Вагиф Абилов

Фронтенд на функциональном языке? Подержите мое пиво!

Вопрос к докладчику

Как к вам пришла эта идея?

Вопрос к аудитории

Почему вы пришли на этот доклад?

Наш продукт


Til nrk.no
NRK TV
Se TV direkte
Kategorier
TV-guide
Søk
Populært
Nytt
Side om side 6:10
Drømmeturen 4:6
Lilyhammer 1:8
TEMA: Andre verdenskrig
Aktuelt
Populært
Nytt
Meny

Я – разработчик бэкенда

Почему разработчика бэкенда должен заботить фронтенд?

Должен ли каждый разработчик понимать это?

```
git branch -r | grep origin | xargs -I {} bash -c 'if
[[ ! `git cherry HEAD {}` ]]; then echo "{} -- $(git
show --format=%an {} | head -n 1)"; fi'
```

Должен ли каждый разработчик понимать это?

```
git branch -r | grep origin | xargs -I {} bash -c 'if
[[ ! `git cherry HEAD {}` ]]; then echo "{} -- $(git
show --format=%an {} | head -n 1)"; fi’
```

Выдает авторов неготовых (unmerged) бранчей

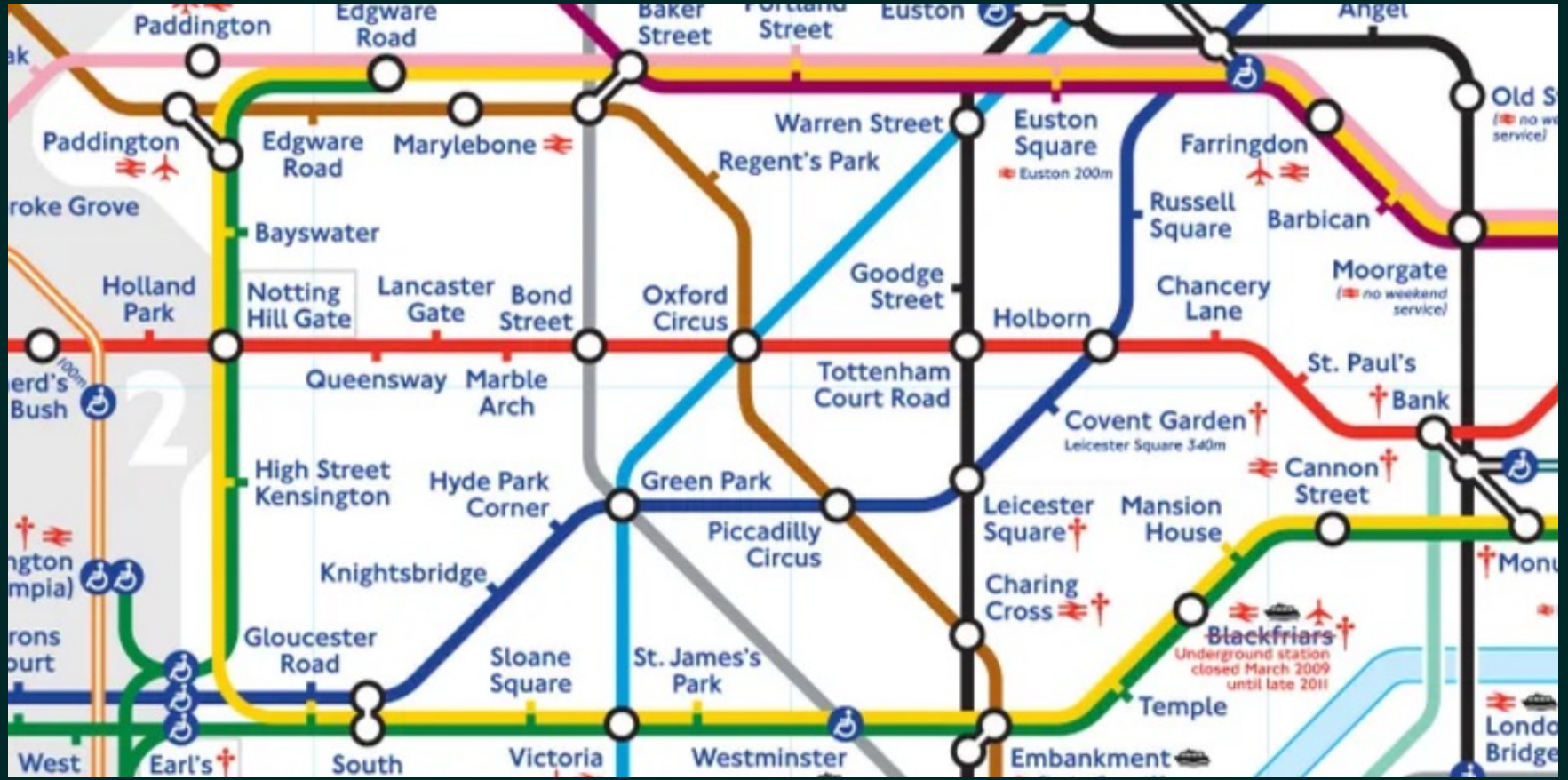

TV
Radio
1
0
0
0
0
155,847
1,783
29,289
337,285
2,170
35,209
NRK EdgeTV
Akamai
IP-Only
Telenor
MiniCDN
38
N/A
Critical error
29421

Настоятельно рекомендую доклад

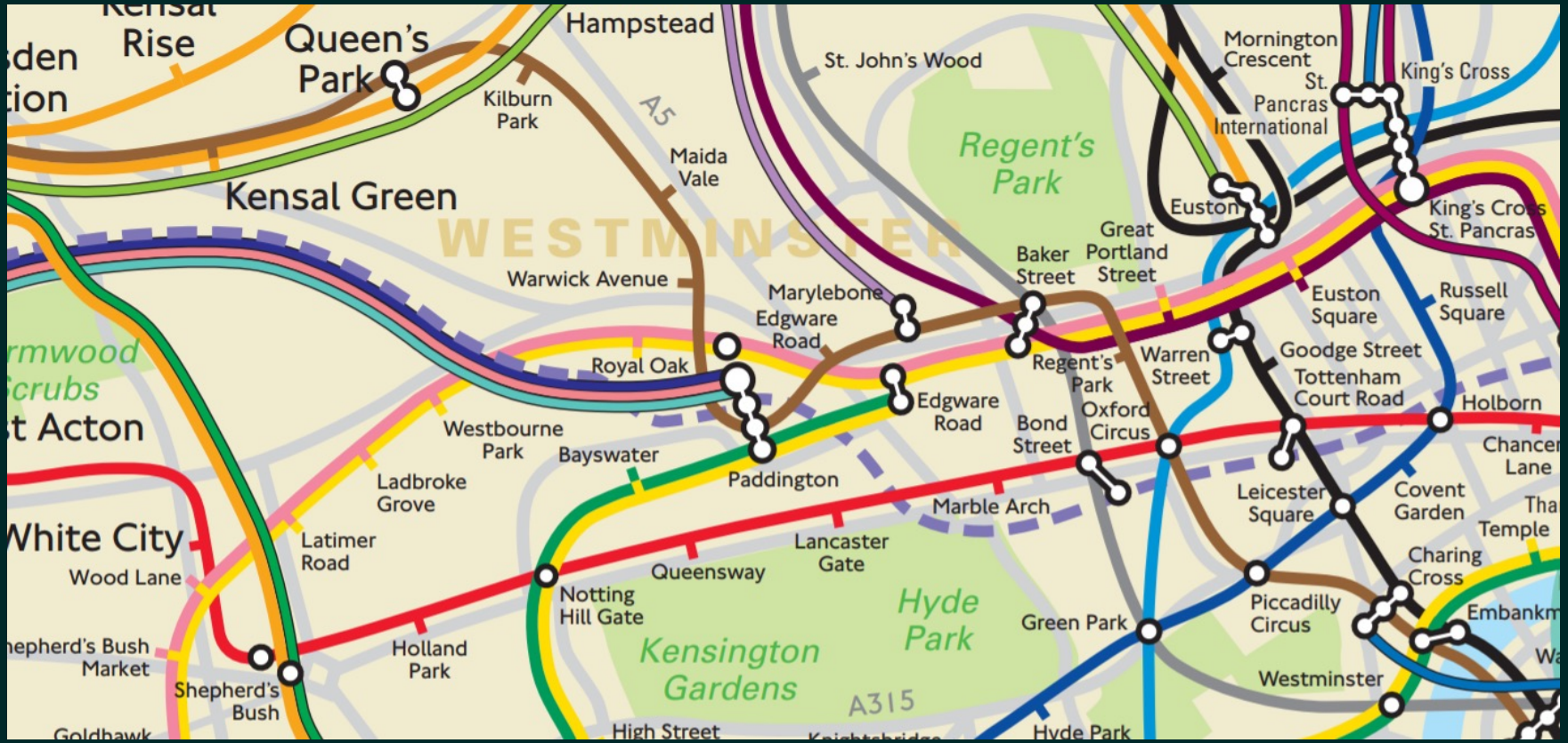
Дэвид Эванс
«Визуализация качества»

https://www.youtube.com/watch?v=jzY_i-QMW8M

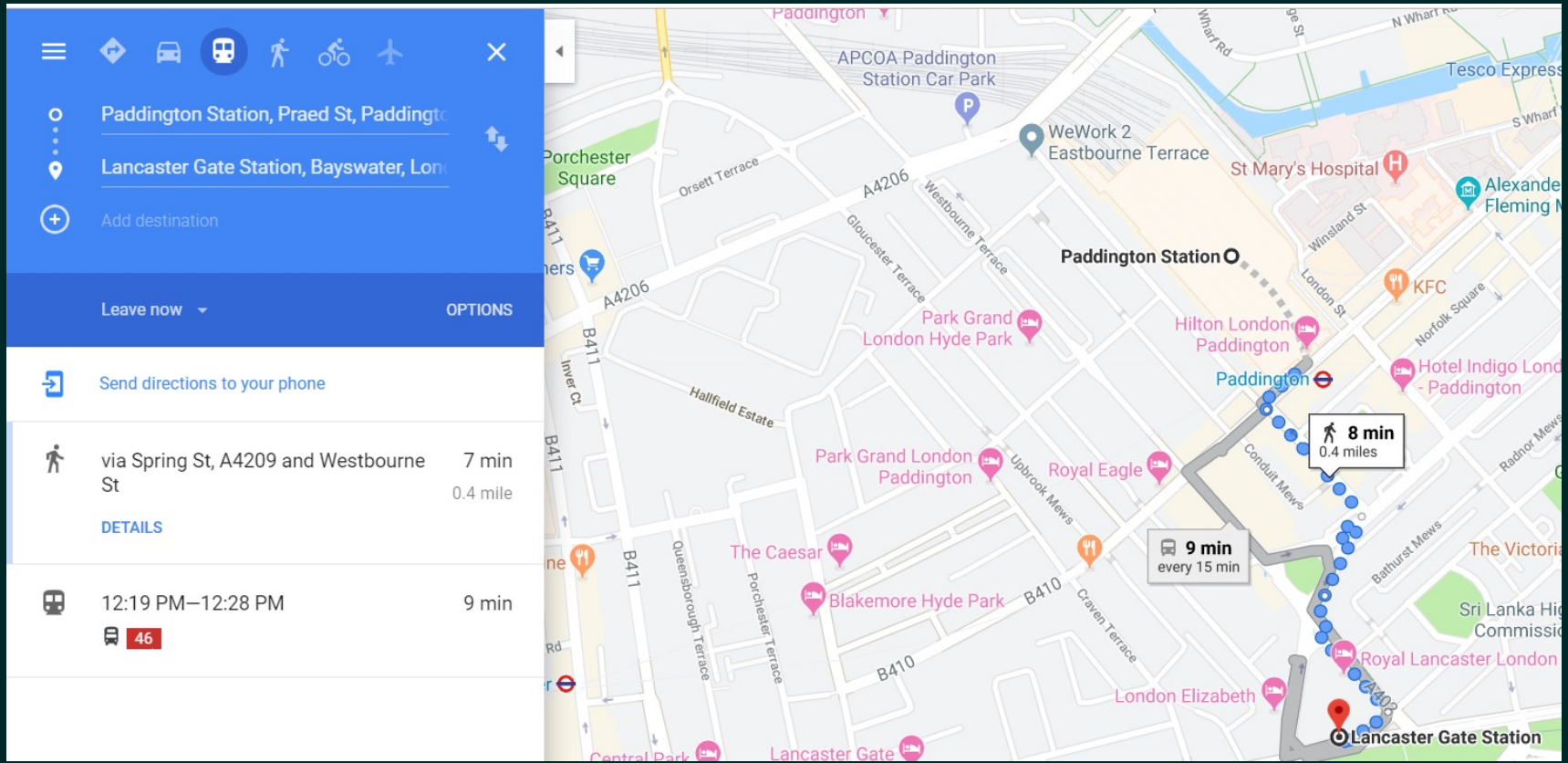
Поездка от вокзала Паддингтон до Ланкастер гейт



Поездка от вокзала Паддингтон до Ланкастер гейт



Поездка от вокзала Паддингтон до Ланкастер гейт



Исследования показывают, что 30% пассажиров выбирают более длинный маршрут по знаменитой карте лондонской подземки, искажающей расстояние между станциями

https://www.dailymail.co.uk/news/article-2000847/30-passengers-longer-routes-Londons-Tube-map-misrepresents-distances-stations.html

Фронтенд нужен не только конечному пользователю
Фронтенд нужен для визуализации качества работы системы

Должен ли каждый разработчик изучить HTML, JavaScript и CSS?

Критерии выбора технологии для фронтенда

- Сложность изучения
- Эффективность
- Интеграция в текущие средства разработки
- Интеграция в текущий программный код

Не существует общего совета для всех, варианты…

- JavaScript/CSS
- ASP.NET (PHP, JSP, Spring MVC…)
- WebAssembly для C#/Blazor (Rust, Java, Python…)
- Функциональный подход с Elm, F#/Fable, ReasonML, ClojureScript…

Функциональный подход, зачем?

Преимущества функционального подхода к фронтенду

- Неизменяемость данных
- Функцинальные паттерны (promises/futures)
- Алгебраические типы данных
- Легче тестировать
- Отладчик с машиной времени (time travelling debugger)
- Web функционален, не объектно-ориентирован

Наш выбор – функциональный подход с архитектурой Elm

Что такое Elm?

- Elm как язык
  - Чисто функциональный
  - Статически типизирован
  - Отсутствуют значения *null* или *undefined*
  - Полностью иммутабелен
  - Реактивен
- Elm компилируется в JavaScript
- Elm создан специально для разработки Web приложений

Модель программирования Elm: Model-View-Update

Ключевые элементы архитектуры Elm

- Model
- Initial state

```
init : ( Model, Cmd Msg )
```

- View

```
view : Model -> Html Msg
```

- Message
- Update

```
update : Msg -> Model -> ( Model, Cmd Msg )
```

- Subscription

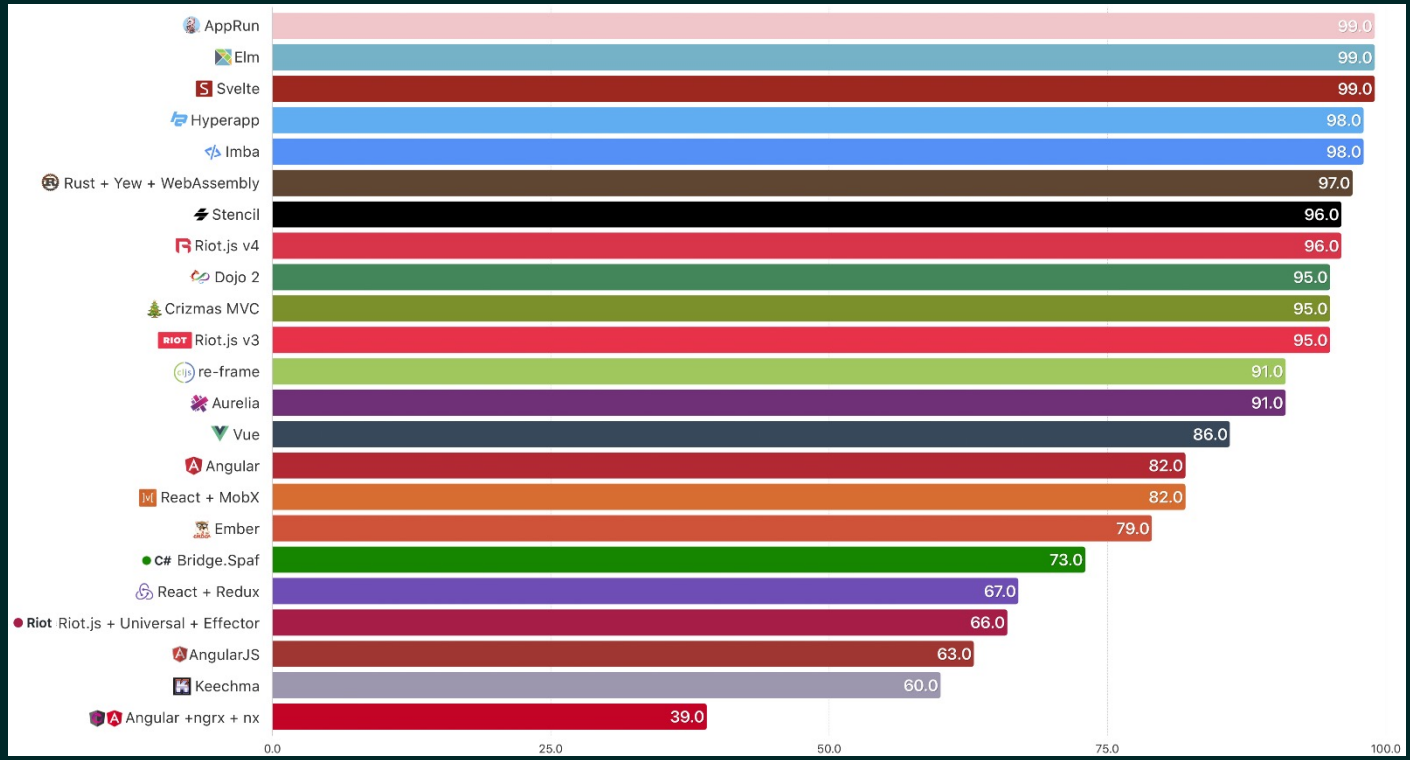
```
subscriptions : Model -> Sub Msg
```

Программы на Elm

```
main : Program Never Model Msg
main =
    Html.program
        { init = init
        , view = view
        , update = update
        , subscriptions = subscriptions
        }
```

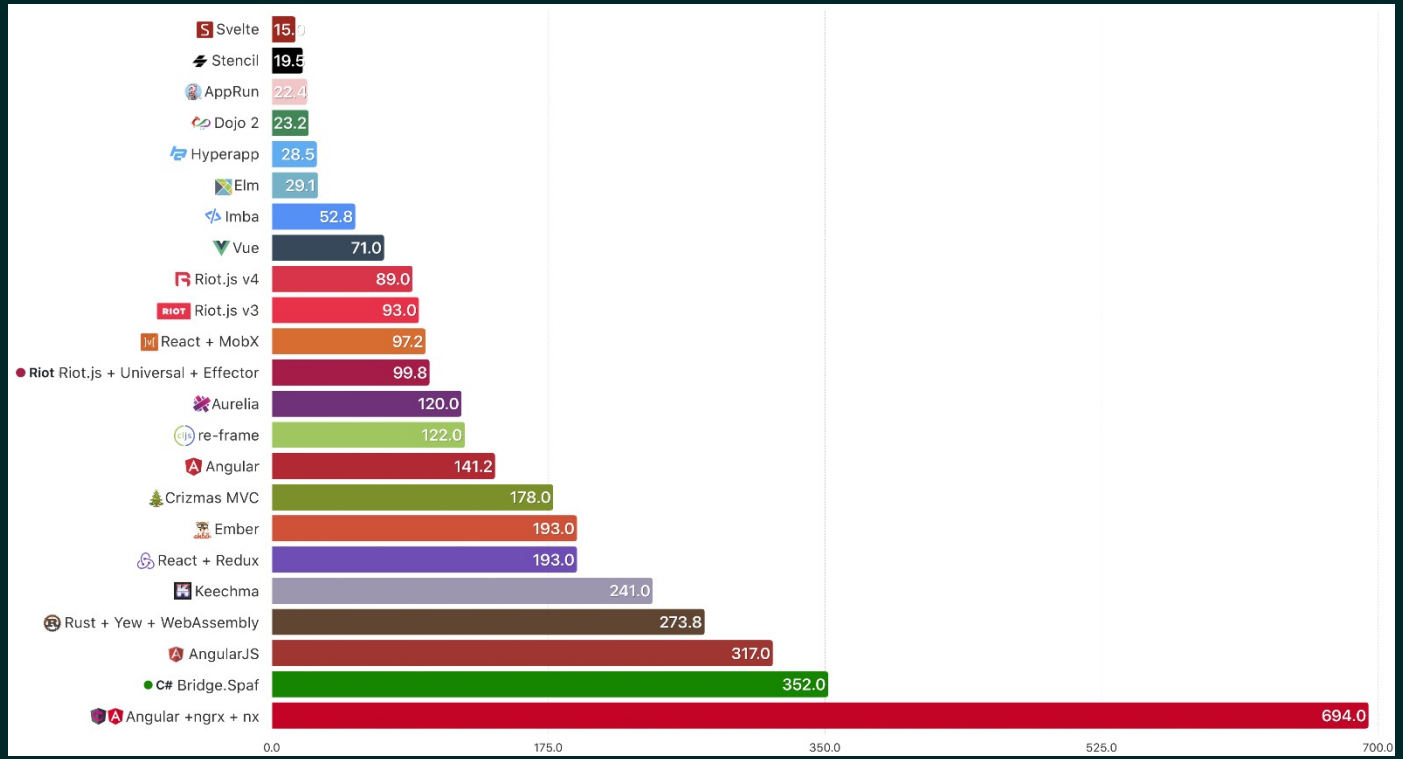
Насколько эффективна генерация JavaScript из других языков?

Быстродействие



https://medium.com/dailyjs/a-realworld-comparison-of-front-end-frameworks-2020-4e50655fe4c1

Объем передаваемого кода (response headers+body)



https://medium.com/dailyjs/a-realworld-comparison-of-front-end-frameworks-2020-4e50655fe4c1

Fable и Fable.Elmish переносят
принципы языка Elm
на платформу.NET

Что такое Fable?

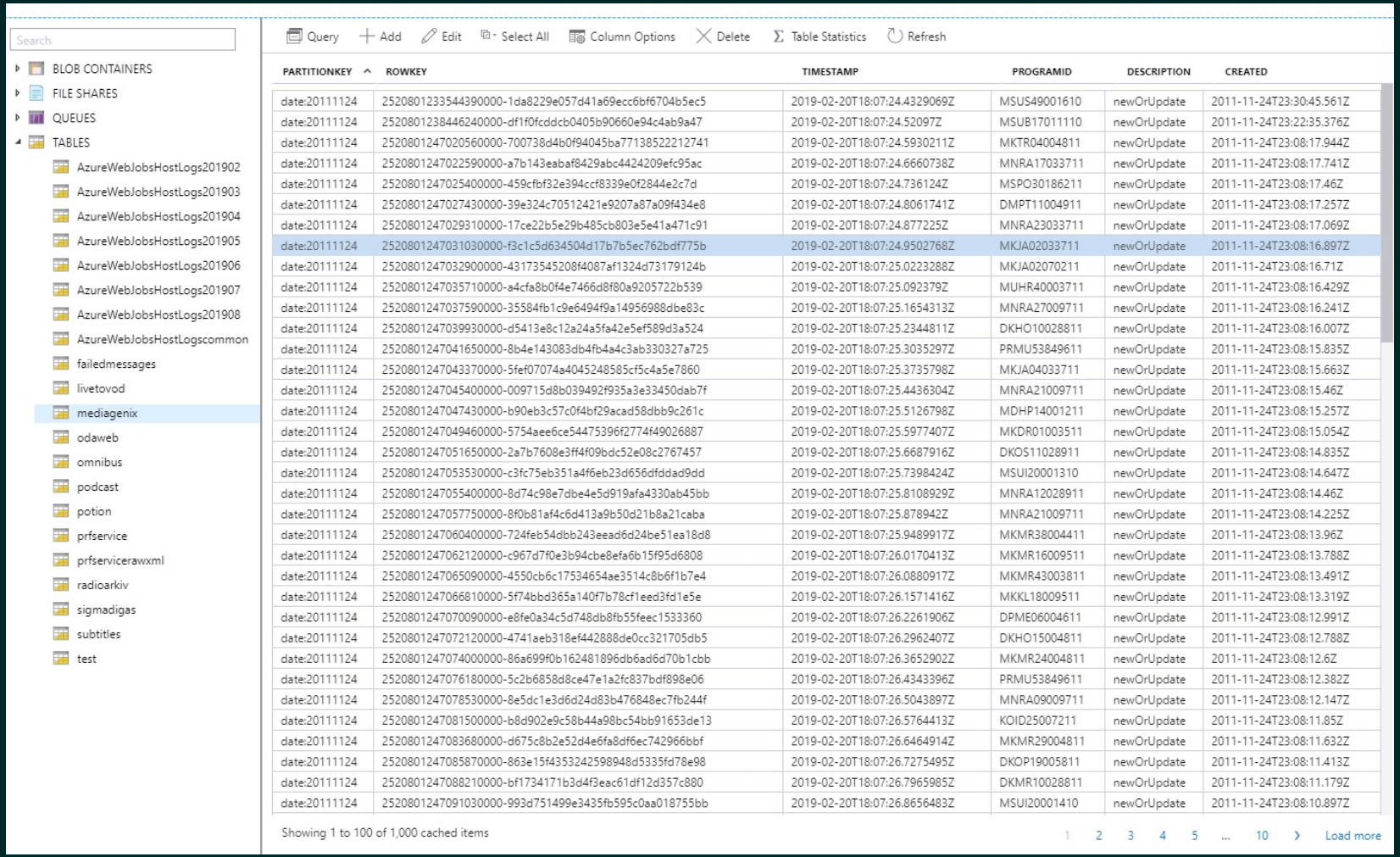
- Fable – это компилятор для программирования на F# в экосистеме JavaScript
- Типы данных неизменяемы по умолчанию
- На выходе получается код на JavaScript
- Статические типы (type safety)
- Простой интероп с компонентами на JavaScript
- Отличные средства разработки (Visual Studio Code, Rider)

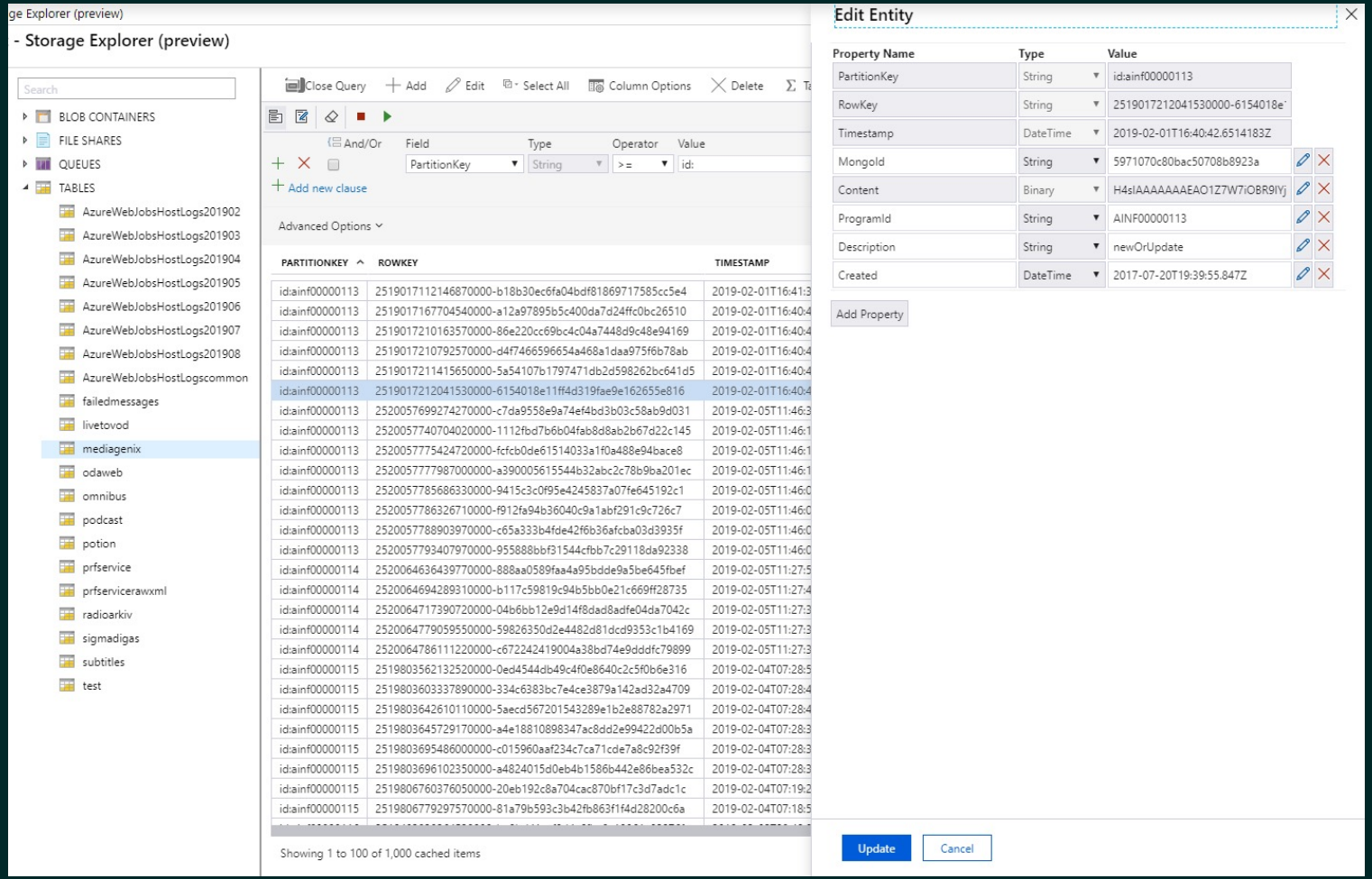
Что *еще* дает Fable?

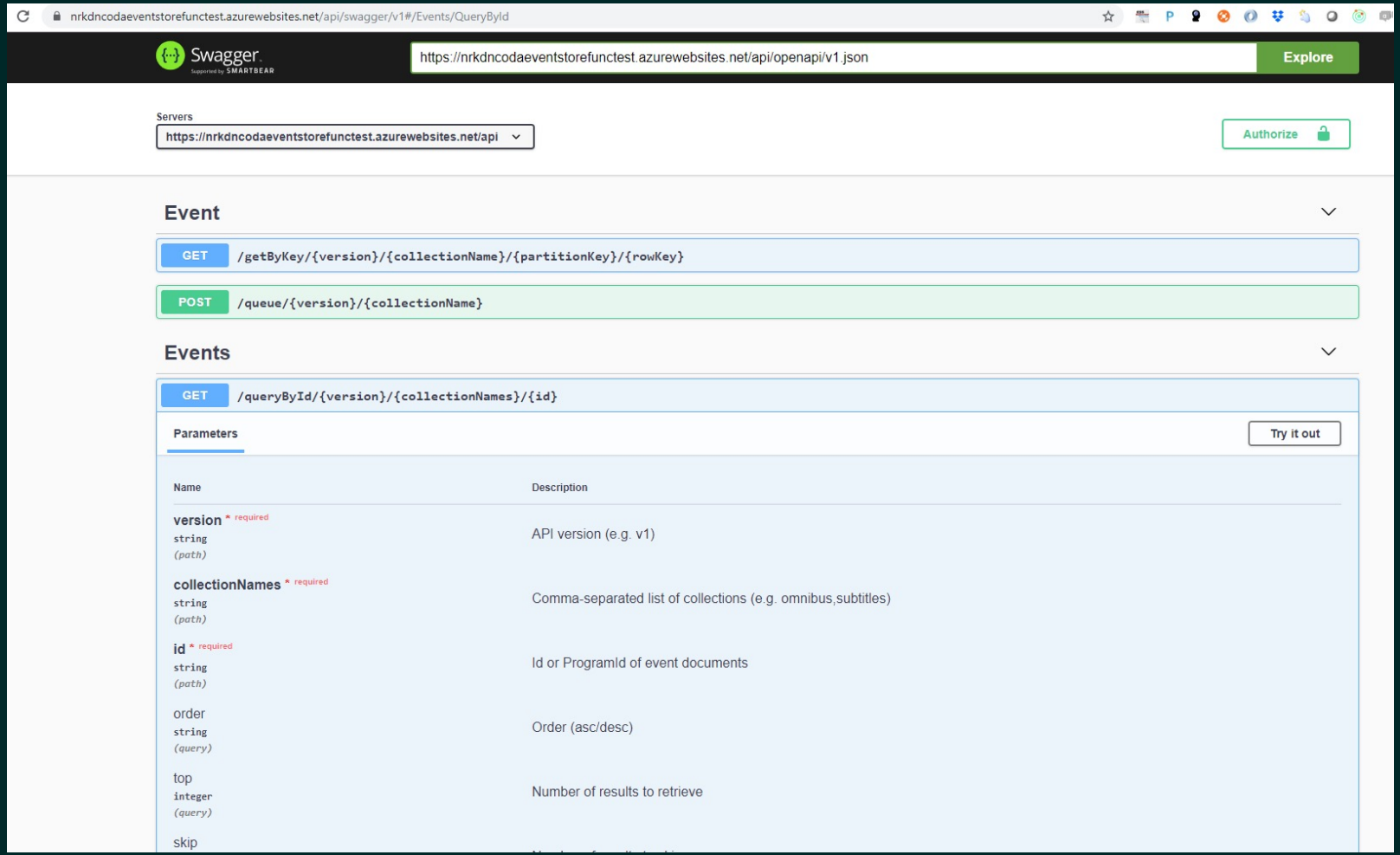
- Fable.React для импорта компонент React
- Вы не обязаны выбирать React, работая с Fable (среди альтернатив – Sutil, реализующий подход Svelte)
- Совместимость с Redux.DevTools (если вы выбрали Fable.React)
- Fable.Elmish поддерживает паттерн MVU
- Библиотеки для импорта CSS (Fulma, Material UI etc.)

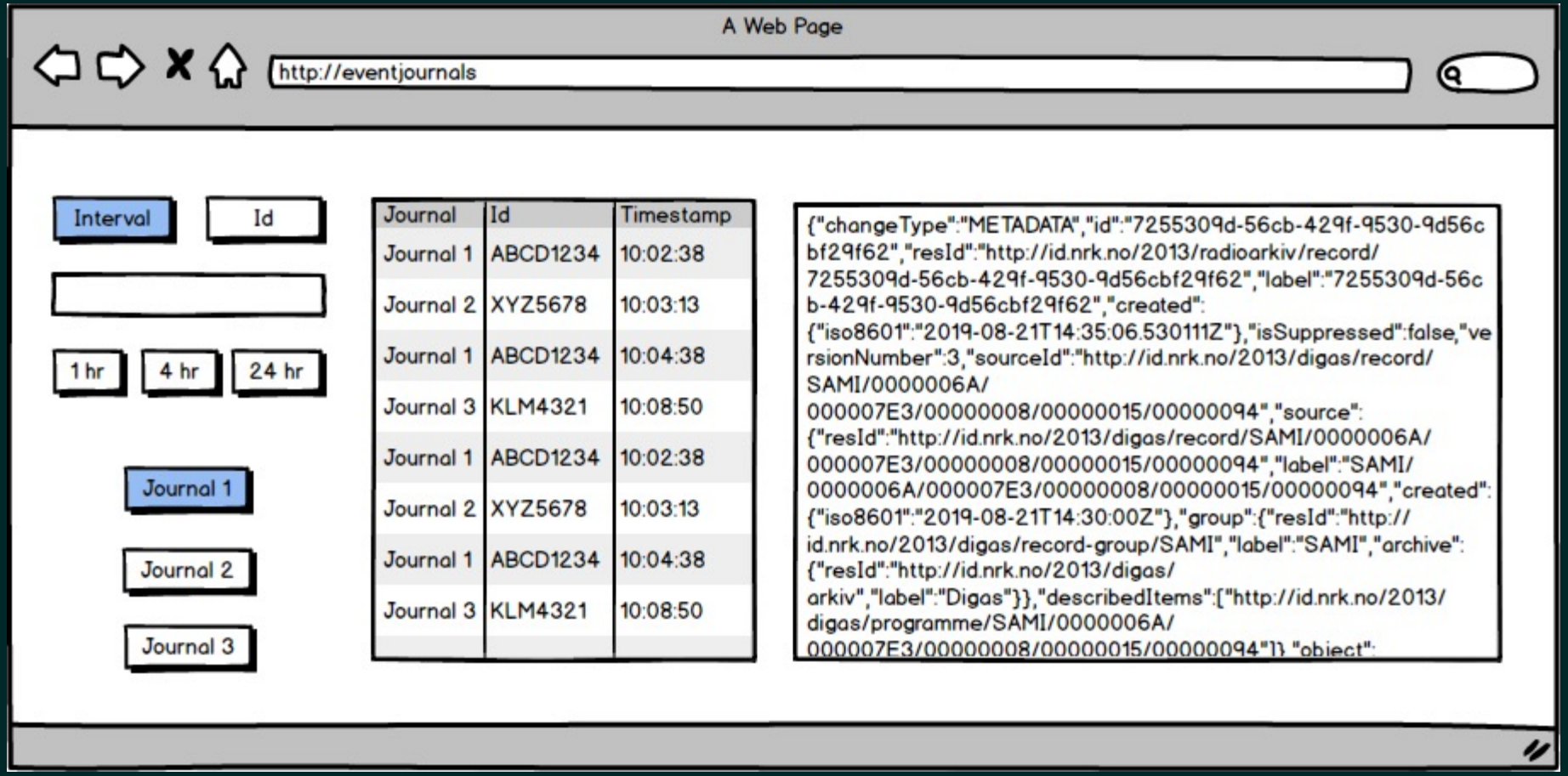
Пример реализации

Фронтенд для журналов событий


Search
BLOB CONTAINERS
FILE SHARES
QUEUES
TABLES
AzureWebJobsHostLogs201902
AzureWebJobsHostLogs201903
AzureWebJobsHostLogs201904
AzureWebJobsHostLogs201905
AzureWebJobsHostLogs201906
AzureWebJobsHostLogs201907
AzureWebJobsHostLogs201908
AzureWebJobsHostLogscommon
failedmessages
livetovod
mediagenix
odaweb
omnibus
podcast
potion
prfservice
prfservicerawxml
radioarkiv
sigmadigas
subtitles
test
Query
Add
Edit
Select All
Column Options
Delete
Table Statistics
Refresh
PARTITIONKEY
ROWKEY
TIMESTAMP
PROGRAMID
DESCRIPTION
CREATED
date:20111124 252080123354439000-1da8229e057d41a69ecc6bf6704b5ec5 2019-02-20T18:07:24.4329069Z MSUS49001610 newOrUpdate 2011-11-24T23:30:45.561Z
date:20111124 252080123844624000-df1f0fcddcb0405b90660e94c4ab9a47 2019-02-20T18:07:24.52097Z MSUB17011110 newOrUpdate 2011-11-24T23:22:35.376Z
date:20111124 252080124702056000-700738d4b0f94045ba77138522212741 2019-02-20T18:07:24.5930211Z MKTR04004811 newOrUpdate 2011-11-24T23:08:17.944Z
date:20111124 252080124702259000-a7b143eabaf8429abc4424209efc95ac 2019-02-20T18:07:24.6660738Z MNRA17033711 newOrUpdate 2011-11-24T23:08:17.741Z
date:20111124 252080124702540000-459cfbf32e394ccf8339e0f2844e2c7d 2019-02-20T18:07:24.736124Z MSPO30186211 newOrUpdate 2011-11-24T23:08:17.46Z
date:20111124 252080124702743000-39e324c70512421e9207a87a09f434e8 2019-02-20T18:07:24.8061741Z DMPT11004911 newOrUpdate 2011-11-24T23:08:17.257Z
date:20111124 252080124702931000-17ce22b5e29b485cb803e5e41a471c91 2019-02-20T18:07:24.877225Z MNRA23033711 newOrUpdate 2011-11-24T23:08:17.069Z
date:20111124 252080124703103000-f3c1c5d634504d17b7b5ec762bdf775b 2019-02-20T18:07:24.9502768Z MKJA02033711 newOrUpdate 2011-11-24T23:08:16.897Z
date:20111124 252080124703290000-43173545208f4087af1324d73179124b 2019-02-20T18:07:25.0223288Z MKJA02070211 newOrUpdate 2011-11-24T23:08:16.71Z
date:20111124 252080124703571000-a4cfa8b0f4e7466d8f80a9205722b539 2019-02-20T18:07:25.092379Z MUHR40003711 newOrUpdate 2011-11-24T23:08:16.429Z
date:20111124 252080124703759000-35584fb1c9e6494f9a14956988dbe83c 2019-02-20T18:07:25.1654313Z MNRA27009711 newOrUpdate 2011-11-24T23:08:16.241Z
date:20111124 252080124703993000-d5413e8c12a24a5fa42e5ef589d3a524 2019-02-20T18:07:25.2344811Z DKHO10028811 newOrUpdate 2011-11-24T23:08:16.007Z
date:20111124 252080124704165000-8b4e143083db4fb4a4c3ab330327a725 2019-02-20T18:07:25.3035297Z PRMU53849611 newOrUpdate 2011-11-24T23:08:15.835Z
date:20111124 252080124704337000-5fef07074a4045248585cf5c4a5e7860 2019-02-20T18:07:25.3735798Z MKJA04033711 newOrUpdate 2011-11-24T23:08:15.663Z
date:20111124 252080124704540000-009715d8b039492f935a3e33450dab7f 2019-02-20T18:07:25.4436304Z MNRA21009711 newOrUpdate 2011-11-24T23:08:15.46Z
date:20111124 252080124704743000-b90eb3c57c0f4bf29acad58dbb9c261c 2019-02-20T18:07:25.5126798Z MDHP14001211 newOrUpdate 2011-11-24T23:08:15.257Z
date:20111124 252080124704946000-5754aee6ce54475396f2774f49026887 2019-02-20T18:07:25.5977407Z MKDR01003511 newOrUpdate 2011-11-24T23:08:15.054Z
date:20111124 252080124705165000-2a7b7608e3ff4f09bdc52e08c2767457 2019-02-20T18:07:25.6687916Z DKOS11028911 newOrUpdate 2011-11-24T23:08:14.835Z
date:20111124 252080124705353000-c3fc75eb351a4f6eb23d656dfddad9dd 2019-02-20T18:07:25.7398424Z MSUI20001310 newOrUpdate 2011-11-24T23:08:14.647Z
date:20111124 252080124705540000-8d74c98e7dbe4e5d919afa4330ab45bb 2019-02-20T18:07:25.8108929Z MNRA12028911 newOrUpdate 2011-11-24T23:08:14.46Z
date:20111124 252080124705775000-8f0b81af4c6d413a9b50d21b8a21caba 2019-02-20T18:07:25.878942Z MNRA21009711 newOrUpdate 2011-11-24T23:08:14.225Z
date:20111124 252080124706040000-724feb54dbb243eead6d24be51ea18d8 2019-02-20T18:07:25.9489917Z MKMR38004411 newOrUpdate 2011-11-24T23:08:13.96Z
date:20111124 252080124706212000-c967d7f0e3b94cbe8efa6b15f95d6808 2019-02-20T18:07:26.0170413Z MKMR16009511 newOrUpdate 2011-11-24T23:08:13.788Z
date:20111124 252080124706509000-4550cb6c17534654ae3514c8b6f1b7e4 2019-02-20T18:07:26.0880917Z MKMR43003811 newOrUpdate 2011-11-24T23:08:13.491Z
date:20111124 252080124706681000-5f74bbd365a140f7b78cf1eed3fd1e5e 2019-02-20T18:07:26.1571416Z MKKL18009511 newOrUpdate 2011-11-24T23:08:13.319Z
date:20111124 252080124707009000-e8fe0a34c5d748db8fb55feec1533360 2019-02-20T18:07:26.2261906Z DPME06004611 newOrUpdate 2011-11-24T23:08:12.991Z
date:20111124 252080124707212000-4741aeb318ef442888de0cc321705db5 2019-02-20T18:07:26.2962407Z DKHO15004811 newOrUpdate 2011-11-24T23:08:12.788Z
date:20111124 252080124707400000-86a699f0b162481896db6ad6d70b1cbb 2019-02-20T18:07:26.3652902Z MKMR24004811 newOrUpdate 2011-11-24T23:08:12.6Z
date:20111124 252080124707618000-5c2b6858d8ce47e1a2fc837bdf898e06 2019-02-20T18:07:26.4343396Z PRMU53849611 newOrUpdate 2011-11-24T23:08:12.382Z
date:20111124 252080124707853000-8e5dc1e3d6d24d83b476848ec7fb244f 2019-02-20T18:07:26.5043897Z MNRA09009711 newOrUpdate 2011-11-24T23:08:12.147Z
date:20111124 252080124708150000-b8d902e9c58b44a98bc54bb91653de13 2019-02-20T18:07:26.5764413Z KOID25007211 newOrUpdate 2011-11-24T23:08:11.85Z
date:20111124 252080124708368000-d675c8b2e52d4e6fa8df6ec742966bbf 2019-02-20T18:07:26.6464914Z MKMR29004811 newOrUpdate 2011-11-24T23:08:11.632Z
date:20111124 252080124708587000-863e15f43532425989480 5335fd78e98 2019-02-20T18:07:26.7275495Z DKOP19005811 newOrUpdate 2011-11-24T23:08:11.413Z
date:20111124 252080124708821000-bf17341 71b3d4f3eac61df12d357c880 2019-02-20T18:07:26.7965985Z DKMR10028811 newOrUpdate 2011-11-24T23:08:11.179Z
date:20111124 252080124709103000-993d751499e3435fb595c0aa018755bb 2019-02-20T18:07:26.8656483Z MSUI20001410 newOrUpdate 2011-11-24T23:08:10.897Z
Showing 1 to 100 of 1,000 cached items
1 2 3 4 5 ... 10
Load more


ge Explorer (preview)
- Storage Explorer (preview)
Search
BLOB CONTAINERS
FILE SHARES
QUEUES
TABLES
AzureWebJobsHostLogs201902
AzureWebJobsHostLogs201903
AzureWebJobsHostLogs201904
AzureWebJobsHostLogs201905
AzureWebJobsHostLogs201906
AzureWebJobsHostLogs201907
AzureWebJobsHostLogs201908
AzureWebJobsHostLogscommon
failedmessages
livetovod
mediagenix
odaweb
omnibus
podcast
potion
prfservice
prfservicerawxml
radioarkiv
sigmadigas
subtitles
test
Close Query
Add
Edit
Select All
Column Options
Delete
And/Or
Field
Type
Operator
Value
PartitionKey
String
>=
id:
Add new clause
Advanced Options
PARTITIONKEY
ROWKEY
TIMESTAMP
id:ainf00000113 2519017112146870000-b18b30ec6fa04bdf81869717585cc5e4 2019-02-01T16:41:3
id:ainf00000113 2519017167704540000-a12a97895b5c400da7d24ffc0bc26510 2019-02-01T16:40:4
id:ainf00000113 2519017210163570000-86e220cc69bc4c04a7448d9c48e94169 2019-02-01T16:40:4
id:ainf00000113 2519017210792570000-d4f7466596654a468a1daa975f6b78ab 2019-02-01T16:40:4
id:ainf00000113 2519017211415650000-5a54107b1797471db2d598262bc641d5 2019-02-01T16:40:4
id:ainf00000113 2519017212041530000-6154018e11ff4d319fae9e162655e816 2019-02-01T16:40:4
id:ainf00000113 2520057699274270000-c7da9558e9a74ef4bd3b03c58ab9d031 2019-02-05T11:46:3
id:ainf00000113 2520057740704020000-1112fbd7b6b04fab8d8ab2b67d22c145 2019-02-05T11:46:1
id:ainf00000113 2520057775424720000-fcfcb0de61514033a1f0a488e94bace8 2019-02-05T11:46:1
id:ainf00000113 2520057777987000000-a390005615544b32abc2c78b9ba201ec 2019-02-05T11:46:1
id:ainf00000113 2520057785686330000-9415c3c0f95e4245837a07fe645192c1 2019-02-05T11:46:0
id:ainf00000113 2520057786326710000-f912fa94b36040c9a1abf291c9c726c7 2019-02-05T11:46:0
id:ainf00000113 2520057788903970000-c65a333b4fde42f6b36afcba03d3935f 2019-02-05T11:46:0
id:ainf00000113 2520057793407970000-955888bbf31544cfbb7c29118da92338 2019-02-05T11:46:0
id:ainf00000114 2520064636439770000-888aa0589faa4a95bdde9a5be645fbef 2019-02-05T11:27:5
id:ainf00000114 2520064694289310000-b117c59819c94b5bb0e21c669ff28735 2019-02-05T11:27:4
id:ainf00000114 2520064717390720000-04b6bb12e9d14f8dad8adfe04da7042c 2019-02-05T11:27:3
id:ainf00000114 2520064779059550000-59826350d2e4482d81dcd9353c1b4169 2019-02-05T11:27:3
id:ainf00000114 2520064786111220000-c672242419004a38bd74e9dddfc79899 2019-02-05T11:27:3
id:ainf00000115 2519803562132520000-0ed4544db49c4f0e8640c2c5f0b6e316 2019-02-04T07:28:5
id:ainf00000115 2519803603337890000-334c6383bc7e4ce3879a142ad32a4709 2019-02-04T07:28:4
id:ainf00000115 2519803642610110000-5aecd567201543289e1b2e88782a2971 2019-02-04T07:28:4
id:ainf00000115 2519803645729170000-a4e18810898347ac8dd2e99422d00b5a 2019-02-04T07:28:3
id:ainf00000115 2519803695486000000-c015960aaf234c7ca71cde7a8c92f39f 2019-02-04T07:28:3
id:ainf00000115 2519803696102350000-a4824015d0eb4b1586b442e86bea532c 2019-02-04T07:28:3
id:ainf00000115 2519806760376050000-20eb192c8a704cac870bf17c3d7adc1c 2019-02-04T07:19:2
id:ainf00000115 2519806779297570000-81a79b593c3b42fb863f1f4d28200c6a 2019-02-04T07:18:5
Showing 1 to 100 of 1,000 cached items
Edit Entity
Property Name
Type
Value
PartitionKey String id:ainf00000113
RowKey String 2519017212041530000-6154018e
Timestamp DateTime 2019-02-01T16:40:42.6514183Z
Mongold String 5971070c80bac50708b8923a
Content Binary H4sIAAAAAAAAEAO1Z7W7iOBR9lYj
ProgramId String AINF00000113
Description String newOrUpdate
Created DateTime 2017-07-20T19:39:55.847Z
Add Property
Update
Cancel


nrkdncodaeventstorefunctest.azurewebsites.net/api/swagger/v1#/Events/QueryById
Swagger
https://nrkdncodaeventstorefunctest.azurewebsites.net/api/openapi/v1.json
Explore
Servers
https://nrkdncodaeventstorefunctest.azurewebsites.net/api
Authorize
Event
GET /getByKey/{version}/{collectionName}/{partitionKey}/{rowKey}
POST /queue/{version}/{collectionName}
Events
GET /queryById/{version}/{collectionNames}/{id}
Parameters
Try it out
Name
Description
version * required
string
(path)
API version (e.g. v1)
collectionNames * required
string
(path)
Comma-separated list of collections (e.g. omnibus,subtitles)
id * required
string
(path)
Id or ProgramId of event documents
order
string
(query)
Order (asc/desc)
top
integer
(query)
Number of results to retrieve
skip


A Web Page
http://eventjournals
Interval
Id
1 hr
4 hr
24 hr
Journal 1
Journal 2
Journal 3
Journal Id Timestamp
Journal 1 ABCD1234 10:02:38
Journal 2 XYZ5678 10:03:13
Journal 1 ABCD1234 10:04:38
Journal 3 KLM4321 10:08:50
Journal 1 ABCD1234 10:02:38
Journal 2 XYZ5678 10:03:13
Journal 1 ABCD1234 10:04:38
Journal 3 KLM4321 10:08:50
{"changeType":"METADATA","id":"7255309d-56cb-429f-9530-9d56cbf29f62","resId":"http://id.nrk.no/2013/radioarkiv/record/7255309d-56cb-429f-9530-9d56cbf29f62","label":"7255309d-56cb-429f-9530-9d56cbf29f62","created":{"iso8601":"2019-08-21T14:35:06.530111Z"},"isSuppressed":false,"versionNumber":3,"sourceId":"http://id.nrk.no/2013/digas/record/SAMI/0000006A/000007E3/00000008/00000015/00000094","source":{"resId":"http://id.nrk.no/2013/digas/record/SAMI/0000006A/000007E3/00000008/00000015/00000094","label":"SAMI/0000006A/000007E3/00000008/00000015/00000094","created":{"iso8601":"2019-08-21T14:30:00Z"},"group":{"resId":"http://id.nrk.no/2013/digas/record-group/SAMI","label":"SAMI","archive":{"resId":"http://id.nrk.no/2013/digas/arkiv","label":"Digas"}},"describedItems":["http://id.nrk.no/2013/digas/programme/SAMI/0000006A/000007E3/00000008/00000015/00000094"]} "object":

Interval | Id/Program

1 hour | 4 hours | 24 hours

Custom interval | Local time

Raw content | max. results

Search

MediaGeniX

Omnibus

Radioarkiv

Sigma/Digas

Subtitles

PRF Metadata

PRF Raw XML

PRF Collections

Podcast

Potion

Live-to-Vod

Transcoding

Failed Messages

Status: Idle

Last search: 73 ms

Transcoding	**NNFA41017021**	NNFA41017021AH	2021-04-03	12:15:12
Transcoding	**NNFA41017021TOLK**	NNFA41017021AT	2021-04-03	12:13:43
Transcoding	**DMTL36170621**	DMTL36170621AH	2021-04-03	11:57:27
Transcoding	**NNFA06040321**	NNFA06040321AH	2021-04-03	06:09:41
Transcoding	**MSUS52040221**	MSUS52040221AA	2021-04-03	01:41:08
Transcoding	**MSUS52040121**	MSUS52040121AA	2021-04-03	01:38:08
Transcoding	**KMTE30001217**	KMTE30001217AA	2021-04-03	01:31:55
Transcoding	**KMTE30001117**	KMTE30001117AA	2021-04-03	01:29:56
Transcoding	**KMTE30001417**	KMTE30001417CA	2021-04-03	01:24:25
Transcoding	**KOID32005420**	KOID32005420AA	2021-04-03	01:23:43
Transcoding	**KOID20004921**	KOID20004921AA	2021-04-03	01:23:06
Transcoding	**KMTE30001317**	KMTE30001317BA	2021-04-03	01:14:24
Transcoding	**KOID20004621**	KOID20004621AA	2021-04-03	01:12:29
Transcoding	**KOIF23008020**	KOIF23008020AA	2021-04-03	01:08:05
Transcoding	**KOID20004821**	KOID20004821AA	2021-04-03	01:06:38
Transcoding	**KOID20004721**	KOID20004721AA	2021-04-03	01:05:23
Transcoding	**KOID30004320**	KOID30004320AA	2021-04-03	00:45:26
Transcoding	**KOID30004220**	KOID30004220AA	2021-04-03	00:44:54
Transcoding	**MUHH51003221**	MUHH51003221AA	2021-04-03	00:35:29
Transcoding	**DMPP21000821**	DMPP21000821AA	2021-04-03	00:35:05
Transcoding	**DKSF45000311**	DKSF45000311AA	2021-04-03	00:27:58
Transcoding	**DMYT21001521**	DMYT21001521AW	2021-04-03	00:21:24
Transcoding	**MUHH51002421**	MUHH51002421AA	2021-04-03	00:20:58
Transcoding	**KMNO10003221**	KMNO10003221AA	2021-04-03	00:10:15
Transcoding	**MSUM41550120**	MSUM41550120AA	2021-04-03	00:05:48
Transcoding	**MSUS64040621**	MSUS64040621AA	2021-04-03	00:05:46
Transcoding	**NNFA23040221**	NNFA23040221AH	2021-04-02	23:33:07
Transcoding	**NNFA41018021**	NNFA41018021AH	2021-04-02	21:36:38
Transcoding	**DVNR04001321TOLK**	DVNR04001321AT	2021-04-02	20:55:20
Transcoding	**DKIN98033021**	DKIN98033021BA	2021-04-02	20:54:50
Transcoding	**MUHU11000721**	MUHU11000721BA	2021-04-02	20:48:34
Transcoding	**NNFA19040221**	NNFA19040221AH	2021-04-02	20:07:48
Transcoding	**MUHU11000521TOLK**	MUHU11000521AT	2021-04-02	20:07:04
Transcoding	**NNFA19040221TOLK**	NNFA19040221AT	2021-04-02	20:03:00


```
{
  "transcodingFinished" : {
    "carrierId" : "KOID20004921AA"
    "files" : [
      0 : {
        "rungName" : "ID180"
        "bitrate" : 288900
        "duration" : "PT50M56S"
        "video" : {
          "dynamicRangeProfile" : "SDR"
          "displayAspectRatio" : "16:9"
          "width" : 320
          "height" : 180
          "frameRate" : 25
        }
        "audio" : {
          "mixdown" : "Stereo"
        }
        "filePath" : "\\felles.ds.nrk.no\nrk\produksjonsdata\odadistribusjon\_incoming\KOID20004921AA_00000000000000075966_ID180.mp4"
      }
      1 : {
        "rungName" : "ID270"
        "bitrate" : 382600
        "duration" : "PT50M56S"
        "video" : {
          "dynamicRangeProfile" : "SDR"
          "displayAspectRatio" : "16:9"
          "width" : 480
          "height" : 270
          "frameRate" : 25
        }
        "audio" : {
          "mixdown" : "Stereo"
        }
        "filePath" : "\\felles.ds.nrk.no\nrk\produksjonsdata\odadistribusjon\_incoming\KOID20004921AA_00000000000000075966_ID270.mp4"
      }
      2 : {
        "rungName" : "ID360"
        "bitrate" : 659200
        "duration" : "PT50M56S"
        "video" : {
          "dynamicRangeProfile" : "SDR"
          "displayAspectRatio" : "16:9"
          "width" : 640
          "height" : 360
          "frameRate" : 25
        }
        "audio" : {
          "mixdown" : "Stereo"
```

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

ПОЕХАЛИ!

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

Создаем и проверяем начальный проект

- https://fable.io/docs/2-steps/setup.html

- dotnet new -i 'Fable.Template::*’
- Создаем каталог для проекта
- dotnet new fable

- npm install
- npm start

Добавляем дополнительные библиотеки

- dotnet add src\App.fsproj package Fable.Elmish.React
- dotnet add src\App.fsproj package Fable.Elmish.Debugger
- dotnet add src\App.fsproj package Fable.Elmish.HMR
- dotnet add src\App.fsproj package Fable.Fetch

Не забываем про JavaScript

```
"dependencies": {
  "react": "^16.7.0",
  "react-dom": "^16.7.0",
  "remotedev": "^0.2.7"
 }
```

После добавления новых пакетов делаем npm install

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

Добавляем код на F#

```
<Compile Include="Model.fs" />
<Compile Include="Messages.fs" />
<Compile Include="UpdateUtils.fs" />
<Compile Include="Update.fs" />
<Compile Include="ViewUtils.fs" />
<Compile Include="HtmlView.fs" />
```

Меняем код вызывающего модуля App.fs

```
module EventStoreViewer.App
open Elmish
open Elmish.Debug
open Elmish.HMR
open Update

Program.mkProgram init update HtmlView.view
|> Program.withDebugger
|> Program.withReactBatched "elmish-app"
|> Program.run
```

Подправляем index.html

```
<!doctype html>*
<html>
<head>
  <title>Event Store Viewer</title>
  <meta http-equiv='Content-Type’
     content='text/html; charset=utf-8’>
  <meta name="viewport«
     content="width=device-width, initial-scale=1">
  <link rel="shortcut icon" href="fable.ico" />
</head>
<body>
  <div id="elmish-app"></div>
  <script src="bundle.js"></script>
</body>
</html>
```

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

Добавляем библиотеку Fulma

- dotnet add src\App.fsproj package Fulma

<Compile Include="MinimalFulmaView.fs" />

Не забываем про JavaScript

npm install bulma
npm install @fortawesome/fontawesome-free
npm install sass
npm install file-loader
npm install css-loader
npm install style-loader
npm install sass-loader (версию 10.x.x)

После добавления новых пакетов делаем npm install

Обновляем webpack.config.js

Добавляем в проект main.scss

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

План разработки

1. Создать начальный проект и посмотреть, что под капотом
2. Создать типы и функции на F# для представления и обновления состояния приложения
3. Создать HTML страницу с элементами и обработчиками, используя Fable.React
4. Проверить минималистскую уродливую Web страницу
5. Добавить F# CSS библиотеку Fulma для придания странице приличного вида
6. Добавить элементы CSS для более гладкого внешнего вида
7. Добавить компоненты React для вывода XML/JSON

Добавляем новые компоненты

npm install react-json-view
npm install react-xml-viewer
npm install prop-types

После добавления новых пакетов делаем npm install

Используем компоненты в коде на F#

```
let inline prettifyJson json : ReactElement =
  ofImport "default" "react-json-view"
    {|
      src = json
      name = false
      enableClipboard = false
      displayObjectSize = false
      displayDataTypes = false
    |}
    []

let inline prettifyXml xml : ReactElement =
  ofImport "default" "react-xml-viewer"
    {| xml = xml |} []
```

Сравниваем код старого и нового приложений

- Event store Web на F# с использованием Fable/Fulma
  - 7 F# файлов
  - 397 строк кода типов и бизнес-логики
  - 302 строк кода презентаций (views) и логики презентаций
- Старое Web приложение на C#, JavaScript и HTML
  - 6 C# файлов (657 строк)
  - 1 HTML файл (80 строк)
  - 3 JavaScript файлов (455 строк)
  - CSS (121 строк)

Дополнительные материалы

- Elm: https://elm-lang.org/
- Fable: https://fable.io/docs/
- The Elmish Book: https://zaid-ajaj.github.io/the-elmish-book/#/
- SAFE Stack: https://safe-stack.github.io/
- Bulma: https://bulma.io/documentation/
- Fulma: https://fulma.github.io/Fulma/

- Дэвид Эванс «Визуализация качества»: https://www.youtube.com/watch?v=jzY_i-QMW8M

- Исходный код презентации: https://github.com/object/EventStoreViewer3

Спасибо!

- Работаю в компании Miles
- Программирую на F# и C#

- Контактная информация:
  - @ooobject
  - vagif.abilov@mail.com