

# Performance Tips

---



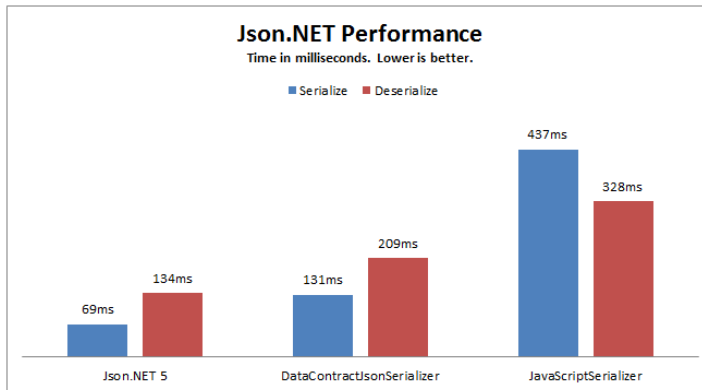
**Xavier Morera**

HELPING DEVELOPERS UNDERSTAND SEARCH & BIG DATA

@xmorera [www.xaviermorera.com](http://www.xaviermorera.com)



# Json.NET vs. Other Serializers



Multiple benchmarks put Json.NET as the fastest serializer

But more than comparing...

Focus on improving within Json.NET



# Performance Scenarios

**Manual  
Serialization**

**Fragments**

**Populate Objects**

**Merge**

**Attributes**



# How We Are Going to Measure

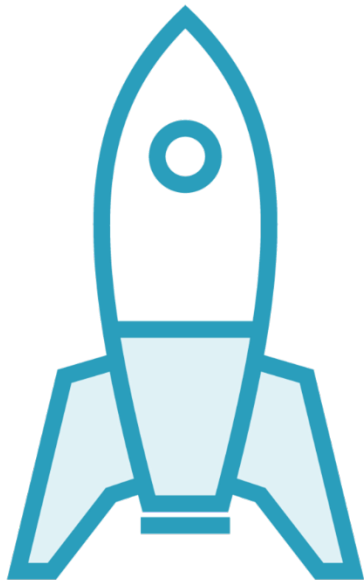
- Stopwatch → Easy way

```
Stopwatch watch = new Stopwatch();  
watch.Start();
```

```
Stopwatch watchNoAttributes = new Stopwatch();  
watchNoAttributes.ElapsedMilliseconds
```



# Serialize and Deserialize Manually



**Json.NET is extremely fast**

**But it relies on reflection**

**If speed is key**

- Serialize manually

**Use JsonTextReader and JsonTextWriter**

- Avoids using Reflection
- Less memory usage

**Fastest way of reading and writing JSON**

```
using (JsonWriter jsonWriter = new JsonTextWriter(stringWriter))
{
    jsonWriter.WriteStartObject();

    ...

    jsonWriter.WriteEndObject();
}
```

# Serialize and Deserialize Manually

## **Serialization with JsonConvert uses reflection**

- Reflection is an expensive operation

## **If performance is your top priority**

- Serialize and deserialize using manually



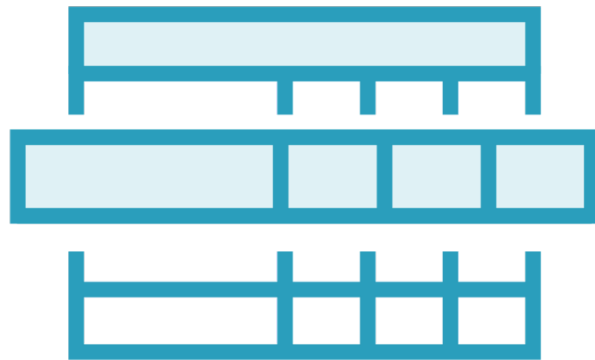
# Demo



**Serialize and deserialize manually**



# JSON Fragments



## Large JSON objects

- May take considerable time
  - Or consume a lot of memory
- While serializing or deserializing

## Possible to focus

- On specific subsection of the data

## Fragments

- Using Linq





logs1.json

```

1  [{
2    "ip": "192.168.1.1",
3    "cookie": "__ar_v4=4YCMENXFKFBQLNQCLQOV3GS%3A20141026%3A2%7CNPTOMQSYYZABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A20141026%3A2",
4    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.89 Safari/537.36",
5    "acceptLanguage": "en-US,en;q=0.8,es;q=0.6",
6    "courseView": {
7      "userId": 3641,
8      "user": "user_3641",
9      "course": "Solr",
10     "watchedDate": "2015-07-13T00:00:00",
11     "secondsWatched": 696
12   },
13   "privateImportantData": {
14     "m_MaxCapacity": 2147483647,
15     "Capacity": 1024,
16     "m_StringValue":
17     "UWSVRLQCSBYHYBNHEMEDHXFXXXFTCKULYOTRTMNGXIJBXUTJRYVBIKWQUXSGWAI FMIKGUNRYWEXOXVACKKKKYKPEWAYNTYBIBBOPKMNHDHVDLHRXXPULPWGBRGJTODEXCPNWHKRXWGC
18     CHKSOHAPDDOSDTMWEDSOEREBUTUQNXVOITETGMTLOMATNWJCTMMUDQXJCJWUMJCFHYJIORPQIGAQQKNWJSVAKBLQVUQWPYYEKUMNNBMCVVLWQHRSVAKSMBTFNRMTASBJGUFVWYII FSDOO
19     OANBYBONGRJBGAVOSBDTCPVCVTVQBNSWEQYQYRBRLUADVTQYMTMRAUMXELEEEXMSXLWOAEPAALIGORGCDWDHABWJEAMMMVMMIHFJWBXHELROLGILECBHTERVOMKAQCMDXUWHMVECYUCOI
20     IWWQBIJUFSEOIUJXPQCXOPQJBXHXIYIBDMGPMIOSCHSCSREBTLBEAPTRFNPDALVWTCUNTICOBIGVOPITGDMGTBMSNJYRLSQKVHPFIMAILDQTMUQXHITIJPAAYCAJDWOPEVAGYCKSCDUP
21     JSFSSBKXHBGHPGGMPFONBOQUJXEXBFFTJQFVGQCLLQVTTYJVEFLXFBIDNXIDGECVWLWXAJFYDDBKRWBKNHSUSHOHICII IYFHSTWHCLUUYMUCAWKQWQXLCIOWOPWAWPDWTQXYSICMWUF
22     LLHKRRLKJGVQKIKMVWTCVQEFGLQDOMLNGCCLGEFIKFTQXAMPHUGOCSJUGCDFAHVVTIKIGYGQRLQQVNHGYAVSGWKMSGEEBYICIWVOOEJIOOVQBRYSTWGYLCPWWWOYTBPRCOHXQRQTQMEQA
23     SLAHLKDSIQGHYQRPJNNYDNMJABBFLIYXXVELNRIQLYTXBQRAYEGXQVRVEXPHKIWHARHJCMFBRFVNETOIIJAGGBAHGYAMWKXINQTFTHONSDOSTDRJJHXAHAQAXQADDMRDGHNYMAIOBUMMDMG
24     MCHPDPMMWAVQUOS",
25     "m_currentThread": 0
26   }
27 },
28 {
29   "ip": "192.168.1.1",
30   "cookie": "__ar_v4=4YCMENXFKFBQLNQCLQOV3GS%3A20141026%3A2%7CNPTOMQSYYZABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A20141026%3A2",
31   "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.89 Safari/537.36",
32   "acceptLanguage": "en-US,en;q=0.8,es;q=0.6",
33   "courseView": {
34     "userId": 1567,
35     "user": "user_1567",
36     "course": "dotTrace",
37     "watchedDate": "2015-07-02T00:00:00",
38     "secondsWatched": 22
39   }
40 }

```

Fragments

```
logEntry["courseView"]["secondsWatched"].Value<int>();
```

## JSON Fragments

Extract a subsection of a JSON text

Deserialize it

Instead of the full object



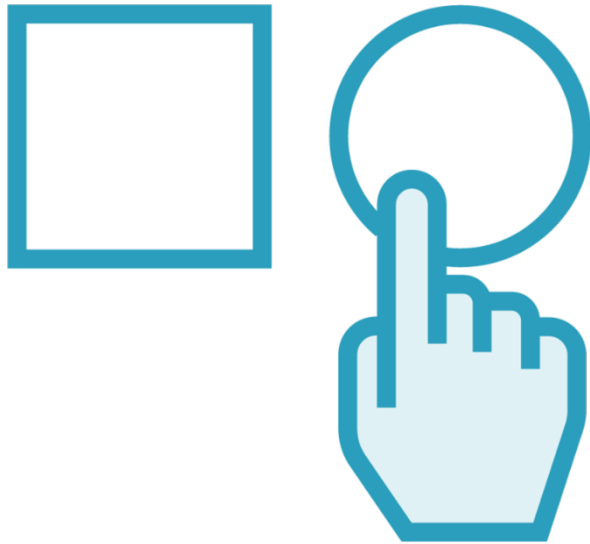
# Demo



## JSON Fragments



# JSON Populate



## **PopulateObject**

- Write specific values
- To large JSON objects

D:\Dropbox\xavier-morera-pluralsight-dev\json-csharp-jsondotnet-getting-started\demos\m5-18-memory-demo\logs1.json - Notepad++

FileEditSearchViewEncodingLanguageSettingsMacroRunPluginsWindow?

logs1.json

```
1  [{
2      "ip": "192.168.1.1",
3      "cookie": "__ar_v4=4YCMENXFKFBQLNQCLOV3GS%3A20141026%3A2%7CNPTOMQSYYZABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A20141026%3A2",
4      "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.89 Safari/537.36",
5      "acceptLanguage": "en-US,en;q=0.8,es;q=0.6",
6      "courseView": {
7          "userId": 3641,
8          "user": "user_3641",
9          "course": "Solr",
10         "watchedDate": "2015-07-13T00:00:00",
11         "secondsWatched": 696
12     },
13     "privateImportantData": {
14         "m_MaxCapacity": 2147483647,
15         "Capacity": 1024,
16         "m_StringValue":
17         "UWSVRLQCSBYHYBNHEMEDHXFXXXFTCKULYOTRTMNGXIJBXUTJRYVBIKWQUXSGWAI FMIKGUNRYWEXOXVACKKKKYKPEWAYNTYBIBBOPKMNDHVDELHRXXPULPWGBRGJTODEXCPNWHKRXWGC
18         CHKSOHAPDDOSDTMWEDSOEREBUTUQNXVOITETGMTLOMATNWJCTMMUDQXJCJWUMJCFHYJIORPQIGAQQKNWJSVAKBLQVUQWPYYEKUMNNBMCVVLWQHRSVAKSMBTFNRMTASBJGUFVWYIIFSDOO
19         OANBYBONGRJBGAVOSBDTCPVCVTVQBNWEQYQYRBRRLUADVTQYMTMRAUMXELEEEXMSXLWOAEPAALIGORCWDHABWJEAMMMVMMIHFJWBXHELROLGILECBHTERVOMKAQCMDXUWHMVECYUCOI
20         IWWQBIJUFSEOIUJXPQCXOPQJBXHXIYIXBDMGPMIOSCHSCSREBTLBEAPTRFNPDALVWTCCLUNTQCOBIUGVOPITGDMGTBMSNJYRLSQKVHPFIMAILDQTMUQXHITIJPAAYCAJDWOPEVAGYCKSCDUP
21         JSFSSBKXHBGHPGGMPFONBOQUJXEXBFFTJQFVGQCLLQVTTYJVEFLXFBIDNXIDGECVWLXAJFYDDDBKRWBKNHSUSHOHICIIIFYHSTWHCLUUYMUCAWKQWQWOLCOWOPWAWPDWTQXYSICMWUF
22         LLHKRRLKJGVQKIKMVWTCVQEFGLQDOMLNGCCLGEFIKFTQXAMPHUGOCSJUGCDFAHVVTIKIGYGQRLQQVNHGYAVSGWKMSGEEBYICIWVOOEJIOOVQBRYSTWGYLCPWWWOYTBPRCOHXQRQTQMEQA
23         SLAHLKDSIQGHYQRPJNNYDNMJABBFLIYXXVELNRIQLYTXBQRAYEGXQVRVEXPHKIWHARHJCMFBRFVNETOIIJAGGBAHGYAMWKXINQTFTHONSDOSTDRJHXAHQAXQADDMRDGHNYMAIOBUMMDMG
24         MCHPDPMAVQUOS",
25         "m_currentThread": 0
26     }
27 },
28 {
29     "ip": "192.168.1.1",
30     "cookie": "__ar_v4=4YCMENXFKFBQLNQCLOV3GS%3A20141026%3A2%7CNPTOMQSYYZABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A20141026%3A2",
31     "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.89 Safari/537.36",
32     "acceptLanguage": "en-US,en;q=0.8,es;q=0.6",
33     "courseView": {
34         "userId": 1567,
35         "user": "user_1567",
36         "course": "dotTrace",
37         "watchedDate": "2015-07-02T00:00:00",
38         "secondsWatched": 22
39     }
40 }
41 ]
```

JavaScript filelength: 161325 lines: 1900Ln: 16 Col: 1029 Sel: 0 | 0UNIXUTF-8INS

PopulateObject

```
JsonConvert.PopulateObject(jsonReviewed, log);
```

## JSON Populate

**Write a specific value to a JSON object**

- With a single call
- Using **PopulateObject**



# Demo



## Populate JSON Objects



# JSON Merge



## Merge two JSON objects

- Might be a complex operation

## Json.NET provides merge functionality

### Logic

- Name/value pairs copied across
  - Skipping nulls if the existing property is not null
  - Specify how to handle arrays



\*D:\Dropbox\xavier-moreira-pluralsight-dev\json-csharp-jsondotnet-getting-started\demos\m5-18-memory-demo\logs1.json - Notepad++

FileEditSearchViewEncodingLanguageSettingsMacroRunPluginsWindow?

logs1.json

update.json

1

[{

2

"ip": "192.168.1.1",

3

"cookie":

4

"\_\_ar\_v4=4YCMENXFKFBQLNQCLQOV3GS%3A20141026%3A2%7CNPTOMQSYYZ

5

ABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A2014

6

1026%3A2",

7

"userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64)

8

AppleWebKit/537.36 (KHTML, like Gecko)

9

Chrome/44.0.2403.89 Safari/537.36",

10

"acceptLanguage": "en-US,en;q=0.8,es;q=0.6",

11

"courseView": null,

12

"privateImportantData": {

13

"m\_MaxCapacity": 2147483647,

14

"Capacity": 1024,

15

"m\_StringValue":

16

"UWSVRLQCSBYHYBNHEMEDHXFXXXFTCKULYOTRTMNGXIJBXUTJRYVBI

17

KWQUXSGWAI FMIKGUNRYWEXOXVACKKKKYKPEWAYNTYBIBBOPKMNNDHVDE

18

LHRXXPULPWGBRGJTODEXCPNWHKRXWGCHKSOHAPDDOSDTMWEDSOEREB

19

UTUQNXVOITETGMTLOMATNWJCTMMUDQXJCJWUMJCFHYJIORPQIGAQQKN

20

WJSVAKBLQVUQWPYYEKUMNNBMCVVLWQHRSVAKSMBTFNRMTASBJGUFVWY

21

IIFSDOOOANBYBONGRJBGAVOSBDTCPVCVTVQBNSWEQYQYRBRUADVTQY

22

MTMRAUMXELEEEXMSXLWOAEPAALIGORGCDWDHABWJEAMMMVMMIHFWJBX

23

HELROLGILECBHTERVOMKAQCMDXUWHMVECYUCOIIWWQBIJUFSEOIUJPX

24

QCXOPQJBXHXIYIXBDMGPMIOSCHSCSREBTLBEAPTRFNPDALVWTCUNTCO

25

BIUGVOPITGDMGTBMSNJYRLSQKVHPFIMAILDQTMUQXHITIJPAAYCAJDW

26

OPEVAGYCKSCDUPJSFSSBKXHBGHPGGMPFONBOQUJXEXBFFTJQFVGQCL

27

LQVTITYJVEFLXFBIDNXIDGEWCVLWXAJFYDDBKRWBKNHSUSHOHICIIYF

28

HSTWHCLUUYMUCAWKQWWOXLCIOWOPWAWPDWTQXYSICMWUFLHKKRLKJG

29

VQKIKMVWTCVQEFLQDOMLNGCCLGEFIKFTQXAMPHUGOCSJUGCDFAHVVT

30

IKIGYGQRLQQVNHGYAVSGWKMSGEEBYICIWVOOEJIOOVQBRYSTWGYLECPW

31

WWOYTBPRCOHXQRQTQMEQASLAHLKDSIQGHYQRPJNNYDNMJABBFLIYXXV

32

ELNRIQLYTXBQRAYEGXQRVEXPHKIWHARHJCMFBRFVNETOIIJAGGBAHGY

33

AMWKXINQTFTHONSODSTRJHXAHQAXQADDMRDGHNYMAIOBUMMDMGMC

34

PDPMAVQUOS",

35

"m\_currentThread": 0

36

}

37

},

38

}

1

[{

2

"courseView": {

3

"userId": 3641,

4

"user": "user\_3641",

5

"course": "Solr",

6

"watchedDate": "2015-07-13T00:00:00",

7

"secondsWatched": 696

8

},

9

},

10

{

11

"courseView": {

12

"userId": 1567,

13

"user": "user\_1567",

14

"course": "dotTrace",

15

"watchedDate": "2015-07-02T00:00:00",

16

"secondsWatched": 33

17

}

18

},

19

{

20

"courseView": {

21

"userId": 775,

22

"user": "user\_775",

23

"course": "dotTrace",

24

"watchedDate": "2015-07-04T00:00:00",

25

"secondsWatched": 796

26

}

27

},

28

{

29

"courseView": {

30

"userId": 769,

31

"user": "user\_769",

32

"course": "Solr",

33

"watchedDate": "2015-07-27T00:00:00",

34

"secondsWatched": 91

35

}

36

},

37

{

38

"courseView": {

39

"userId": 769,

40

"user": "user\_769",

41

"course": "Solr",

42

"watchedDate": "2015-07-27T00:00:00",

43

"secondsWatched": 91

44

}

45

},

46

{

47

"courseView": {

48

"userId": 769,

49

"user": "user\_769",

50

"course": "Solr",

51

"watchedDate": "2015-07-27T00:00:00",

52

"secondsWatched": 91

53

}

54

},

55

{

56

"courseView": {

57

"userId": 769,

58

"user": "user\_769",

59

"course": "Solr",

60

"watchedDate": "2015-07-27T00:00:00",

61

"secondsWatched": 91

62

}

63

},

64

{

65

"courseView": {

66

"userId": 769,

67

"user": "user\_769",

68

"course": "Solr",

69

"watchedDate": "2015-07-27T00:00:00",

70

"secondsWatched": 91

71

}

72

},

73

{

74

"courseView": {

75

"userId": 769,

76

"user": "user\_769",

77

"course": "Solr",

78

"watchedDate": "2015-07-27T00:00:00",

79

"secondsWatched": 91

80

}

81

},

82

{

83

"courseView": {

84

"userId": 769,

85

"user": "user\_769",

86

"course": "Solr",

87

"watchedDate": "2015-07-27T00:00:00",

88

"secondsWatched": 91

89

}

90

},

91

{

92

"courseView": {

93

"userId": 769,

94

"user": "user\_769",

95

"course": "Solr",

96

"watchedDate": "2015-07-27T00:00:00",

97

"secondsWatched": 91

98

}

99

},

100

{

101

"courseView": {

102

"userId": 769,

103

"user": "user\_769",

104

"course": "Solr",

105

"watchedDate": "2015-07-27T00:00:00",

106

"secondsWatched": 91

107

}

108

},

109

{

110

"courseView": {

111

"userId": 769,

112

"user": "user\_769",

113

"course": "Solr",

114

"watchedDate": "2015-07-27T00:00:00",

115

"secondsWatched": 91

116

}

117

},

118

{

119

"courseView": {

120

"userId": 769,

121

"user": "user\_769",

122

"course": "Solr",

123

"watchedDate": "2015-07-27T00:00:00",

124

"secondsWatched": 91

125

}

126

},

127

{

128

"courseView": {

129

"userId": 769,

130

"user": "user\_769",

131

"course": "Solr",

132

"watchedDate": "2015-07-27T00:00:00",

133

"secondsWatched": 91

134

}

135

},

136

{

137

"courseView": {

138

"userId": 769,

139

"user": "user\_769",

140

"course": "Solr",

141

"watchedDate": "2015-07-27T00:00:00",

142

"secondsWatched": 91

143

}

144

},

145

{

146

"courseView": {

147

"userId": 769,

148

"user": "user\_769",

149

"course": "Solr",

150

"watchedDate": "2015-07-27T00:00:00",

151

"secondsWatched": 91

152

}

153

},

154

{

155

"courseView": {

156

"userId": 769,

157

"user": "user\_769",

158

"course": "Solr",

159

"watchedDate": "2015-07-27T00:00:00",

160

"secondsWatched": 91

161

}

162

},

163

{

164

"courseView": {

165

"userId": 769,

166

"user": "user\_769",

167

"course": "Solr",

168

"watchedDate": "2015-07-27T00:00:00",

169

"secondsWatched": 91

170

}

171

},

172

{

173

"courseView": {

174

"userId": 769,

175

"user": "user\_769",

176

"course": "Solr",

177

"watchedDate": "2015-07-27T00:00:00",

178

"secondsWatched": 91

179

}

180

},

181

{

182

"courseView": {

183

"userId": 769,

184

"user": "user\_769",

185

"course": "Solr",

186

"watchedDate": "2015-07-27T00:00:00",

187

"secondsWatched": 91

188

}

189

},

190

{

191

"courseView": {

192

"userId": 769,

193

"user": "user\_769",

194

"course": "Solr",

195

"watchedDate": "2015-07-27T00:00:00",

196

"secondsWatched": 91

197

}

198

},

199

{

200

"courseView": {

201

"userId": 769,

202

"user": "user\_769",

203

"course": "Solr",

204

"watchedDate": "2015-07-27T00:00:00",

205

"secondsWatched": 91

206

}

207

},

208

{

209

"courseView": {

210

"userId": 769,

211

"user": "user\_769",

212

"course": "Solr",

213

"watchedDate": "2015-07-27T00:00:00",

214

"secondsWatched": 91

215

}

216

},

217

{

218

"courseView": {

219

"userId": 769,

220

"user": "user\_769",

221

"course": "Solr",

222

"watchedDate": "2015-07-27T00:00:00",

223

"secondsWatched": 91

224

}

225

},

226

{

227

"courseView": {

228

"userId": 769,

229

"user": "user\_769",

230

"course": "Solr",

231

"watchedDate": "2015-07-27T00:00:00",

232

"secondsWatched": 91

233

}

234

},

235

{

236

"courseView": {

237

"userId": 769,

238

"user": "user\_769",

239

"course": "Solr",

240

"watchedDate": "2015-07-27T00:00:00",

241

"secondsWatched": 91

242

}

243

},

244

{

245

"courseView": {

246

"userId": 769,

247

"user": "user\_769",

248

"course": "Solr",

249

"watchedDate": "2015-07-27T00:00:00",

250

"secondsWatched": 91

251

}

252

},

253

{

254

"courseView": {

255

"userId": 769,

256

"user": "user\_769",

257

"course": "Solr",

258

"watchedDate": "2015-07-27T00:00:00",

259

"secondsWatched": 91

260

}

261

},

262

{

263

"courseView": {

264

"userId": 769,

265

"user": "user\_769",

266

"course": "Solr",

267

"watchedDate": "2015-07-27T00:00:00",

268

"secondsWatched": 91

269

}

270

},

271

{

272

"courseView": {

273

"userId": 769,

274

"user": "user\_769",

275

"course": "Solr",

276

"watchedDate": "2015-07-27T00:00:00",

277

"secondsWatched": 91

278

}

279

},

280

{

281

"courseView": {

282

"userId": 769,

283

"user": "user\_769",

284

"course": "Solr",

285

"watchedDate": "2015-07-27T00:00:00",

286

"secondsWatched": 91

287

}

288

},

289

{

290

"courseView": {

291

"userId": 769,

292

"user": "user\_769",

293

"course": "Solr",

294

"watchedDate": "2015-07-27T00:00:00",

295

"secondsWatched": 91

296

}

297

},

298

{

299

"courseView": {

300

"userId": 769,

301

"user": "user\_769",

302

"course": "Solr",

303

"watchedDate": "2015-07-27T00:00:00",

304

"secondsWatched": 91

305

}

306

},

307

{

308

"courseView": {

309

"userId": 769,

310

"user": "user\_769",

311

"course": "Solr",

312

"watchedDate": "2015-07-27T00:00:00",

313

"secondsWatched": 91

314

}

315

},

316

{

317

"courseView": {

318

"userId": 769,

319

"user": "user\_769",

320

"course": "Solr",

321

"watchedDate": "2015-07-27T00:00:00",

322

"secondsWatched": 91

323

}

324

},

325

{

326

"courseView": {

327

"userId": 769,

328

"user": "user\_769",

329

"course": "Solr",

330

"watchedDate": "2015-07-27T00:00:00",

331

"secondsWatched": 91

332

}

333

},

334

{

335

"courseView": {

336

"userId": 769,

337

"user": "user\_769",

338

"course": "Solr",

339

"watchedDate": "2015-07-27T00:00:00",

340

"secondsWatched": 91

341

}

342

},

343

{

344

"courseView": {

345

"userId": 769,

346

"user": "user\_769",

347

"course": "Solr",

348

"watchedDate": "2015-07-27T00:00:00",

349

"secondsWatched": 91

350

}

351

},

352

{

353

"courseView": {

354

"userId": 769,

355

"user": "user\_769",

356

"course": "Solr",

357

"watchedDate": "2015-07-27T00:00:00",

358

"secondsWatched": 91

359

}

360

},

361

{

362

"courseView": {

363

"userId": 769,

364

"user": "user\_769",

365

"course": "Solr",

366

"watchedDate": "2015-07-27T00:00:00",

367

"secondsWatched": 91

368

}

369

},

370

{

371

"courseView": {

372

"userId": 769,

373

"user": "user\_769",

374

"course": "Solr",

375

"watchedDate": "2015-07-27T00:00:00",

376

"secondsWatched": 91

377

}

378

},

379

{

380

"courseView": {

381

"userId": 769,

382

"user": "user\_769",

383

"course": "Solr",

384

"watchedDate": "2015-07-27T00:00:00",

385

"secondsWatched": 91

386

}

387

},

388

{

389

"courseView": {

390

"userId": 769,

391

"user": "user\_769",

392

"course": "Solr",

393

"watchedDate": "2015-07-27T00:00:00",

394

"secondsWatched": 91

395

}

396

},

397

{

398

"courseView": {

399

"userId": 769,

400

"user": "user\_769",

401

"course": "Solr",

402

"watchedDate": "2015-07-27T00:00:00",

403

"secondsWatched": 91

404

}

405

},

406

{

407

"courseView": {

408

"userId": 769,

```
authorConcat.Merge(authorConcatMerge, new JsonMergeSettings  
{  
    MergeArrayHandling = MergeArrayHandling.Concat  
});
```

## JSON Merge

### Merge two JSON objects

- Specify how to handle arrays
- Concat, union, replace, and merge



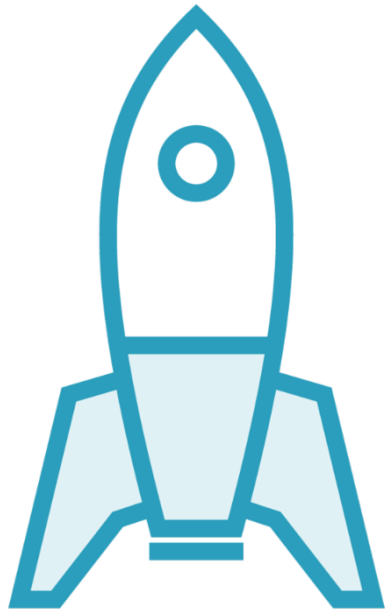
# Demo



## Merge JSON Objects



# Using Attributes for Performance



**If you are working with a large JSON object**

- But you only need certain values

**You can use attributes**

- To control what gets serialized
- Can lead to an increase in performance

**Serialize and deserialize**

- Only what you need



\*C:\Users\xavier\Documents\update.json - Notepad++

FileEditSearchViewEncodingLanguageSettingsMacroRunPluginsWindow?

logs1.json

14 {

15 "ip": "192.168.1.1",

16 "cookie":

17 "\_\_ar\_v4=4YCMENXFKFBQLNQCLOV3GS%3A20141026%3A2%7CNPTOMQSYYZABNNUIQDRAKL%3A20141026%3A2%7CBFLWHRV7W5FLTIZIQ4OSO6%3A20141026%3A2",

18 "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.89 Safari/537.36",

19 "acceptLanguage": "en-US,en;q=0.8,es;q=0.6",

20 "courseView": {

21 "userId": 1567,

22 "user": "user\_1567",

23 "course": "dotTrace",

24 "watchedDate": "2015-07-02T00:00:00",

25 "secondsWatched": 33

26 },

27 "privateImportantData": {

28 "m\_MaxCapacity": 2147483647,

29 "Capacity": 1024,

30 "m\_StringValue":

31 "YHKGEEJASVFTEWKJMAUVNNTKMCVETUWILLDEVGXDGDVPQFRTKEGDPPELLNASBHEGXXV

32 "IRSFNGCRPKQDSGKCTPOTSGYRFPVNWUBVXRUAAPYAXNBTKGLQFJPIDDBRAPRSFVNNRK

33 "XJJBPYOSTRXWFOBFPPQGNQUIYOXGUBYMBYICGETQTKIISDJDQYGWHCWGNBBABHWSRLV

34 "FFYYKWOUCHLOVETPVTCEXBBNFCOLMPPGGDWRYUYRGTYCLEVGRBXPBWDDKSIDAJPRUH

35 "PAWXVXDLQHCLTWWHTLQAYJASGGMVRWEHGDOUTEYMUFBVBYKGRPDATIFADRPLGSTDUE

36 "QIBXAHRNEDDRGHBOAHAGLTITGCNNFBLSGTHCESOVIGECWKKIJQSLDNSVYGRXSEICRKL

37 "LYYKGRDXFPGWUQFDNIYPRBLCWBQDEWCJLYXIJGMTYPNFSIVGMATTSUSKMWMEBATHNVL

38 "URVSHUEPCNGIOEJPDDIQFNUMSHSXUFRVWBOPHMXNEQVGNXYQKPEFFUPDBFNNLQGIOM

39 "QDUYUPJCLOROXUHCNFPALPCLKOBKBNJNRBGVULTOUCIALLRHFDJBHCVORKJTQYBXFEV

40 "VNLGCKLPNSKYDTGEQKHGPFDRDKBAXUBKTHPLDSDAQULHUGKGAJTJNWWGPVIVAJNULIPR

41 "NHDNLNJQKDWVYUANARVWJIUGKGYGGQVMPSSRRGIUEGMDRALWEIATSLSIEMVBHJJEMBIF

42 "FCIHRWRQVFCGXJUXGVDBWKNEKGNTPVPQQFQBARCVSCLYJOCWOLWPXHIUODWKJXTNOLO

43 "SSSWOTGHTXOUUEPFKKASJVUMVTQMUHWJXYJDUNIUQANWAICGSSJDORFVXGBTCTLMBEV

44 "UTLJVCEBIIIXLMRKSCWTUFSNFOLOSBNBUVMGLGIMAKPWWAUOBRAJOGJPBFGUJNKEUGCP

45 "ALXMTJSJOVAETWNBVELOLVEGYNCMYLBGXENOSMPPEMYBFEBBYJIPTMDEHRMFHFK",

46 "m\_currentThread": 0

47 }

48 },

49 }

update.json

1 [{

2 "courseView": {

3 "userId": 3641,

4 "user": "user\_3641",

5 "course": "Solr",

6 "watchedDate": "2015-07-13T00:00:00",

7 "secondsWatched": 696

8 }

9 }]

JavaScript file

length: 155428 lines: 1867

Ln: 39 Col: 1 Sel: 0 | 0

Dos\Windows UTF-8 INS

[JsonIgnoreAttribute]

```
public int userId { get; set; }
```

## Using Attributes for Performance

### Attributes to increase performance

- Specifying what you want to serialize and deserialize



# Demo



## Using Attributes for Increasing Performance



# Takeaway



**Manual Serialization**

**Fragments**

**Populate Objects**

**Merge**

**Attributes**

