --> 01:35:28,046

Allows events, no.

1632

01:35:28,796 --> 01:35:32,506

All the, called a dictionary of data

1633

01:35:32,606 --> 01:35:38,225

with the information that keeps evolving as things are evolving.

1634

01:35:38,664 --> 01:35:40,533

I'm not sure if we have it with COVID.

1635

01:35:40,783 --> 01:35:41,912

No, they haven't done it.

1636

01:35:42,012 --> 01:35:44,400

I've seen some places that...

1637

01:35:47,928 --> 01:35:51,685

I've seen some places with COVID related information

1638

01:35:51,785 --> 01:35:57,054

that you can harvest and service to do some analysis later on.

1639

01:35:58,762 --> 01:36:00,273

But anyhow, yeah.

1640

01:36:01,910 --> 01:36:03,227

I've just pressed F5.

1641

01:36:03,327 --> 01:36:07,685

So, the same ID, the same thinking is happening.

1642

01:36:08,513 --> 01:36:10,042

For these 200...

1643

01:36:10,961 --> 01:36:16,620

I have one worker that collects information, the listing details about

1644

01:36:16,720 --> 01:36:18,869

these 200 listings in Jersey.

1645

01:36:19,088 --> 01:36:21,875

And they took some time, over here, to complete.

1646

01:36:21,975 --> 01:36:26,254

I mean, you can see this request took two seconds,

1647

01:36:26,354 --> 01:36:30,902

this one took another two seconds, that one took three seconds, and etc.

1648

01:36:31,390 --> 01:36:33,392

So, if you collect 200,

1649

01:36:33,492 --> 01:36:35,525

it would take many, many minutes.

1650

01:36:35,650 --> 01:36:39,040

If you had 1000, it would take many, many, hours.

1651

01:36:39,140 --> 01:36:42,964

So, one of the things that is interesting is that

1652

01:36:43,064 --> 01:36:46,150

using this approach, you can see the environment

1653

01:36:46,250 --> 01:36:49,009

because there are more workers that can work in parallel,

1654

01:36:52,390 --> 01:36:56,923

make all of the application requests concurrently.

1655

01:36:57,233 --> 01:36:59,485

And to do that,

1656

01:36:59,585 --> 01:37:00,626

let's call again.

1657

01:37:01,377 --> 01:37:05,600

What we would do to collect more...

1658

01:37:06,898 --> 01:37:08,217

Scale.

1659

01:37:08,497 --> 01:37:12,997

So, again, I'm telling it instead of having only one worker,

1660

01:37:13,097 --> 01:37:18,276

I want to have two workers or let's get three workers or ten.

1661

01:37:21,974 --> 01:37:23,021

Scale.

1662

01:37:23,971 --> 01:37:26,713

No, scale please. Yes, thank you.

1663

01:37:27,172 --> 01:37:28,721

And give me three workers

1664

01:37:28,870 --> 01:37:31,029

so we can run it a little bit faster.

1665

01:37:32,336 --> 01:37:34,984

So, basically, I'm creating over here...

1666

01:37:37,169 --> 01:37:38,950

So, what I'm creating over here is...

1667

01:37:40,697 --> 01:37:41,716

Come on.

1668

01:37:42,917 --> 01:37:48,115

So, I am putting our two workers over here in the ecosystem

1669

01:37:48,215 --> 01:37:50,092

and all of them, both of them,

1670

01:37:50,781 --> 01:37:52,720

now three of them, they connect to the broker

1671

01:37:52,918 --> 01:37:56,934

and they are asking for requests for tasks to run to command.

1672

01:37:57,034 --> 01:37:59,383

And all of them, they are doing it individually

1673

01:38:02,211 --> 01:38:04,669

without interfering with each other.

1674

01:38:04,769 --> 01:38:06,576

There are some other interesting things

1675

01:38:06,676 --> 01:38:09,313

that with this technology, with this architecture.

1676

01:38:09,513 --> 01:38:11,530

If one of these workers fails for whatever reason,

1677

01:38:11,836 --> 01:38:14,735

the task becomes outstanding and the broker knows, okay,

1678

01:38:14,835 --> 01:38:19,323

I should have got a result from worker X

1679

01:38:19,423 --> 01:38:23,079

but I never got it so I assume that he died

1680

01:38:23,179 --> 01:38:26,688

so I'm reissuing the task for another alive worker to do.

1681

01:38:27,455 --> 01:38:29,294

And, yeah.

1682

01:38:31,059 --> 01:38:33,368

Basically, one interesting thing you could do...

1683

01:38:33,847 --> 01:38:36,294

Local host.

1684

01:38:38,241 --> 01:38:41,049

We have a visual...

1685

01:38:41,698 --> 01:38:45,594

Rabbit, the broker, has a web interface that tells us

1686

01:38:45,694 --> 01:38:47,002

what's happening in their queues.

1687

01:38:47,102 --> 01:38:50,160

So, rabbit, carrots.

1688

01:38:50,260 --> 01:38:51,788

All of these. So, here we can see

1689

01:38:51,888 --> 01:38:55,756

it's the outstanding tasks that need to happen.

1690

01:38:56,395 --> 01:38:58,067

They are pending. So, we still have

1691

01:38:58,167 --> 01:39:04,802

another 112 tasks that

1692

01:39:09,894 --> 01:39:11,642

they are still in line

1693

01:39:13,129 --> 01:39:14,908

until that finishes.

1694

01:39:15,105 --> 01:39:18,673

And they are coming back with grids.

1695

01:39:18,763 --> 01:39:20,451

It's been like a long time so...

1696

01:39:21,240 --> 01:39:22,419

Just going to...

1697

01:39:24,656 --> 01:39:26,036

Scale.

1698

01:39:26,136 --> 01:39:28,652

Just put another, make it 10 workers.

1699

01:39:29,612 --> 01:39:30,801

And see it's a bit faster.

1700

01:39:30,901 --> 01:39:32,650

See how...

1701

01:39:36,507 --> 01:39:39,827

So, basically, I'm having 10 concurrent workers

1702

01:39:39,927 --> 01:39:41,829

bombarding Airbnb.

1703

01:39:44,271 --> 01:39:46,848

Bombarding Airbnb and asking details about everything.

1704

01:39:46,948 --> 01:39:49,490

And here, we can see here we've got

1705

01:39:51,871 --> 01:39:54,354

the remaining tasks.

1706

01:39:54,454 --> 01:39:58,057

They complete much, much faster.

1707

01:39:58,157 --> 01:40:01,076

We still have 54 to come back.

1708

01:40:01,694 --> 01:40:02,903

Forty-eight.

1709

01:40:05,906 --> 01:40:07,086

Thirty-eight.

1710

01:40:10,725 --> 01:40:14,595

Just 20.

1711

01:40:17,098 --> 01:40:18,900

Yeah, I wasn't expecting it to take that long.

1712

01:40:19,000 --> 01:40:24,473

Last time I let this scenario, Jersey didn't have that many listings.

1713

01:40:26,651 --> 01:40:28,250

Six.

1714

01:40:30,440 --> 01:40:31,598

And we're done.

1715

01:40:31,698 --> 01:40:33,277

Well, almost.

1716

01:40:36,875 --> 01:40:41,113

And if we open again the responses table over here,

1717

01:40:41,213 --> 01:40:47,348

we could extract all the information en masse of what we want,

1718

01:40:47,448 --> 01:40:50,597

of the listings that we're interested in.

1719

01:40:52,384 --> 01:40:54,033

Obviously, getting the listing details,

1720

01:40:54,133 --> 01:40:58,319

it's just that we ask for the things that we want to get.

1721

01:40:59,364 --> 01:41:01,647

There's particular interest on the calendar.

1722

01:41:01,733 --> 01:41:03,904

So, the same idea applies.

1723

01:41:04,945 --> 01:41:07,874

We have defined, pardon me.

1724

01:41:08,323 --> 01:41:11,839

We have defined our area of interest here,

1725

01:41:11,939 --> 01:41:14,507

that we want to use it to collect all the calendars.

1726

01:41:14,607 --> 01:41:18,755

So, again, what we do is

1727

01:41:19,367 --> 01:41:23,186

issue the relevant commands.

1728

01:41:23,404 --> 01:41:24,551

Now, which one?

1729

01:41:29,800 --> 01:41:31,358

Don't stop everything.

1730

01:41:32,698 --> 01:41:33,866

This is very annoying.

1731

01:41:37,765 --> 01:41:39,074

[inaudible].

1732

01:41:44,484 --> 01:41:46,698

Excuse me, folks, for a second here.

1733

01:41:47,397 --> 01:41:51,035

Things are happening when I press the button at the wrong time.

1734

01:41:53,356 --> 01:41:55,024

I just closed the database.

1735

01:42:15,462 --> 01:42:18,662

Here, we've got the database and the rabbit's running, okay.

1736

01:42:19,362 --> 01:42:22,307

So, what we want to do is

1737

01:42:22,407 --> 01:42:23,535

change worker to one.

1738

01:42:26,726 --> 01:42:27,864

Yeah, we've got it.

1739

01:42:27,964 --> 01:42:29,961

Okay, so let's open a new one.

1740

01:42:30,498 --> 01:42:32,458

cd src.

1741

01:42:32,558 --> 01:42:34,708

cd Airbnb test.

1742

01:42:36,136 --> 01:42:37,454

As I was saying,

1743

01:42:37,824 --> 01:42:39,631

we can collect the calendars as well.

1744

01:42:39,731 --> 01:42:40,938

Another bit of information.

1745

01:42:41,038 --> 01:42:44,254

The calendar, it says for, maybe in context it tells...

1746

01:42:45,400 --> 01:42:47,269

We'll give it a look in a little bit.

1747

01:42:50,557 --> 01:42:52,627

And, again, we apply for that.

1748

01:42:53,007 --> 01:42:54,121

So, we created tasks.

1749

01:42:54,221 --> 01:42:57,705

So, the framework it's creating for each

1750

01:42:59,691 --> 01:43:03,725

its listing task to get the calendar.

1751

01:43:04,962 --> 01:43:07,159

Airbnb is trying to fight with us.

1752

01:43:07,259 --> 01:43:10,087

It's 403, I'm not giving you access to it.

1753

01:43:10,187 --> 01:43:17,035

So, we're rerouting our query,

1754

01:43:17,389 --> 01:43:20,269

response, request through our proxies.

1755

01:43:21,490 --> 01:43:24,447

And we defeated it a little bit.

1756

01:43:24,480 --> 01:43:25,688

Let's say in quotes.

1757

01:43:25,788 --> 01:43:27,867

And we're getting back the results,

1758

01:43:28,165 --> 01:43:31,123

the responses that contain the calendars.

1759

01:43:31,874 --> 01:43:34,164

And we can see them here.

1760

01:43:35,644 --> 01:43:39,366

The response table is a little bit big.

1761

01:43:39,755 --> 01:43:42,423

So, we got also the type.

1762

01:43:42,632 --> 01:43:43,991

So, the code here for

1763

01:43:45,350 --> 01:43:48,138

type equals CAL.

1764

01:43:51,407 --> 01:43:53,796

So, we've got all the information,

1765

01:43:53,944 --> 01:43:56,722

we've got all the responses for the calendar of that listing ID.

1766

01:43:57,744 --> 01:43:59,492

If we copy that,

1767

01:43:59,592 --> 01:44:02,551

let's try to do with this one.

1768

01:44:02,890 --> 01:44:05,358

Again, copy and paste.

1769

01:44:05,579 --> 01:44:09,386

JSON pretty.

1770

01:44:10,088 --> 01:44:13,506

And we've got a nicely documented JSON response

1771

01:44:13,606 --> 01:44:16,125

that has all the information that one might need.

1772

01:44:16,434 --> 01:44:18,861

So, for example, for this listing ID,

1773

01:44:18,961 --> 01:44:20,259

if we close it

1774

01:44:22,046 --> 01:44:24,505

and open it like this.

1775

01:44:26,475 --> 01:44:27,491

It's coming back.

1776

01:44:27,591 --> 01:44:29,750

You see, it's a bit hard to navigate here.

1777

01:44:29,850 --> 01:44:31,499

I don't know how they do it.

1778

01:44:31,648 --> 01:44:34,625

But, basically, what it tells us is that

1779

01:44:34,916 --> 01:44:36,670

the listing ID.

1780

01:44:37,171 --> 01:44:38,314

Interesting.

1781

01:44:39,334 --> 01:44:42,142

So, what it's telling us for this date is

1782

01:44:46,522 --> 01:44:51,999

the price was 64 units of local currency,

1783

01:44:52,099 --> 01:44:54,423

which was Great British Pounds,

1784

01:44:54,523 --> 01:44:56,502

that's what they have in Jersey.

1785

01:44:56,602 --> 01:44:58,863

In different countries, they have different currencies.

1786

01:44:59,131 --> 01:45:03,246

And I'm sure that you can see that this is quite structured,

1787

01:45:03,346 --> 01:45:04,553

you can get the information.

1788

01:45:04,908 --> 01:45:08,141

Although, it's a little bit too much information

1789

01:45:08,241 --> 01:45:13,540

so you might have problems getting value out of it.

1790

01:45:14,349 --> 01:45:18,905

Thankfully, an interesting tip that I can share with you guys,

1791

01:45:19,005 --> 01:45:22,334

it's basically that we could use JSONPath

1792

01:45:22,774 --> 01:45:25,094

to extract the information that you want.

1793

01:45:25,194 --> 01:45:26,851

JSONPath, basically,

1794

01:45:28,159 --> 01:45:31,838

you can give instructions.

1795

01:45:31,838 --> 01:45:33,907

There are libraries that can do that with Python,

1796

01:45:34,007 --> 01:45:36,251

and libraries that you can give instructions to extract

1797

01:45:36,351 --> 01:45:38,270

the information that you want from a document.

1798

01:45:38,381 --> 01:45:40,317

Now, let's say that you want

1799

01:45:41,905 --> 01:45:43,957

to see all the dates that are available.

1800

01:45:44,246 --> 01:45:46,458

There you could run

1801

01:45:48,428 --> 01:45:51,105

the appropriate description,

1802

01:45:51,205 --> 01:45:53,304

not description, the query

1803

01:45:53,784 --> 01:45:57,080

for the library to use that would navigate through the data

1804

01:45:57,180 --> 01:45:58,918

and extract all the pieces that you need.

1805

01:45:59,156 --> 01:46:01,123

For example, if you wanted the date,

1806

01:46:05,581 --> 01:46:09,451

date, okay, you want all the dates,

1807

01:46:09,924 --> 01:46:13,602

you tell it double dots, give me all the days

1808

01:46:13,702 --> 01:46:15,110

and the top one.

1809

01:46:15,867 --> 01:46:18,807

And I want all of them, just them.

1810

01:46:18,907 --> 01:46:22,256

And from that one, I just want the date.

1811

01:46:22,356 --> 01:46:23,804

Give me all the dates on the right.

1812

01:46:24,395 --> 01:46:29,075

And if it's available.

1813

01:46:30,732 --> 01:46:34,007

So, you've got a much easier structure

1814

01:46:34,107 --> 01:46:35,885

that you can work with.

1815

01:46:36,646 --> 01:46:41,995

So, you can divide it by 2 or for this date.

1816

01:46:42,095 --> 01:46:43,702

It's available for these dates.

1817

01:46:43,971 --> 01:46:45,548

Not available, sorry.

1818

01:46:45,648 --> 01:46:47,597

And not available, not available, not available.

1819

01:46:47,766 --> 01:46:49,435

For these dates, it's available, and etc.

1820

01:46:49,535 --> 01:46:51,634

You could do more fun stuff

1821

01:46:52,024 --> 01:46:53,559

to extend the code.

1822

01:46:53,759 --> 01:46:57,337

So, you could go all the way down

1823

01:46:57,437 --> 01:46:59,885

and go to the price node.

1824

01:47:00,963 --> 01:47:01,977

Here it is.

1825

01:47:02,077 --> 01:47:06,226

Here we've got another sub or nested environment.

1826

01:47:06,557 --> 01:47:09,886

And from here, on all of that,

1827

01:47:09,962 --> 01:47:13,290

I want from here, I want the dates

1828

01:47:13,390 --> 01:47:16,229

and I also want the price.

1829

01:47:17,829 --> 01:47:20,457

Dates, price.

1830

01:47:22,591 --> 01:47:23,671

Dates.

1831

01:47:24,839 --> 01:47:26,087

It's local price.

1832

01:47:29,026 --> 01:47:32,256

So, for that date, it was £64.

1833

01:47:32,356 --> 01:47:33,955

On that date, £95.

1834

01:47:34,055 --> 01:47:36,322

That one is £115.

1835

01:47:36,492 --> 01:47:40,723

So, you see how as the summer is progressing,

1836

01:47:41,082 --> 01:47:46,211

the prices are becoming more steep as the summer is finishing.

1837

01:47:46,610 --> 01:47:49,220

The prices are becoming smaller,

1838

01:47:49,852 --> 01:47:50,870

lower.

1839

01:47:50,970 --> 01:47:57,499

And the lowest price is £64 for that particular listing.

1840

01:47:59,158 --> 01:48:01,825

Now, it's starting to become more interesting over here.

1841

01:48:01,925 --> 01:48:04,994

Still, we have not finished all...

1842

01:48:07,221 --> 01:48:08,351

Scale.

1843

01:48:10,764 --> 01:48:12,172

Ten workers.

1844

01:48:14,500 --> 01:48:17,615

Over here so we can finish this one faster, again.

1845

01:48:17,995 --> 01:48:20,463

Let's go back to this management.

1846

01:48:20,563 --> 01:48:22,333

We still have 148.

1847

01:48:24,270 --> 01:48:25,534

What did I want to tell?

1848

01:48:26,125 --> 01:48:28,031

Okay. And now there are listings,

1849

01:48:28,131 --> 01:48:33,416

let's talk about another aspect that we could do.

1850

01:48:33,516 --> 01:48:37,625

Another piece of information that I've added here.

1851

01:48:38,135 --> 01:48:39,162

I'm conscious of the time

1852

01:48:39,262 --> 01:48:42,201

so I'm going to show the other interesting thing.

1853

01:48:42,301 --> 01:48:46,740

I think it's the reviews.

1854

01:48:46,956 --> 01:48:49,005

Reviews. Reviews are basically

1855

01:48:50,453 --> 01:48:54,822

reviews about an Airbnb listing

1856

01:48:54,922 --> 01:49:01,034

which contains a review of if it was good or if it was not good.

1857

01:49:01,134 --> 01:49:02,909

Also, who made the review.

1858

01:49:03,729 --> 01:49:07,657

The reviewee, to whom, and to what listing.

1859

01:49:07,800 --> 01:49:09,289

And also, we have this...

1860

01:49:12,189 --> 01:49:15,389

We have this approach, no, this functionality built in.

1861

01:49:17,697 --> 01:49:22,685

And I could tell if you had to do it for all the listings,

1862

01:49:23,303 --> 01:49:25,171

that would take an awful amount of time.

1863

01:49:25,771 --> 01:49:28,602

For each review, we also have the guys,

1864

01:49:28,702 --> 01:49:30,361

the people that made them,

1865

01:49:30,501 --> 01:49:32,078

and we have a different table for that.

1866

01:49:32,178 --> 01:49:34,216

We're collecting the data for this as well.

1867

01:49:34,687 --> 01:49:39,758

So, instead of telling you the review for one listing,

1868

01:49:39,858 --> 01:49:42,387

hopefully it will finish.

1869

01:49:45,017 --> 01:49:47,710

And just open a new panel here.

1870

01:49:47,810 --> 01:49:52,235

cd src. cd Airbnb, test.

1871

01:49:52,505 --> 01:49:56,183

So, this still isn't ideal because I do not know if that exists or not.

1872

01:49:56,283 --> 01:49:58,489

So, I'm going to take this one.

1873

01:49:58,916 --> 01:50:00,533

Actually, no, it's very remote.

1874

01:50:00,633 --> 01:50:01,915

Maybe that'll be interesting.

1875

01:50:02,105 --> 01:50:03,347

Take this one.

1876

01:50:05,393 --> 01:50:08,850

Copy.

1877

01:50:12,710 --> 01:50:13,997

Yes, it does.

1878

01:50:14,097 --> 01:50:15,102

It did work.

1879

01:50:16,088 --> 01:50:18,295

So, I'm telling it to fetch reviews for

1880

01:50:19,112 --> 01:50:20,148

this listing.

1881

01:50:21,444 --> 01:50:23,229

This hasn't understood what I want to tell.

1882

01:50:23,329 --> 01:50:25,221

Again, use the worker, right?

1883

01:50:25,421 --> 01:50:26,785

So...

1884

01:50:28,515 --> 01:50:30,478

That's its worker, okay.

1885

01:50:32,678 --> 01:50:36,023

So, yes, let me...

1886

01:50:36,375 --> 01:50:40,782

So, what it does is it connects to that Airbnb listing.

1887

01:50:43,508 --> 01:50:46,915

It found this number of reviews only.

1888

01:50:48,744 --> 01:50:51,474

And if we close that.

1889

01:50:51,574 --> 01:50:54,710

I have an extra table for this one so we can go to that listing,

1890

01:50:54,810 --> 01:50:57,649

to that table, we can see all the reviews.

1891

01:50:59,689 --> 01:51:01,598

So, that's the review ID,

1892

01:51:01,698 --> 01:51:04,667

when it was made, when we entered it into the database.

1893

01:51:04,767 --> 01:51:06,765

So, the same time, today.

1894

01:51:06,935 --> 01:51:10,172

And what's the text? "Brilliant property, perfect for us.

1895

01:51:10,272 --> 01:51:12,221

The ability to cook" and etc. etc.

1896

01:51:12,340 --> 01:51:14,168

Who made it, for which ID.

1897

01:51:14,268 --> 01:51:15,527

Which recipient, to whom.

1898

01:51:15,627 --> 01:51:17,445

And what's the response for it.

1899

01:51:18,605 --> 01:51:21,800

So, for the reviews, we have two pieces of information.

1900

01:51:21,900 --> 01:51:24,287

We have the actual reviews, the text,

1901

01:51:24,603 --> 01:51:29,352

which contain information things and in which language it was made.

1902

01:51:29,530 --> 01:51:31,079

And who made it.

1903

01:51:31,511 --> 01:51:32,749

Hopefully... Oh, yeah.

1904

01:51:32,849 --> 01:51:35,295

Okay. What is happening is also that

1905

01:51:38,516 --> 01:51:40,655

the system, whenever it finds

1906

01:51:40,755 --> 01:51:42,955

a new user that we have not encountered before,

1907

01:51:45,897 --> 01:51:51,026

it creates a new task and asks Airbnb to collect data from the user's API

1908

01:51:51,126 --> 01:51:53,086

and put them into our database.

1909

01:51:53,505 --> 01:51:57,050

And since we have made 10 workers over here,

1910

01:51:57,150 --> 01:52:02,839

it was fast enough to harvest all these users

1911

01:52:02,939 --> 01:52:04,698

in this small amount of time.

1912

01:52:04,898 --> 01:52:10,355

So, for that, essentially what is happening is that for that listing

1913

01:52:10,625 --> 01:52:18,052

we have these 38 people that have interacted with that one.

1914

01:52:18,412 --> 01:52:22,479

And that's all the public information that we collect from Airbnb's API.

1915

01:52:22,579 --> 01:52:24,388

That's the first name,

1916

01:52:24,876 --> 01:52:26,033

what it's about,

1917

01:52:26,133 --> 01:52:27,542

how many listings they have.

1918

01:52:27,642 --> 01:52:30,383

Some of the users are hosts as well.

1919

01:52:30,751 --> 01:52:32,349

Where they say they are from.

1920

01:52:32,449 --> 01:52:35,596

That different guy is from London.

1921

01:52:35,729 --> 01:52:38,188

And how he has verified himself.

1922

01:52:38,288 --> 01:52:40,896

He has verified using an email, phone, and Facebook.

1923

01:52:41,177 --> 01:52:43,474

That's what's the picture URL.

1924

01:52:43,604 --> 01:52:46,262

We're not storing the picture URLs per se.

1925

01:52:46,362 --> 01:52:51,479

We are storing the URL to that picture.

1926

01:52:53,368 --> 01:52:55,363

And when he made his profile.

1927

01:52:55,463 --> 01:52:59,800

When he added the profile and when was the last time he updated that profile.

1928

01:53:00,270 --> 01:53:02,090

And we click onto that one.

1929

01:53:02,190 --> 01:53:05,848

Here we can see a photograph of Stephen.

1930

01:53:07,316 --> 01:53:10,194

Hopefully, I was allowed to do that.

1931

01:53:10,694 --> 01:53:13,162

And, yeah, basically,

1932

01:53:13,865 --> 01:53:14,902

then you can collect...

1933

01:53:15,102 --> 01:53:18,310

If you have this piece of information, it's quite interesting because

1934

01:53:18,410 --> 01:53:21,837

the size of the user for the reviews...

1935

01:53:23,296 --> 01:53:24,884

Let's go back. So, you can see

1936

01:53:24,984 --> 01:53:27,237

when this review was made.

1937

01:53:27,337 --> 01:53:30,305

And if you group all these ones,

1938

01:53:30,405 --> 01:53:32,042

you can see that in 2000,

1939

01:53:33,219 --> 01:53:35,384

the first review was made in 2016

1940

01:53:35,484 --> 01:53:38,310

and he has only one review from that year.

1941

01:53:38,853 --> 01:53:43,341

For 2017, he had only five reviews.

1942

01:53:43,441 --> 01:53:45,789

For 2018, the guy was more popular.

1943

01:53:45,889 --> 01:53:49,684

He had this many for 2019. He had this many for 2020.

1944

01:53:49,784 --> 01:53:51,202

He had only this one, COVID.

1945

01:53:51,302 --> 01:53:54,932

And 2021, at the beginning of the year, he has one.

1946

01:53:55,656 --> 01:54:00,829

And, obviously, you can group by month,

1947

01:54:00,929 --> 01:54:02,935

days, different hours if you want.

1948

01:54:03,035 --> 01:54:06,704

You can go all the way down. You could create interesting graphs

1949

01:54:06,804 --> 01:54:09,172

to see how the reviews... And aggregate that.

1950

01:54:09,382 --> 01:54:11,760

Take all the reviews for all the listings for all of Jersey

1951

01:54:11,860 --> 01:54:16,717

and see how many reviews they have for 2008, 2020, 2021,

1952

01:54:16,817 --> 01:54:18,923

or whatever you want to do.

1953

01:54:21,985 --> 01:54:25,893

Basically, that is it from me.

1954

01:54:26,343 --> 01:54:30,581

I will skip all this,

1955

01:54:30,681 --> 01:54:32,660

all the commands that I put over here.

1956

01:54:33,646 --> 01:54:34,805

Oh, yeah, and this bit.

1957

01:54:35,204 --> 01:54:39,562

So far, what we've done is we've issued the commands manually.

1958

01:54:40,434 --> 01:54:44,559

Manually, give me all the calendars from that area of interest that

1959

01:54:44,659 --> 01:54:46,397

I told you that I want the calendars from.

1960

01:54:47,271 --> 01:54:49,279

That's not very convenient if you want a service.

1961

01:54:49,599 --> 01:54:51,917

So, what is happening inside,

1962

01:54:52,017 --> 01:54:53,486

it also has a scheduler.

1963

01:54:53,634 --> 01:55:00,024

Scheduler, it's a piece of code that

1964

01:55:00,124 --> 01:55:02,773

kicks these tasks, the initial tasks,

1965

01:55:02,873 --> 01:55:04,302

the starting tasks,

1966

01:55:04,821 --> 01:55:07,947

prespecified every four hours,

1967

01:55:08,047 --> 01:55:09,883

specified times, time schedules.

1968

01:55:10,201 --> 01:55:12,605

And also, it's embedded in that.

1969

01:55:12,705 --> 01:55:15,754

If we activate the mode,

1970

01:55:19,378 --> 01:55:21,007

activate the framework,

1971

01:55:21,107 --> 01:55:24,035

then a beat more, how it's called, like a heartbeat.

1972

01:55:24,675 --> 01:55:27,623

So, I've set it up in the code.

1973

01:55:33,234 --> 01:55:39,653

cd src. cd Airbnb explorer dot hear.

1974

01:55:41,012 --> 01:55:44,420

src. dj_airbnb.

1975

01:55:44,650 --> 01:55:46,561

We want the [inaudible].

1976

01:55:48,488 --> 01:55:51,447

I'm going to open this folder

1977

01:55:52,037 --> 01:55:53,935

and click this file.

1978

01:55:56,004 --> 01:55:57,958

I might even try to do [inaudible].

1979

01:56:01,718 --> 01:56:03,486

This one. Okay.

1980

01:56:03,975 --> 01:56:05,794

So, what's happening over here is

1981

01:56:05,894 --> 01:56:08,984

the database, basically, I'm telling it every four hours,

1982

01:56:09,573 --> 01:56:11,511

at minute zero every fourth hour,

1983

01:56:11,869 --> 01:56:13,168

run this command.

1984

01:56:13,268 --> 01:56:15,087

What this command does is that

1985

01:56:15,187 --> 01:56:17,215

it's going to collect all the listings

1986

01:56:17,315 --> 01:56:19,611

for the areas that we enabled it.

1987

01:56:20,259 --> 01:56:21,264

What this command...

1988

01:56:21,364 --> 01:56:24,663

Again, how many? Up to 5000.

1989

01:56:24,763 --> 01:56:26,021

If we have like 20,000,

1990

01:56:26,121 --> 01:56:28,990

it will only collect the first 5000,

1991

01:56:29,090 --> 01:56:31,005

but they are the oldest ones.

1992

01:56:32,516 --> 01:56:35,444

What we do here, update the reviews,

1993

01:56:35,544 --> 01:56:38,663

similar to that, every four hours.

1994

01:56:41,576 --> 01:56:43,754

All the reviews up to 1500,

1995

01:56:43,854 --> 01:56:46,399

that they have an age of,

1996

01:56:46,516 --> 01:56:49,164

they are older than this many hours,

1997

01:56:49,264 --> 01:56:50,603

that is 14 days.

1998

01:56:52,229 --> 01:56:55,378

So, you define your own schedule here programmatically

1999

01:56:55,478 --> 01:56:57,816

and you're telling it how you want your service to run.

2000

01:56:57,916 --> 01:56:58,995

How you want it.

2001

01:56:59,983 --> 01:57:02,951

In our case, I mean, I think we're doing about 80,000,

2002

01:57:03,719 --> 01:57:05,388

90,000 requests per day,

2003

01:57:05,488 --> 01:57:08,607

mostly in a couple of areas that we are interested in

2004

01:57:08,707 --> 01:57:10,732

and we are generating our tasks.

2005

01:57:13,051 --> 01:57:14,739

Let's go back a little bit to my,

2006

01:57:15,769 --> 01:57:19,063

the final bit of our story.

2007

01:57:19,931 --> 01:57:22,222

Go back to the thing.

2008

01:57:22,742 --> 01:57:25,019

We've already talked about scaling.

2009

01:57:25,308 --> 01:57:26,846

Obviously, the other interesting thing

2010

01:57:26,946 --> 01:57:29,906

if you are not familiar with the Docker technology

2011

01:57:30,006 --> 01:57:31,286

and things like Kubernetes, things like that,

2012

01:57:31,386 --> 01:57:34,614

and all these containers and things like that,

2013

01:57:36,383 --> 01:57:42,191

you're not restricted to running these workers only at your computer.

2014

01:57:42,291 --> 01:57:44,539

You could distribute them around the different nodes.

2015

01:57:44,639 --> 01:57:47,998

So, you could have hyper scalability.

2016

01:57:49,077 --> 01:57:52,187

If you're computer over here can only support 100 workers,

2017

01:57:52,287 --> 01:57:54,937

that's fine, then buy another, second node,

2018

01:57:55,037 --> 01:57:56,135

add it together,

2019

01:57:56,503 --> 01:58:01,098

and then you could have 200, 300, 800.

2020

01:58:02,295 --> 01:58:06,192

And that's it for me. That's my presentation.

2021

01:58:07,019 --> 01:58:10,188

The one thing I want to mention is that

2022

01:58:10,288 --> 01:58:12,095

if you try to do this by yourself,

2023

01:58:14,071 --> 01:58:15,629

it's going to be very, very hard.

2024

01:58:16,189 --> 01:58:17,488

Mainly, not because of the code,

2025

01:58:17,588 --> 01:58:19,577

it's because we, at this moment,

2026

01:58:19,837 --> 01:58:22,345

in our demonstrations, are using proxy networks,

2027

01:58:22,445 --> 01:58:25,592

we are using smart proxies through a private company.

2028

01:58:25,692 --> 01:58:28,986

Basically, what the smart proxy does is that...

2029

01:58:29,545 --> 01:58:33,572

Okay, I'll talk from the beginning. Airbnb, when you are bombarding it,

2030

01:58:33,672 --> 01:58:36,361

similarly, when it's bombarding it with requests,

2031

01:58:36,639 --> 01:58:39,178

it starts to throttle you,

2032

01:58:39,427 --> 01:58:42,762

meaning that it will be alive in the beginning

2033

01:58:42,862 --> 01:58:44,631

but after three minutes,

2034

01:58:44,930 --> 01:58:46,678

two minutes or whenever they're putting it,

2035

01:58:46,778 --> 01:58:50,427

they're putting you on their blacklist, let's say.

2036

01:58:50,527 --> 01:58:51,555

They are not...

2037

01:58:52,104 --> 01:58:55,102

They are going to

2038

01:58:55,668 --> 01:58:58,308

block your requests for a certain amount of time.

2039

01:58:58,408 --> 01:59:01,166

And you keep doing it, they're going to keep blocking you even more.

2040

01:59:02,035 --> 01:59:04,261

At some point, you won't be able to get any information

2041

01:59:04,361 --> 01:59:05,978

until you wait, like, a couple of hours

2042

01:59:06,078 --> 01:59:09,677

to go back to the normal operations.

2043

01:59:09,877 --> 01:59:11,975

So, to go around that,

2044

01:59:12,501 --> 01:59:16,009

the only way to go around that is actually to mask your requests

2045

01:59:16,109 --> 01:59:18,103

so that they come from different IPs.

2046

01:59:19,192 --> 01:59:21,802

And for us, it's the easiest way to do it.

2047

01:59:22,472 --> 01:59:25,701

We found a commercial provider that gave us a smart proxy,

2048

01:59:26,389 --> 01:59:29,749

which, basically, what it, all our requests are going through them.

2049

01:59:30,143 --> 01:59:32,133

And the smart proxy, what it does is,

2050

01:59:32,674 --> 01:59:37,074

I am requesting them

2051

01:59:37,174 --> 01:59:39,383

to do that.

2052

01:59:39,821 --> 01:59:42,042

They're doing that per my requests

2053

01:59:42,142 --> 01:59:44,198

through a worker, their worker,

2054

01:59:44,336 --> 01:59:46,085

and if the worker gets banned,

2055

01:59:46,713 --> 01:59:50,612

then they do it using a different worker with a different IP.

2056

01:59:50,948 --> 01:59:52,717

So, what I get back is

2057

01:59:53,017 --> 01:59:55,211

only the correct result.

2058

01:59:56,555 --> 01:59:57,875

The end result.

2059

01:59:58,075 --> 02:00:02,635

That's not 100% bulletproof

2060

02:00:02,735 --> 02:00:05,813

but we have about over 95% success with that,

2061

02:00:05,981 --> 02:00:07,120

which I'm very happy with.

2062

02:00:10,139 --> 02:00:11,647

So, if you try to do it by yourself,

2063

02:00:11,747 --> 02:00:14,282

initially you would be, in the large scale especially,

2064

02:00:14,500 --> 02:00:17,681

initially you would be able to have some successes

2065

02:00:17,781 --> 02:00:22,998

but don't find yourself surprised

2066

02:00:23,098 --> 02:00:26,167

if at some point you're going to have a lot of 404s,

2067

02:00:26,267 --> 02:00:27,325

503s, 501s,

2068

02:00:27,425 --> 02:00:29,743

I don't remember what the error codes are

2069

02:00:29,843 --> 02:00:32,732

that come back from Airbnb when you fail something.

2070

02:00:34,348 --> 02:00:38,250

And I know that I've spilled over five minutes over time.

2071

02:00:38,495 --> 02:00:41,195

If you have any questions, I'll be happy to answer them.

2072

02:00:43,401 --> 02:00:47,263

I'll be happy to answer them now.

2073

02:00:47,591 --> 02:00:48,970

Thanks, Nick. That's great.

2074

02:00:49,070 --> 02:00:51,308

And we have spilled a wee bit over the time

2075

02:00:51,408 --> 02:00:53,534

but I thought it was hopefully useful

2076

02:00:53,634 --> 02:00:56,302

to let Nick finish with just the last few points there.

2077

02:00:57,690 --> 02:01:01,177

We did have one question, just about the API key that we use,

2078

02:01:01,277 --> 02:01:02,936

and I already answered that in the chat.

2079

02:01:03,036 --> 02:01:05,235

But for everyone else's benefit who didn't see that,

2080

02:01:05,812 --> 02:01:08,647

you do not need your own API key for this application.

2081

02:01:08,747 --> 02:01:11,725

We actually use Airbnb's own generic API key,

2082

02:01:11,825 --> 02:01:14,224

the one that is used by the website itself

2083

02:01:14,324 --> 02:01:15,753

in its normal course of operation.

2084

02:01:15,853 --> 02:01:18,739

And, as I've sort of mentioned in the chat, we're effectively mimicking

2085

02:01:18,839 --> 02:01:22,795

the behaviour of Airbnb's responsive web application.

2086

02:01:22,895 --> 02:01:24,439

And rather than

2087

02:01:24,957 --> 02:01:28,206

extracting just the HTML results of

2088

02:01:28,306 --> 02:01:32,303

that hydration process that Nick described earlier on.

2089

02:01:33,209 --> 02:01:35,411

Rather than grab an HTML text

2090

02:01:35,511 --> 02:01:38,908

or, you know, navigating the Document Object Model,

2091

02:01:39,076 --> 02:01:40,966

we're just going a little bit further upstream

2092

02:01:41,293 --> 02:01:44,289

and grabbing the data from source via the API.

2093

02:01:45,759 --> 02:01:48,258

The other question we've received, will you be providing instruction

2094

02:01:48,358 --> 02:01:50,295

to the walkthrough session so we can have a go later on?

2095

02:01:50,395 --> 02:01:53,973

Absolutely. So, what we will circulate

2096

02:01:54,491 --> 02:01:58,530

in the next day or so will be, obviously, the slides that you've seen.

2097

02:01:58,630 --> 02:02:00,038

But, in addition to that,

2098

02:02:01,345 --> 02:02:03,557

a link to, which was actually already within these slides,

2099

02:02:03,657 --> 02:02:07,122

a link to our GitHub page where you can download the code.

2100

02:02:07,222 --> 02:02:13,128

There are README files there that outline the installation steps

2101

02:02:13,292 --> 02:02:17,933

for the, and point to those resources

2102

02:02:18,033 --> 02:02:19,129

that are dependencies.

2103

02:02:19,229 --> 02:02:20,957

So, it will point you to where you can grab Docker

2104

02:02:21,057 --> 02:02:22,605

and all of the things you'll need to know.

2105

02:02:24,548 --> 02:02:27,397

Within a Docker container as well as scalability that Nick mentioned.

2106

02:02:27,497 --> 02:02:29,970

It also offers us platform independence as well

2107

02:02:30,070 --> 02:02:32,507

so you can run this on a whole range of

2108

02:02:32,607 --> 02:02:34,095

different hosts as well.

2109

02:02:34,195 --> 02:02:36,410

So, the steps will all be there

2110

02:02:36,510 --> 02:02:40,589

so you can absolutely have a go and let us know how you get on with that.

2111

02:02:41,297 --> 02:02:43,873

But I also invited other questions a few moments ago and there's

2112

02:02:43,973 --> 02:02:46,657

no other questions that have come in and we are over time just now

2113

02:02:46,757 --> 02:02:49,374

so I think I'll probably just draw things to a close.

2114

02:02:49,474 --> 02:02:52,442

I'll thank Nick for the presentation

2115

02:02:52,542 --> 02:02:54,051

and for the coverage of

2116

02:02:55,190 --> 02:02:58,319

what we're really excited about as a data collection platform.

2117

02:02:58,868 --> 02:03:00,918

Really keen to hear how other people get on.

2118

02:03:01,018 --> 02:03:05,138

And thanks very much for joining us today.

2119

02:03:06,283 --> 02:03:07,861

Please, by all means, write to us.

2120

02:03:07,961 --> 02:03:11,846

You can contact us at UBDC with any follow up questions

2121

02:03:11,946 --> 02:03:13,666

or conversation points.

2122

02:03:15,145 --> 02:03:18,414

And, once again, for those of you particularly in research roles,

2123

02:03:18,514 --> 02:03:22,094

I would encourage you to register to take part in Thursday's session

2124

02:03:22,194 --> 02:03:24,764

as you'll see some of the ways that we're using these data.

2125

02:03:25,213 --> 02:03:28,081

And the possibilities that have been outlined today are

2126

02:03:30,106 --> 02:03:34,404

finding a really interesting purpose

2127

02:03:34,504 --> 02:03:35,722

and a research application.

2128

02:03:36,280 --> 02:03:39,509

Have a great day everyone and thanks again, bye bye.

Economic and Social Research Council
University of Glasgow
Urban Big Data Centre
University of Glasgow
7 Lilybank Gardens
Glasgow, G12 8RZ
T: +44 (0)141 330 2764
E: ubdc@glasgow.ac.uk
W: www.ubdc.ac.uk