

<https://github.com/Gauthamjm007/NodeJS-MongoDB-Express-Interview-Questions>

Hogyan lehet megkapni a paraméterek neveit express-ben?

Ez a tulajdonság egy objektum, amely a megnevezett útvonal „paraméterekhez” hozzárendelt tulajdonságokat tartalmaz. Például, ha van egy /user/:name útvonal, akkor a „name” tulajdonság req.params.name néven érhető el. Ez az objektum alapértelmezés szerint

```
// GET /user/tj
req.params.name
// => "tj"
```

Hogyan lehet lekérni a query string paramétereit az express segítségével?

A query string az a rész, amely az URL elérési útja után következik, és kérdőjellel kezdődik?

```
?height=6&weight=60
//req.query.height - 6
//req.query.weight - 60
```

Hogyan lehet visszaküldeni a választ expressz használatával?

e parancsok bármelyikét használhatjuk

```
function(req, res) {
    res.send('Hello World!')
}
function(req, res) {
    res.end('Hello World!')
}
function(req, res) {
    res.json({title:'Hello World!'})
}
```

Hogyan állítsuk be a http válasz állapotát az expressz használatával?

használhatjuk a res.status () vagy a res.sendStatus ()

```
res.status(404).send('File not found')

//if sendStatus we no need to write send method , i will pre send a few inbuilt messages upon using that

res.sendStatus(200)
// === res.status(200).send('OK')

res.sendStatus(403)
// === res.status(403).send('Forbidden')

res.sendStatus(404)
// === res.status(404).send('Not Found')

res.sendStatus(500)
// === res.status(500).send('Internal Server Error')
```

Hogyan lehet megváltoztatni a válasz http fejléc értékét?

Bármely HTTP fejléc értéket megváltoztathat a Response.set () használatával:

```
res.set('Content-Type', 'text/html')
res.type('json')
// => 'application/json'

res.type('application/json')
// => 'application/json'

res.type('png')
// => image/png:
```

Hogyan lehet átirányítani más oldalakra szerver oldalon?

Az átirányítások gyakoriak a webfejlesztésben. Átirányítást a `Response.redirect ()` módszerrel hozhat létre:

```
res.redirect('/go-there')
//it can be either a url or a path of file
res.redirect(301, '/go-there')
```

Hogyan lehet biztosítani a fájlletöltést az express segítségével?

Az Express praktikus módszert kínál a fájlok mellékletként történő átvitelére: `Response.download ()`.

Amint a felhasználó elér egy útvonalat, amely fájlt küld ezzel a módszerrel, a böngészők felszólítják a felhasználót a letöltésre.

A `Response.download ()` metódus lehetővé teszi a kéréshez csatolt fájl elküldését, és a böngésző ahelyett, hogy megjelenítené az oldalon, lemezre menti.

```
app.get('/', (req, res) => res.download('./file.pdf'))
```

Hogyan használjuk a `Response.cookie ()` metódust a sütik kezelésére?

A cookie-k kis adatok, amelyeket egy weboldalról küldenek, és a felhasználó webböngészőjében tárolódnak, miközben a felhasználó az adott webhelyet böngészi. Minden alkalommal, amikor a felhasználó visszatölti a webhelyet, a böngésző visszaküldi a tárolt adatokat a webhelyre vagy a szerverre, hogy megkülönböztesse a felhasználó korábbi tevékenységét.

```
res.cookie('username', 'Adam')

This method accepts a third parameter which contains various options:
res.cookie('username', 'Adam', { domain: '.bangalore.com', path: '/administrator', secure: true })

res.cookie('username', 'Adam', { expires: new Date(Date.now() + 900000), httpOnly: true })

//clear cookie
res.clearCookie('username')
```

Value	Description
domain	a cookie domain név

Value	Description
expires	állítsa be a süti lejáratát dátumát. Ha hiányzik, vagy 0, akkor a cookie session-cookie
httpOnly	állítsa be a cookie-t, hogy csak a webszerver érje el. Lásd: HttpOnly
maxAge	állítsa be a lejáratát időt az aktuális időhöz viszonyítva, milliszekundumban kifejezve
path	a sütiút. Alapértelmezés szerint /
secure	Csak a HTTPS sütit jelöli
signed	sütit signed-ra állítja
sameSite	A SameSite értéke

Hogyan lehet kezelni a munkameneteket az expressz használatával?

Az Express-session modult fogjuk használni, amelyet az Express csapat karbantart. Az implementáció után az Ön API-jának vagy webhelyének minden felhasználójának egyedi munkamenetet kap, és ez lehetővé teszi a userstate tárolását. Alapértelmezés szerint az Express kérések szekvenciális és egyetlen kérés sem kapcsolható össze egymással. Nem lehet tudni, hogy ez a kérés olyan kientstől származik-e, amely korábban már végrehajtott egy kérést.

```
const express = require('express')
const session = require('express-session')

const app = express()
app.use(session({
  'secret': '343ji43j4n3jn4jk3n'
}))
```

Minden megoldás a munkamenet azonosítóját egy cookie-ban tárolja, és megőrzi az adatkezelő oldalt. A kliens cookie-ban kapja meg a munkamenet azonosítóját, és minden HTTP-kéréssel együtt elküldi.

Arra a kiszolgálói oldalra hivatkozunk, hogy a munkamenet-azonosítót a helyben tárolt adatokhoz társítsuk.

Hogyan lehet engedélyezni a Cors-t az Expressjs-ben?

A CORS engedélyezéséhez az Express.js fájlban adja hozzá a következő kódot a server.js fájlba:

```
app.all('*', function(req, res, next) {
  res.set('Access-Control-Allow-Origin', '*');
  res.set('Access-Control-Allow-Methods', 'GET, POST, DELETE, PUT');
});
```

```
res.set('Access-Control-Allow-Headers', 'X-Requested-With, Content-Type');
if ('OPTIONS' == req.method) return res.send(200);
next();
});
```

vagy telepíthet egy cors nevű csomagot, a CORS egy node.js csomag egy Connect / Express köztes szoftver biztosítására, amely a CORS engedélyezéséhez használható különféle opciókkal

```
var express = require('express')
var cors = require('cors')
var app = express()

app.use(cors())
```

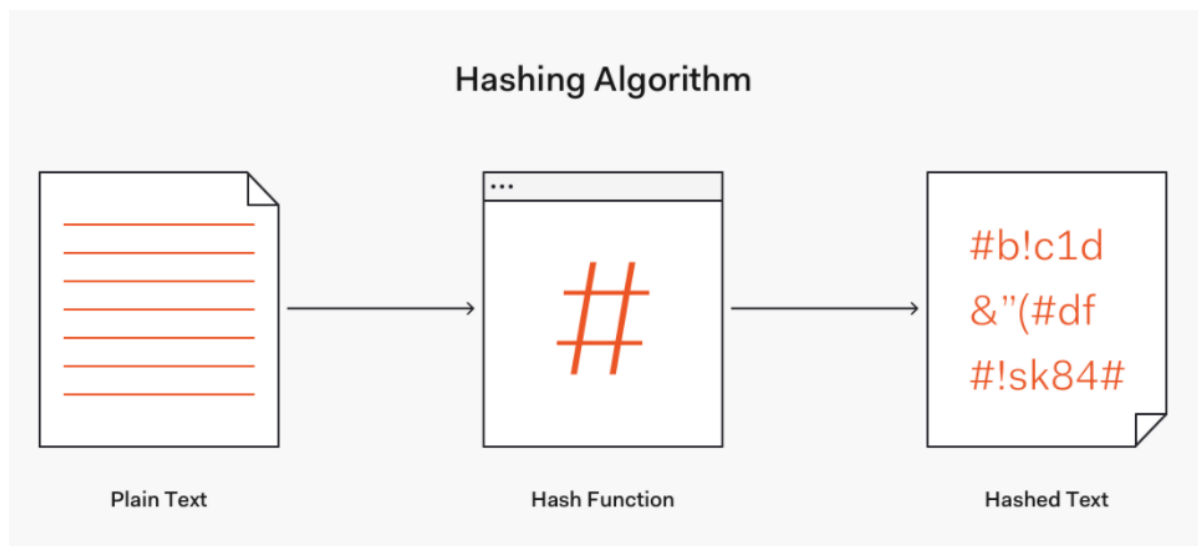
Mi a különbség a NoSQL és az SQL között?

Parameter	SQL	NOSQL
Definition	Az SQL adatbázisokat elsősorban RDBMS vagy Relációs adatbázisoknak hívják	A NoSQL adatbázisokat elsősorban nem relációs vagy elosztott adatbázisnak hívják
Query Language	Strukturált lekérdezési nyelv (SQL)	Nincs deklaratív lekérdezési nyelv
Type	Az SQL adatbázisok tábla alapú adatbázisok	A NoSQL adatbázisok lehetnek dokumentum alapúak, kulcs-érték párok, grafikonok
Schema	Az SQL adatbázisok előre definiált sémával rendelkeznek	A NoSQL adatbázisok dinamikus sémát használnak a strukturálatlan adatokhoz.
Ability to scale	Az SQL adatbázisok vertikálisan skálázhatóak	A NoSQL adatbázisok vízszintesen méretezhetőek
Examples	Oracle, Postgres és MS-SQL.	MongoDB, Redis, Neo4j, Cassandra, Hbase.
Best suited for	Ideális választás a komplex lekérdezésigényes környezethez.	Nem megfelelő komplex lekérdezések.
Hierarchical data storage	Az SQL adatbázisok nem alkalmasak hierarchikus adattárolásra.	Alkalmasabb a hierarchikus adattárházhoz, mivel támogatja a kulcs-érték párosítási módszert
Variations	Egy típus kisebb variációkkal	Sokféle típus, amelyek kulcsérték

Parameter	SQL	NOSQL
Consistency	Erős konzisztencia érdekében kell konfigurálni.	Ez a DBMS
Hardware	Speciális DB hardver (Oracle Exadata stb.)	Commodity hardware
Network	(Infiniband, Fabric Path stb.)	Commodity network (Ethernet, etc.)
Best features	Cross-platform support, Secure and free	Könnyen használható, nagy teljesítményű és rugalmas eszköz.
Top Companies Using	Hootsuite, CircleCI, Gauges	Airbnb, Uber, Kickstarter
ACID vs. BASE Mode	ACID(Atomicity, Consistency, Isolation, and Durability) is a standard for RDBMS	Base (Basically Available, Soft state, Eventually Consistent) is a model of many NoSQL systems

Mi a hash és magyarázza el, hogyan működik?

A hashelés az a folyamat, amelynek során bármilyen hosszúságú bemenetet matematikai függvény segítségével alakítunk át rögzített méretű szöveggé.



Amikor a felhasználó megad egy inputot, akkor egy hash függvény által rögzített hosszúságú értékke konvertálódik, és az így kapott értéket hashed szövegnek hívják, és ennek mindig egyedinek kell lennie a különböző értékeknél

Mik a salt és miért olyan fontosak?

Ez egy egyedi érték, amelyet hozzáadhat a jelszó végéhez, hogy más hash értéket hozzon létre. Ez egy biztonsági szintet ad hozzá a hash folyamathoz

Annyira fontosak, hogy megakadályozzák a durva erőszakos támadásokat (a jelszó minden lehetséges kombinációjának kipróbálása), valamint a rainbow táblát (az összes elterjedt kivonatolt szöveget és azok jelszavát tartalmazó táblázat)

Mi a pepper, és miért olyan fontos?

A pepper egy titok, amelyet hozzáadnak egy bemenethez, például egy jelszóhoz, mielőtt hash-elnek egy cryptographic hash függvényvel

A pepper hasonló szerepet tölt be, mint a salt, de bár a salt nem titkos (csupán egyedi), és a hashed kimenet mellett tárolható

A pepper egy titok, ezért nem szabad a kimenettel tárolni. A hash és a salt általában egy adatbázisban tárolják, de a pepper-t külön kell tárolni (például egy konfigurációs fájlban), hogy megakadályozzák, hogy a támadó megszerezze azt adatbázis-megsértés esetén.

Míg a salt-nak csak elég hosszúnak kell lennie ahhoz, hogy egyedi legyen, a pepper-nek biztonságosnak kell lennie, hogy titokban maradjon (a NIST legalább 112 bitet javasol), ellenkező esetben a támadónak csak egy ismert bejegyzésre van szüksége a pepper feltöréséhez.

Végül a pepper-t újból elő kell állítani minden alkalmazáshoz, amely deploy-olva van, különben az egyik alkalmazás megsértése a másik alkalmazás biztonságának csökkenését eredményezné.

Mi a JWT?

A JSON Web Token (JWT) egy nyílt szabvány (RFC 7519), amely kompakt és önálló módszert határoz meg az információk biztonságos továbbításához a felek között JSON objektumként

néhány olyan eset, ahol a JSON web-tokenek hasznosak:

Authorization: Ez a JWT használatának leggyakoribb forgatókönyve. A felhasználó bejelentkezése után minden további kérelem tartalmazza a JWT-t, amely lehetővé teszi a felhasználó számára, hogy hozzáférjen az adott útvonalakhoz, szolgáltatásokhoz és erőforrásokhoz. Az egyszeri bejelentkezés egy olyan szolgáltatás, amely napjainkban széles körben használja a JWT-t, mivel kicsi az általános költsége és a különböző területeken könnyen használható.

Információcsere: A JSON web tokenek jó módszerek az információk biztonságos továbbítására a felek között. Mivel a JWT-eket alá lehet írni - például nyilvános / privát kulcspárok használatával -, biztos lehet benne, hogy a küldők azok, akiket mondják magukat. Ezenkívül, mivel az aláírást a header és a payload alapján számítják ki, ellenőrizheti, hogy a tartalmat nem hamisították-e meg.

Mivel a session a szerver memóriájában vannak tárolva, a méretezés akkor válik problémává, ha egyszerre hatalmas számú felhasználó használja a rendszert.

A cookie-k általában egyetlen domainen vagy aldomaineken működnek, és általában a böngésző letiltja őket, ha domainek között (cross-domain) működnek (harmadik féltől származó süтик). Problémákat vet fel, amikor az API-kat egy másik tartományból szolgáltatják a mobil és webes eszközök.