| LISTE | CSS SELEKTORI | UTAN deletelle (learnester et l

 | | SJEDNICA (SESSION) | KOLAČIĆI | JAVASCRIPT PODSJETNIK (gradivo MI) |
|--|--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--
--|---|---|---|--|
| = uređena lista, = neuređena lista | oseudo-klase: | HTML podsjetnik: (komentar : komentar Tags

 | EXPRESS (ROUTING) - datoteke tipa NameOfView.routes.js | - slijed vremenski omeđenih I logički povezanih transakcija između | Ograničenje na prosljeđivanje iz drugih domena: | DOM – Document Object Model Glavni objekt DOM-a je document i ima niz podobjekata: |
| <di> = opisna lista, Član liste: naziv Stilizacija člana liste (primjer):</di> | :link :visited :active :hover
:focus :target :lang(xy) :first- | <a> (poveznica)
(novi redak) <button> </button>

 | počinju sa: | pojedinog klijenta I poslužitelja - početak sjednice – određen npr. zahtjevom klijenta prema poslužitelju | SameSite request → First-party cookies (None, Strict, Lax) CrossSite request → Third-party cookies | document.documentElement, predstavlja oznaku <html></html> |
| Man alla hilatraia lista | child :last-child :first-of-type | <div>, <fieldset> (kao form, koristi se</fieldset></div>

 | <pre>const express = require('express'); const router = express.Router();</pre> | nakon duljeg vremena neaktivnosti ili prijavom korisnika etc. | Ograničenja (Nedostaci) kolačića: Vrlo ograničena količina podataka | document.body, predstavlja oznaku <body></body> |
| | :last-of-type :nth-child(i) :nth-last-
child(i) :nth-of-type(i) :only-child | unutar form-a, ima l atribut form) (src, alt) <legend> <h1> , <h2> , <h3> <script></td><td>(možemo I uvesti neke MVC Modele, kao npr:</td><td>- završetak sjednice – određen prestankom rada klijenta ili odjavom etc.</td><td>unutar kolačića (do 4kB), povećanje količina prenošenih podataka,</td><td>može biti null ako stranica nije u potpunosti učitana
document.head. predstavlja oznaku <head></td></tr><tr><td></td><td>:only-of-type :empty :root :enabled</td><td></td><td>const User =</td><td> sjednica ima identifikator – session token, pridijeljen svakoj transakciji
koja pripada određenoj sjednici – session token se nalazi u: </td><td>trajnost podataka unutar kolačića, trajnost podataka unutar kolačića
(privremeni kolačići, istek roka, ručno brisanje,), kolačić je vezan na</td><td>Nad objektima DOM-a moguće je pozivati brojne metode i koristiti</td></tr><tr><td>član koji ima bilokoji id:</td><td> :disabled :checked :not(:first-
child)</td><td>EXPRESS (ROUTING) Middleware</td><td>require('/models/UserModel')) nakon toga moramo implementirati dvije funkcije:</td><td>- prvom retku zaglavlja (dio URI)</td><td>klijenta a ne korisnika (što znači da različiti preglednici ne dijele kolačiće,</td><td>brojna svojstva: .childNodes (svojstvo koje je kolekcija)</td></tr><tr><td></td><td>cnita)
oseudo-elementi:</td><td>- funkcije koje imaju pristup: → request objektu</td><td>1) Funkcija router.get</td><td>- zaglavlju (parovi ime-vrijednost)
- tijelu zahtleva ili odgovoru (resursu)</td><td>a različiti korisnici istog preglednika dijele kolačiće),
a što se tiče Sigurnosti:</td><td>.firstChild (objekt koji je prvo dijete nekog objekta)</td></tr><tr><td>ima klasu "klasa1"</td><td>::first-line ::first-letter ::before </td><td>- odgovara HTTP zahtjevu</td><td>nju obično koristimo za općeniti prikaz stranice,</td><td>ujelu zanijeva ili odgovoru (resursu) Mehanizmi prenošenja podataka o stanju (obilježavanju sjednice):</td><td>informacije unutar kolačića su vidljive I čitljive kod prijenosa nesigurnim</td><td>.lastChild (objekt koji je zadnje dijete nekog objekta)</td></tr><tr><td>ul>li.klasa1{margin-top:0px;}</td><td>::after
kombinatori</td><td> ima svojstva koja odgovaraju query stringu, parametrima,
trilelu, zaglavliima, itd. </td><td>ima elemente req, res, next, te u nju uglavnom
ubacujemo funkciju res.render u kojoj povezujemo</td><td>- Skrivena polja (hidden fields)</td><td>komunikacijskim kanalom, presretanje(krađa) kolačića, Cross-site</td><td>.nextSibling (sljedeći objekt na istoj razini) .previousSibling (prethodni objekt na istoj razini)</td></tr><tr><td></td><td>selector selector → element + descendant</td><td>→ response objektu – odgovara HTTP odgovoru</td><td>elemente iz .ejs-a.</td><td>- Prepisivanje URLa (URL rewriting) - Kolačići (cookies)</td><td>scripting(Session cookie theft, Session hijacking), Cross-site request forgery. Cookie poisoning → third-party cookies su problem privatnosti</td><td>.parentNode (roditeljski objekt)</td></tr><tr><td>h1{background-color: green;}</td><td>selector > selector → element + dijete</td><td> pomoću njega odgovaramo na zahtjev </td><td>Primjer:
datoteka signup.routes.js ima funkciju:</td><td>SKRIVENA POLJA</td><td>STVARANJE SJEDNICA</td><td>.children (elementi djeca) .firstElementChild (prvo djiete element)</td></tr><tr><td>za klase u HTML-u:
ako je u HTML-u <h1 class="klasa1"></td><td>selector + selector → element + element koji</td><td>→ next() funkciji – prosljeđuje obavljanje sljedećoj
middleware funkciji u lancu</td><td>router.get('/', function(req,res,next)</td><td>→ sadržaj HTML stranice uključuje I podatak o stanju/sjednici</td><td> Mogući uzroci nevažeće sjednice: nepostojeći uvid, istekao rok
trajanja sjednice, premašeno vrijeme neaktivnosti sjednice. </td><td>.hrstelementChild (zadnie diiete element)</td></tr><tr><td>u CSS-u je .klasal{border: 1px;}</td><td>selector ~ selector → element + general sibling</td><td>Primjer korištenja:</td><td>{ res.render('signup', {</td><td> dinamička stranica stvara HTML document skriveno polje izgleda ovako: </td><td>neodgovarajuća IP adresa klijenta</td><td>.previousElementSibling (prethodni element na istoj razini)</td></tr><tr><td>za identifikatore:</td><td>HTML FORMS</td><td><pre>const express = require('express'); const app = express(); // ILI</pre></td><td>title: 'Register a new user',
linkActive: 'signup',</td><td><input name="naziv" type="hidden"</td><td>- Tablica sjednica – tablica sjednica koje su trenutno aktivne</td><td>.nextElementSibling (sljedeći element na istoj razini)</td></tr><tr><td>ako je u HTML-u <h2 id="crveno"> u CSS-u je #crveno {border: 1px;}</td><td><u>Sablona:</u>
<form action="X" method="Y" id="Z"></td><td>const router = express.Router();</td><td>user: req.session.user,</td><td>value="SID=34cca95f43d"> Prednosti: - neovisnost o pregledniku, obrasci podržani na svim</td><td>KOMUNIKACIJA S BAZOM PODATAKA - kako tehnički komunicirati sa BP? → treba nam upravljački program,</td><td>document.getElementByld(element id value)</td></tr><tr><td>univerzalni selector (*) = primier:</td><td><pre><label for="username"> Username:</pre></td><td>Middleware - predlošci (ejs)</td><td>err: undefined }); });</td><td>preglednicima, ne mogu se onemogućiti na pregledniku (za razliku od</td><td>driver → najpoznatiji ODBC</td><td>Koristi se id elementa</td></tr><tr><td>* {color: red;} (→ sve je crveno)</td><td><pre><\label> <input type="A" name="B"</pre></td><td>app.set('views', path.join(dirname, 'views'));</td><td><pre>datoteka signup.ejs ima liniju: <% if (err !== undefined) (%></pre></td><td>kolačića), jednostavnost korištenja I performansi</td><td>→ Driver održava bazen (pool) N unaprijed otvorenih konekcija s istim</td><td>document.querySelectorAll(CSS query) i
document.querySelector</td></tr><tr><td>za atribute:</td><td><pre><input type="A" name="B" placeholder="C"></pre></td><td>app.set('view engine', 'ejs');</td><td><pre><%= err %> <% } %></pre></td><td>Nedostaci: - lako dostupni pri pogledu na izvorni kod stranice u
pregledniku, prenose se kod svake transakcije (u oba smjera), zahtijevaju</td><td>korisničkim računom → tzv. CONNECTION POOLING Zašto connection pooling? → Uspostava konekcije je vremenski skupa</td><td>za složenije slučajeve pretraživanja</td></tr><tr><td>[attribute] → sadrži atribut (npr. href, id)
[attribute="x"] → sadrži atribut čija je</td><td><pre><input type="submit" value="Submit"></pre></td><td>Middleware - statički resursi</td><td>err je definiran u routes.js-u, a sintaksa mora biti</td><td>korištenje obrazaca (zbog akcije submit)</td><td>(~30ms), baze podataka ne mogu podnijeti proizvoljan (tj. beskonačan)</td><td>POLJA</td></tr><tr><td>vrijednost jednaka x I samo x</td><td><\form>
X → naziv datoteke (npr index.php) u koju se šalju podac</td><td><pre>app.use(express.static(path.join(dirname, 'public')));</pre></td><td>Ovano Zapidana nako bi ojo Zilao koji error ce na</td><td>PREPISIVANJE URLa</td><td>broj konekcija – optimizirane su one za paralelni rad ALI S MJEROM!,</td><td>let persons = ["Mark", "John", "Joe"] push(items) – dodaje element(e) na kraj</td></tr><tr><td>[attribute~="x"] → sadrži atribut koji u</td><td>la fanna a nationa XIII and talliana accidentis</td><td>Middleware - dekodiranje parametara</td><td>stranici mora izbaciti. 2) Funkcija router.post</td><td> poveznice na stranici uključuju podatak o sjednici mehanizam automatizirane promjene URL dolaznog zahtjeva na </td><td>upiti na jednoj konekciji se obavljaju serijski → → imamo 2 loše opcije: 1) Svaki zahtjev jedna konekcija ILI 2) svi</td><td>pop() - briše i vraća element s kraja</td></tr><tr><td>svojoj listi sadrži x između ostalih vrijednosti</td><td>Y → get ILI post → get – default (šalje podatke iz form-a</td><td>app.use(express.urlencoded({ extended: true</td><td>- smije ih biti više, ovisno o tome koliko mogućih</td><td>poslužitelj Web-a, unutar ulaznog niza filtara</td><td>zahtjevi jedna konekcija – Connection pooling je kompromis između te</td><td>shift() - briše i vraća element s početka
unshift(items) - dodaje element(e) items na početak polja</td></tr><tr><td>[attribute ="x"] → sadrži atribut koji u</td><td>na URL naveden u polju action)
nost – čalja podatka iz form a kao HTTP nost transakciju</td><td>Pokretanje poslužitelja na portu 3000</td><td>redirect-ova ima - definira ponašanje za klik na određeni gumb.</td><td>- mehanizam dodavanja informacija unutar poveznica HTML stranice dostavljene pregledniku</td><td>dvije loše opcije Primjer connection pooling-a u db/index.js:</td><td>unshift(items) - dodaje element(e) items na početak polja
splice(index, num) - od pozicije index briše num elemenata</td></tr><tr><td>svojoj listi sadrzi x između ostalih vrijednosti</td><td>Z → nije nužno imati id, služi za stilizaciju, CSS</td><td>app.listen(3000);</td><td>- delinira ponasanje za klik na određeni gumb. Primjer:</td><td>- podaci: parovi ime-vrijednost</td><td>const {Pool} = require('pg');</td><td>slice(index1, index2) - vraća novo polje s elementima od</td></tr><tr><td>(odvojenih crticom – npr. lang="en-gb")</td><td>A → button, checkbox, text (default), radio, hidden, email.</td><td>5 vrsta Middleware-a:</td><td>u datoteci promotional.ejs imamo ovu liniju:</td><td> alternativan mehanizam označavanja sjednica (kada kolačići nisu </td><td>const pool = new Pool ({</td><td>pozicija indeks1 do index2
concat(items) – nadodaje element(e) items u polje i vraća</td></tr><tr><td>• • • • • • • • • • • • • • • • • • • •</td><td>password, search,
datetime-local, date, etc.
B → ime elementa</td><td>Application-level Middleware Router-level Middleware</td><td><pre><input class="btn" type="submit" formaction="./promotional/save"</pre></td><td>dostupni) Prednosti: - potpuna neovisnost o pregledniku, ne može se neomogućiti</td><td>user: 'postgres',
host: 'localhost',</td><td>kopiju novog polja</td></tr><tr><td>[attribute\$="x"] → završava sa x</td><td>C → tekst koji se pojavljuje dok korisnik još ništa nije</td><td>Error-handling Middleware Third-party Middleware</td><td>value="Save and return to cart"></td><td>na pregledniku, iednostavna implementacija</td><td>database: 'web-lab4',</td><td>indexOf/lastIndexOf(item, pos) - traži element item, počevši s</td></tr><tr><td></td><td>upisao, odnosno input polje nema teksta</td><td>Built-in Middleware (express.static/json/urlencoded)</td><td>u datoteci promotional.routes.js definiramo ponašanje</td><td>Nedostaci: - podaci se prenose unutar polja upita URI, moguće koristiti</td><td>password: 'bazepodataka',</td><td>pozicijom pos
includes(value) – provjerava je li value u polju</td></tr><tr><td>HTTP</td><td>Kako izgleda HTTP odgovor kada korisnik is
HTML formu I submit-a ju? Recimo da imam</td><td>puni PODSJETNIK – gradivo MI o CSS → bodovanje specifičnosti:</td><td>kada korisnik klikne taj gumb:
router.post('/save', function (req, res,</td><td>vrlo ograničenu količinu podataka, podaci dio svih poveznica koje vode
na dinamičke stranice izvorišnog poslužitelja, potrebna dodatna</td><td><pre>port: 5432, }); module.exports = { query: (text, params) => {</pre></td><td>find/filter(func) – vraća sve vrijednosti koje zadovoljavaju</td></tr><tr><td>primjer HTTP zahtjeva:
GET /predmet/rppzw HTTP/1.1</td><td>ovakvu formu:</td><td>- Inline stilovi (1000) (span, itd.),</td><td>next) {</td><td>funkcionalnost kod implementacije, ekstrakcija podataka o sjednici iz</td><td>const start = Date.now();</td><td>uvjet (funkciju) func</td></tr><tr><td>primjer HTTP odgovora:</td><td><pre><form> Ime i prezime: <input type="text"</pre></td><td>- #ID selektori(100), "class i [attribute] i :pseudo-class (10),</td><td><pre>req.session.promotionType = req.body['promo-type'];</pre></td><td>polja upita URI, smanjena čitljivost poveznica</td><td>return pool.query(text, params).then(res => { const duration = Date.now() - start;</td><td>findindex(func) - isto kao i find, ali vraća indeks
forEach(func) - poziva funkciju func za svaki element polja</td></tr><tr><td>HTTP/1.1 200 OK
MIME Type → Multipurpose Internet Mail Exte</td><td></td><td>ciass i [attribute] i :pseudo-class (10),
- <TAG> (osim inline) i ::pseudo-element (1),</td><td>req.session.promotionCode =</td><td>Primjer korištenja: //create a function for adding sessionID to URL</td><td>return res; }); },</td><td>map(func) - zove func za svaki element polja i stvara novo polje</td></tr><tr><td>Struktura: tip/podtip (application/javascript, ir</td><td>mage/png. JMBAG:<input type="text" id="use:</td><td>ID" - <body> i * (0)</td><td>req.body['promo-code'];</td><td>function sessionURLBuilder(sessionID) {</td><td>pool: pool }</td><td>temeljem rezultata te ga vraća</td></tr><tr><td>text/plain, video/mp4)</td><td>name="jmbag"></td><td> ako je težina ista, gleda se poredak I pobjeđuje
zadnii </td><td>res.redirect('/cart'); }) - ovaj primjer konkretno ostvaruje ponašanje da na klik</td><td>//add sessionID parameter to URL query segment
return function(url) {</td><td>Primjer ostvarivanja connection pooling-a u server.js: const app = express();</td><td>→ ove metode ne mijenjaju niz persons, već je njihova return
vrijednost izmijenjeni niz.</td></tr><tr><td> Što možemo provjeriti I mijenjati vezano uz z
odgovore? → HTTP metodu URI kod odgov </td><td>rahtjeve Uzoran student: <input pra, veličina type="checkbox" id="usz">
<in type="checkbox" id="usz">
<in type="checkbox" id="usz">
 </form></td><td>put Relativne I apsolutne jedinice:</td><td>gumba Save, u sjednici se napravi zapis promo tipa I</td><td>return function(url) { let newURL = new URL(url)</td><td>const pg = require('pg')</td><td>sort(func) – sortira polje, te ga vraća</td></tr><tr><td>dostavljenog resursa, vrijeme potrebno za dol</td><td>nvat cype bubilite / 1/202m/</td><td>em → relativna velicina ionta u odnosu na</td><td>promo koda, zatim redirect-a natrag na stranicu cart.</td><td>newURL.searchParams.append(sIDName, sessionID)</td><td>const db = require('./db')</td><td>reverse() – obrće polje, te ga vraća
split()/join() – pretvara string u polje i obmuto</td></tr><tr><td>resursa, zaglavlje zahtjeva I odgovora, cookie</td><td>I da ju je korisnik ispunio na sljedeći način: Ime i prezime: Marko Horvat</td><td>roditelja
rem → relativna veličina fonta u odnosu na root</td><td>Na kraj obavezno staviti:
module.exports = router;</td><td>return newURL.toString() } } COOKIES</td><td><pre>const session = rewuire('express-session') const pgSession =require('connect-pg-simple')(session)</pre></td><td>JSON</td></tr><tr><td>- sigurna verzija protokola HTTP</td><td>JMBAG: 32423423423</td><td>element</td><td>MVC - MODEL</td><td>Mehanizam razmjene male količine slobodno definiranih podataka</td><td>app.use(session({</td><td>U JavaScriptu postoje dvije glavne metode:</td></tr><tr><td>- koristi protocol TLS (port 443), verzija protok</td><td>cola SSL Uzoran student:</td><td>px → apsolutna jedinica, 1 piksel</td><td>- zaduženje: dohvat I manipulacija podacima</td><td>između klijenta I poslužitelja unutar svake transakcije protokola HTTP</td><td>store : new pgSession({ pool: db.pool,
tableName : 'session'}),</td><td>JSON.stringify - pretvara objekte u JSON format JSON.parse - učitava objekte iz JSON formata</td></tr><tr><td> koristi se simetrična kriptografija za šifriranje
se ključ sjednice, nema lažnog predstavljanja </td><td></td><td>→ Primjer korištenja relativnih jedinica rem i em:</td><td>- tipično u suradnji s bazom podataka - JS file</td><td> Ukupna količina sadržaja unutar kolačića je DO 4kB Sadržaj kolačića; jedan par ime=vrijednost </td><td>secret : 'nekitekst',</td><td></td></tr><tr><td> uspostavljanje komunikacije putem TLS-a → </td><td></td><td>(default fs ie 16ny)</td><td></td><td></td><td></td><td>CookieClicker - moduleHighScore.js</td></tr><tr><td></td><td></td><td>(default fs je 16px)
<div style="font-size: 30px;"</td><td></td><td>Meta-podaci (svojstva kolačića) → domena(ako nije definirana onda je</td><td>resave : false,</td><td>export class HighScoreManager {</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunik</td><td>dvije inr=Marka%20Hopyat&imbag=32423442342</td><td><pre>div style="font-size: 30px;"</td><td>- pišemo neke funkcije, definiramo klase, konstruktore
I na kraju to "exportamo" za daljnje korištenje</td><td>Meta-podaci (svojstva kolačića) → domena(ako nije definirana onda je
uzima ime poslužitelja npr. www.fer.unizg.hr), put (domena + put =</td><td>saveUnitialized : true,</td><td>COOMECHICKEY - MODULEHIGNEOUS.S export class HighScoreManager { static getHighScores() { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunik
Opis tijeka komunikacije:</td><td>dvije (acije ipr=Marko%20Horvat&jmbag=32423442342 → u odgovoru se ne nalazi ovo Uzoran stud jer checkbox nema atribut name! da ima pis</td><td><pre>3</td><td>pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer)</td><td>Meta-podaci (svojstva kolačića) → domena(ako nije definirana onda je
uzima ime poslužitelja npr. www.fer.unizg.hr), put (domena + put =
doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost</td><td><pre>saveUnitialized : true, cookie : {maxAge : 60*60*24*1000} // 1 dan})); Kako izvršavati query-e? Jedan primjer kako:</pre></td><td><pre>export class HighScoreManager { static getHighScores() { const highscores = localStorage.getItem("highscores");</pre></td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunik</td><td>odvije ipr=Marko%20Horvat&jmbag=32423442342 odgovoru se ne nalazi ovo Uzoran stud jer checkbox nema atribut name! da ima pis- uzoranstudent=checked</td><td>div style="font-size: 30px;" <pre> <pre> spn. style="font-size: 30px;" </pre> <pre> <pre> spn style="font-size: lobb: spn style="font-size: lrem;">rem</pre> </pre></td><td>- pišemo neke funkcije, definiramo klase, konstruktore
I na kraju to "exportamo" za daljnje korištenje
pomoću: (primjer)
module. exports = (createCart,
addItemToCart)</td><td>Meta-podaci (svojstva kolačića) → domena(ako nije definirana onda je
uzima ime poslužitelja npr. www.fer.unizg.hr), put (domena + put =
doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost
prosljeđivanja, ograničenje na prosljeđivanje iz drugih domena
- kolačiće stvara poslužitelja (definira sadrža) tvojstva kolačića</td><td>saveUnitialized : true, cookie : {maxAge : 60*60*24*1000} // 1 dan})); Kako izvršavati query-e? Jedan primjer kako: (async() => let rows = await db.query('SELECT id,</td><td><pre>export class HighScoreMenager { static geHighScores() { const highscores = localStorage.getItem("highscores"); if (thighscores) {</pre></td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunik
<u>Opis tijeka komunikacije:</u> 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelj u odgovoru dostavlja svoj javni certifikat</td><td>dvije ipr=MarkonScOHorvat&imbag=32423442342 ⇒ u odgovoru se ne nalazi ovo Uzoran stud jer checkbox nema atribut name! da ima pis uzoranstudent=checked HTTP odgovort: Intel®roo_Aputiani_rabitieut Intel®roo_Aput</td><td><pre>div style="font-size: 30px;"</td><td>- pišemo neke funkcije, definiramo klase, konstruktore
I na kraju to "exportamo" za daljnje korištenje
pomoću: (primjer)
module.exports = {
createCart,
addItemToCart)
ili Ovako - module.exports = class User{-}</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je
zuizma ime poslužtelja prv. swof kr.unizg.nh, pul (domena + put =
doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost
prosljedivanja, ograničenje na prosljedivanje iz drugih domena
- kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca
- kolačiće potranjuje kiljent, ako je kolačić istoj imena s istog</td><td><pre>saveUnitialized: true, cookie: (maxAge : 60*60*24*1000) // 1 dan})); Kako izvršavat query-e? Jeddan primjer kako: (async() => let rows = await db.query('SELECT id, name, description</pre></td><td><pre>export class HighScoreManager { static getHighScores() { const highscores = localStorage.getItem("highscores");</pre></td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunik
<u>Opis tijeka komunikacije:</u>
1. Preglednik šalje zahtjev poslužitelju
2. Poslužitelj u odgovoru dostavlja svoj javni
certifikat
3. Preglednik provjerava valjanost javnog klj</td><td>dvije igr=Marko%20Horvat8(imbag~32423442342 3 u odgovoru se ne nalaz ovu Dzoran stude jer checkbox nema atribut nameri da ima pis uzcranstudent-checked HTP odgovort: Uspješno obavljeni zahtjevi: 200 K- uspješno obavljeni zahtjev</td><td>div style="font-size: 30px;" <pre> <pre> spn. style="font-size: 30px;" </pre> <pre> <pre> spn style="font-size: lobb: spn style="font-size: lrem;">rem</pre> </pre></td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer) modula- exports = (createCart, addTeamToCart) III ovako-modula- exports = class User(-) MVC-VLEW</td><td>Meta-podaci (svojstva kolačića) → domena(ako nije definirana onda je
uzima ime poslužitelja npr. www.fer.unizg.hr), put (domena + put =
doseg), rok valjanosta, ograničenje pristupa, ograničenje na sigumost
prosljedivanja, ograničenje na prosljedivanje iz drugih domena
- kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića
- kolačiće potranjuja kiljant, ako je kolačić istog imena si stog
poslužitelja pretendom informacija bit če zamijenjena novom</td><td>saveUnitialized: true, cookie: (maxdge: 60*60*24*1000) // 1 dan})); Kako izwisavati query-e?.Jedan primjer kako: (async) => let rows = await db.query('SELECT id, name, description FROM place_type WHEEE id BETWEEN \$1 AND \$2', [1, 3]);</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); //Add new score for user (and remove old one),</td></tr><tr><td>osnovne faze: Faza ukovanja I Faza komunil
Opis tijeka komunikacije:
1. Preglednik šalje zahljev poslužitelju
2. Poslužitelj u odgovoru dostavlja svoj javni
certifikat
3. Preglednik provjerava valjanost javnog klj
certifikata
4. Ako su certifikat i javni ključ poslužitelja va</td><td>dvije m_makro%20Horvati&imbag~32423442342 acacije Jugovoru se ne nalaž ovo Uzoran stude jer checkbox nema atribut nameł da ima pis uzoranstudent-checked makrowy na ma</td><td>div style="font-size: 30px;" ont one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one one </td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer) sodule. exports = { createCart, addITesmoCact+:} ili ovako - module. exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: przezntacija dostavljenih podataka</td><td>Meta-podací (svojstva kolačíca) → domena(ako nije definirana onda je
uzima ime poslužitelja pr. vemk fer unizg.hr), lot (domena + put =
doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost
prosljedivanja, ograničenje na prosledivanje iz drugih domena
- kolačiće stvara poslužitelj, definira sadrža ji svojstva kolačića
- kolačiće pohranjuje klijjent, ako je kolačić istog imena si stog
poslužitelja prethoda ni formacija bit če zamljenjena novom
- kolačiće šalju i klijent i poslužitelj, zahtljeva ih nitko
Primjer:</td><td>saveUnitialized: true, cookie: (saxAge: 60+60+24+1000) // 1 dan})); Kako [zvfsavai query-e?-Jedan primjer kako: (saync()*) - let rows = await db. query('SELECT id, name, description FROM place_type</td><td><pre>export class HighScoreManager { static getHighScores { const highscores = localStorage.getItem("highscores"); if (!highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list</pre></td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komunil
<i>Opis tileka komunikacije</i>: 1. Preglednik šalije zahtjev poslužitelju 2. Poslužitelj u odgovoru dostavlja svoj javni
certifikati 3. Preglednik provjerava valjanost javnog klj
certifikati 4. Ako su certifikat i javni ključ poslužitelja ve
preglednik genetria jednokratni simetričnik klju</td><td>d vije igr=Markor\$\(\circ{\circ}\)2014ovat\(\circ{\circ}\)kimbar=\$\circ{24244234}{\circ}\) udogrever se ne nelizi vov \(\circ}\)tavid jer checkbox nema airbut namel da ima pis udogrever se nelizi vov \(\circ}\)tavid i u constitutent enche da ima pis udogrever se nelizi vov \(\circ}\)tavid i u constitutent enche da ima pis uda i ud</td><td>div atyle="font-aire: 30px;" <pre> <pre> <pre> </pre> </pre> <pre> <pre</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za dalinje korištenje pomoću: (primjer) module.exports = (createCart, addItcemToCart) ili lovako — module.exports = class User{-} MVC - VIEW - LSI file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim</td><td>Meta-podací (svystva koláčka) → domenajako nije definirana onda je
zuzima ime poslužtelja prv. swm fer. unizg. nh., nul (domena + put =
doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost
prosljedivanja, ograničenje na prosjedivanje iz drugih domena
- kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića
- kolačiće spranjuje kiljent, ako je kolačići stoja ijmana s istog
poslužitelja prethodna informacija bit če zamljenjena novom
- kolačiće Spluji kiljenti į poslužitelj, zahtlijeva in nitko
Primjer:
Set-Cookie: SID=4a57e40d0ee; Path-/nastava;</td><td>saveUnitialized: true, coxóis: (saxAge: 60*60*24*1000) // 1 dan})); Kako [zvřsavatí query-e?-Jedan primjer kako: (saync)(*) - let rows = awalt db.query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem("highscores"); if (!highscores) { throw new Error("Highscores not set!") } return JSGN.paree(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateRiighScore(score, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tileka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelj u odgovoru dostavlja svoj javni certifikati 3. Preglednik provjerava valjanost javnog klj certifikati 4. Ako su certifikat i javni ključ poslužitelja va preglednik genetria jednokratni simetričn klju 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja svajmim ključem poslužitelja</td><td>dvije igredikarko%2046-yatikjimbac=23423442342 sacaje 9- 4) odgovore se ne nelazi ovo Uzoran studi jer checktor nema atribut nameł de ima pis- uzoranstudent=checked HTTP odgovori. uča i 1. 200 (WK – uspješno
obavljen zahtjev: 200 (WK – uspješno obavljen zahtjev: 300 Multiple Cholces 300 Multiple Cholces 316/14/14/14/14/14/14/14/14/14/14/14/14/14/</td><td>div style="font-size: 30px;" <pre> <pre> </pre> <pre> <pre> div style="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> cspan style="font-size: 1 c</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer) module. exports = { createCart, addItentroCart} ili ovako - module. exports = class User{-} MVC - UEW - EJS lile - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manijm promjenama</td><td>Meta-podací (svystva koláčía) → domenajako nije definirana onda je uzima ime poslužtelja prv. zwm fer.unizg.nh., jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost prosljeđivanje iz drugih domena - kolačićo stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvanjuje kiljent, ako je kolačići solaj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačićo šejlu i klijent i poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain=wnw. fer. unizg. hr; Secure: ittponly; Expires=Med., 07 May 2014 10:18:23 OMT</td><td>saveUnitialized: true, cockie: (maxAge: 60+60+24+1000) // 1 dan})); Kako izwšavali queny-e? Jedan primjer kako: (async() >> let rows = await db.query('SELECT id, name, description FROM place type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS: informanje o mogućnostima poslužitelja i komunikacijskim</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Posubzitelju odgovoru dostavlaja svoj javil
certifikat 3. Preglednik provjerava valjanost javnog klj
certifikata 4. Ako su certifikat i javni ključ poslužitelja vz
preglednik generira jednokratni simetrični ključ
5. Preglednik šalje u odgovoru ključ šejednice
javnim ključem poslužitelja 6. Poslužitelja prima i dešfirira ključ sjednice p</td><td>dvije acacije -> u olgovoru se ne nalaž ovo Uzoran stude jer checktox nema atribut nameł da ima pis uzoranstudent-checked HTP Odgovori: Uspješno obavljeni zahtjev: 200 (Nr. uspješno obavljen zahtjev Redirect (preusmjeravanje) zahtjev Redirect (preusmjeravanje) zahtjev Solinica 301 Moved permanentty – resurs je TRAIh premješten na drugi poslužitelj 302 Found – resurs je PRIVREMENO prem na drugi poslužitelj 302 Found – resurs je PRIVREMENO prem</td><td>div atyle="font-size: 30px;" <pre> </pre> <pre> div atyle="font-size: den":"ees/apan> //fs1 <pre> <pre> <pre> <pre> den":"ees/apan> //fs2 den":"ees/apan den"</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer) sodute, exports = { createCart, addItem?coart} sili ovako - module.exports = class User{-} MVC - USEW - LSI Silie - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manijim promjenama - u EJS-u se mogu koristiti varijable navedene u datoted i routes, ha siljedeći način:</td><td>Meta-podaci (svojstva kolačića) → domenajako nije definirana onda je uzima ime poslužinelja pri, vemker unizgi, nh. jud (domena + put i dosegi), rok valjanosti, ograničenje pristupa, ograničenje na sigumost prosljedivanje, ograničenje na prosljedivanje iz drugih domena - kolačića stvara poslužitelj, definira sadrža ji svojstva kolačića - kolačiće potranjuje kiljent, ako je kolačić istog imena s istog poslužitelja prethoda informacija bit će zamljenjena novom - kolačiće šalju i kiljent i poslužitelj, zahtijeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path=/nastava; Domain=mow. £er. unizg. hr; Secure; Httponly; Expires=Wed., 07 May 2014 10:18:21 GMT Sadrža jokačića: SID=4a54c40doee + sadrža jo GSAVEZAN !!!</td><td>saveUnitialized: true, cockie: (maxAge: 60*60*24*1000) // 1 dan})); Kako izwšavati queny-e? Jedan primjer kako: (async() >> let rows = await db.query('SELECT id, name, description FROM place_type WHERE di BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informizanje om opudnostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje sadržaja resursa (najčešća metoda)</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tileka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelj u odgovoru dostavlja svoj javni certifikati 3. Preglednik provjerava valjanost javnog klj certifikati 4. Ako su certifikat i javni ključ poslužitelja va preglednik genetria jednokratni simetričn klju 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja svajmim ključem poslužitelja</td><td>dvije acaje 3 odgoveru se nealizi ovo Uzorazi salazi salazi se je odgoveru se nealizi ovo Uzorazi se ili sili sili se ili se odgoveru se nealizi ovo Uzorazi se ili se il</td><td>div atyle="font-aixe: 30px;" <pre> <pre> <pre> <pre> div atyle="font-aixe:</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) moduže. *apports = { createCart, addIteamToCart} ili lovako - moduže. *apports = class User{.} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoted 'routes, ja ma sijedeći način: sko je u 'routes, ja datoteć način: sko je u 'routes, ja datoteć način:</td><td>Meta-podací (svystva koláčía) → domenajako nije definirana onda je uzima ime poslužtelja prv. zwm fer.unizg.nh., jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost prosljeđivanje iz drugih domena - kolačićo stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvanjuje kiljent, ako je kolačići solaj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačićo šejlu i klijent i poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain=wnw. fer. unizg. hr; Secure: ittponly; Expires=Med., 07 May 2014 10:18:23 OMT</td><td>saveUnitialized: true, cookie: (maxAge: 60+60+24+1000) // 1 dan})); Kako [zvrSavati query-e?-Jedan primjer kako: (async()*) - let rows = awalt db.query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODEHTTP ZAHTJEVA OPTIONS-informiranje o mogućenostima poslužitelja i komunikacijskim zahljevima povezanima resursom GET -za dohvaćanje sodatkoa o resursu (prp. provjera da postoj),</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getHem ("highscores"); if ((highscores) { throw new Error("Highscores not set!") } return JSON parse(highscores); //Add new score for user (and remove old one), and or the score for the list one to the score, user) { const highscores = HighScoreManager.getHighScores(); let foundIntry = highscores (indIndex(oneEntry) = -) neEntry.username == user); if(SoundEntry! = -1) { tif(SoundEntry! = -1) {</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Posužužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja va preglednik generira jednokratni simetrični klju 5. Preglednik šalje u odgovoru ključ sjednice javnim ključen poslužitelja 6. Poslužitelj prima i dešifirira ključ sjednice svog privatnog ključa i šalje u odgovoru ključ šlifiriran samim sobom 7. Ključ sjednice koristi se za daljnje šifiriranj</td><td>dvije sacacje — i<u>nr=Marko%20Horvat8(imbag~324/2344/2344</u>2 au dgovoru se ne nalaž ovo Uzoran stude jer checktox nema atribut namet da ima pis uzoranstudent-checked HTP odgovori: Uspješno obavljeni zahtjev: 200 (Ar uspješno obavljeni zahtjev: 200 (Ar uspješno obavljeni zahtjev: 301 Moved permanently zahtjev: 301 Moved permanently — resurs je TRAIh stifiran omodu i drugi poslužitelj 302 Found — resurs je PRIVREMENO premnejšest na drugi poslužitelj 304 Not Modified — resurs je PRIVREMENO prema oddijednica sakonica sakonic</td><td>div atyle="font-size: 30px;" <pre> <div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> cspan style="font-size: 1 cspan</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) moduže. «sports = { createCart, addIteamToCart} ili lovako - moduže. «sports = class User{.} MVC - VIEW - EJS file - Embedded
JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kad HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoted 'routes, ja mas jeledeći način: sko je u "routes; ša datože navedeno: user : reg. session. user, cart : reg. session. user, cart : reg. session. user,</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja prv. svm fer. unizg. nh. ju (domena + put i doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje kiljent, ako je kolačići soloj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće Spluži i klijenti poslužitelj. zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain=wnw. fer. unizg. hr.; Secure: iHttpOn1y; Expires=#ed., 07 May 2014 10:18:12 OMT Sadržaj kolačića: SID=4a57e40d0ee → sadržaj je OBAVEZAN !!! Putz ako jik kolačić vrijedi: Path*/nastava Domena za koju kolačić vrijedi: Domain=www.fer.uzi.7hr Kolačić se može prosljedivita stom sigurnim kanalima(Secure)</td><td>saveUnitialized: true, cookie: (maxAge: 60+60+24+1000) // 1 dan})); Kako izvfavati queny-e2-Jedan primjer kako: (async() ⇒ let rows = await db.query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', {1, 3}); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS: informiranje o mogućnostima poslužitelja i komunikacijskim zahijevima povezanim s resursom GET - za dohvaćanje sadržaja resursa (najčešća metoda) HEAD - za dohvaćanje podataka o resursu (npr. provjera da postoji, veličina)</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svolj javni
certifikat 3. Preglednik provjerava valjanost javnog kij
certifikata 4. Ako su certifikat i javni ključ poslužitelja vz
preglednik generira jadnokratni simetrični klju-
5. Preglednik šalje u odgovoru ključ sjednice
javnim ključem poslužitelja 6. Poslužitelja priva tiedsfirira ključ sjednice p
svog privatnog ključa i šalje u odgovoru ključ
slifiran samim sobom 7. Ključ sjednice koristi se za daljnje širiranj
zimeđu preglednika i poslužitelja</td><td>dvije in <u>Iner-Marko%20Horvat8(imbag~324/2344/2344</u>2 au dgovoru se ne nalaž ovo Uzoran stude jer checktox nema atribut namet da ima pis uzoranstudent-checked HTP odgovori: Uspješno obavljeni zahtjev: 200 (Mr. uspješno obavljen zahtjev: 200 (Mr. uspješno obavljen zahtjev: 301 Moved permanentty – resurs je TRAIh stifran omenišem na drugi poslužitelji 302 Found – resurs je PRIVREMENO premaješen na drugi poslužitelji 304 Not Modified – resurs je PRIVREMENO premaješen na drugi poslužitelji 304 Not Modified – resurs je protika da smo ga tražili (If-Modified-reserva je false) Pogreška na klijentu:</td><td>div style="font-size: 30px;" <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> cspan style="font-size:</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjet) sodule. exports = { createCart, addIteat7cGart} ill ovako - module. exports = class User{-} MVC - VIEW - EJS filis - Embedded JavaScript Template - EJS filis - Embedded JavaScript Template - Zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristiti varijable navedene u datoted i "routes; na slijedeći način: sko je u "routes; sa datoted nace." user : req. session. user, cart : req. session. cart, onda u "sey noženo plasi šledeće;</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužinelja pri, vemker unižg nit, nij (domena + put i doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost prosjedivanje z krugih domena - kolačića stvara poslužitelj, definira sadržaj I svojstva kolačića - kolačića pohranuju kiljent, ako je kolačić istog imena s istog poslužitelja prethodna informacija bit de zamijenjena novom - kolačiće šalju i kiljenti poslužitelj, zahtijeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path⇒/nastava; Domain=www.fer.unizgi, hr; Secure; Httponly; Expires=Wed, 07 May 2014 10:18:21 GMT Sadržaj kolačića: Sild=4a57e40d0ee > Sadržaj io GBAVEZAN !!! Put za koji kolačić vijedi. Path⇒/nastava Domena za koji kolačić vijedi. Domain=www.fer.unizf.hr Kolačić se može prosljedivati samo sigumim kanalima(Secure) Kolačića se može prosljedivati samo sigumim kanalima(Secure) Kolačića vijedi. benže prispedivati samo sigumim kanalima(Secure)</td><td>asveUnitialised: true, coxic: (asxAge: 60*60*24*1000) // 1 dan))); Kako [zvršavati query-e? Jedan primjer kako: (async()*) - let rows = awatt dbquery('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.le(rows); // opcionalno DIOUS HITE ZAHIJEW OFFINITION - informizanje o mogućnostima poslužitelja i komunikacijskim OFFINITION - informizanje o</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores = HighScores/manager.getHighScores(); let foundEntry = highscores.findIndex(oneEntry => oneEntry.username == user); if(GoundEntry! =-1) { highscores.splice(foundEntry, 1); highscores.splice(</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Posužužitelju odgovoru dostavlja svoj javni
certifikat 3. Preglednik provjerava valjanost javnog klj
certifikat 4. Ako su certifikat i javni ključ poslužitelja va
preglednik generira jednokratni simetitični klju 5. Preglednik šalje u odgovoru ključ sjednice
javnim ključem poslužitelja 6. Poslužitelj prima i dešifirira ključ sjednice
svog privatnog ključa i šalje u odgovoru ključ -
šifiriran samim sobom 7. Ključa sjednice koristi se za daljnje šifiriranj</td><td>dvije carcije dogodije dogodij</td><td>div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> copan style="font-size: 1 copan sty</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraju to "exportamo" za daljnje korištenje pomoću: (primjer) sodule. exports = { createCart, addIteatCoCart, ill ovako. — module. exports = class User{-} MVC - VIEW - LIS fills — Ernbedded JavaScript Template - EJS fills — Ernbedded JavaScript Template - ZaduZenje; prezentacija dostavljenih podataka - sintaksa gotlovo identična kao HTML-ova, s manijm promjenama - u EJS-u se mogu koristiti varijable navedene u datoted i "routes, in a slijedeći način: sko je u "noutes is datoledicina cart, onda u "esp-unoženo pisali sljedeće; - C\$ for (item of object. values (cart. items) } \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime posluželjal prv. svom fer. unizg hr.) nij (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigumost prosljeđivanje z krugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje klijeni, ako je kolačići sloji imena s istog posluživelja prethodna informacija bit če zamljenjena novom - kolačiće šajula i klijeni t poslužicije, zahtijeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path**-nastava; bomain**-www. Fer. unizg hr.; Securue: ilttponīy; set-cookie: SID=4a57e40d0ee; Path**-nastava; bomain**-www. Fer. unizg hr.; Securue: ilttponīy; Securus in several proslužica i securistica subrada s</td><td>asveUnitialized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kako (zvršavati query.e²) Jedan primjer kako: (async()*) Let rows = awalt db.query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HITP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezamina resursom GET - za dohvaćanje podataka o resursu (prp. provjera da postoj, veličina) POST - aktivranje resursa (nr.g. slanje podataka o brazaca) PUT - postavljanje entiteta (prp. promjena podataka - kod REST-a) DELETE - brasanje resursa (nr.o kod REST-a)</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, gettem ("highscores"); if (highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (core, user) {</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Qost tijeka komunikacije</i>: 1. Preglednik šalje zatijev poslužitelju 2. Posužužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja vz preglednik generira jednokratni simetrični ključ. S. Preglednik šalje u odgovoru ključ sjednice
(javnim ključem poslužitelja ve svog privatnog ključa i salje u odgovoru ključ sidnice svog privatnog ključa i salje u odgovoru ključ sidiriran samim sobom. 7. Ključ sjednice koristi se za daljnje šifriranjizmeđu preglednika i poslužitelja identifikacija resursa URI — Uniform Resource Identifier URN — Uniform Resource Identifier URN — Uniform Resource Identifier URN — Uniform Resource Identifier</td><td>dvije acaje po dvagove ze ne nelazi vo u Uzona zadaje po dvagove ze ne nelazi vo u Uzona zadaje u Odgoveru se ne nelazi vo u Uzona zadaje u Odgoveru se ne nelazi vo u Uzona du Propositi u Uzona subenit-re-face de ma pisu uzona u Dragove de manga po u Uzona de manga zadaje nelaziona de manga zadaje nelaziona de manga zadaje po posta zadaje poda ka da semo ga trazili (If-Modifiedu po poda Reguesta pogreška na klijentu: 400 Bad Reguesta pogreška u sintaksi zah 401 Unauthorized – zahljevu nedostaje adoto posta zadaje puta kada semo ga trazili (If-Modifiedu poda Reguesta pogreška na klijentu: 400 Bad Reguesta – pogreška u sintaksi zah 401 Unauthorized – zahljevu nedostaje adoto posta zadaje posta kada semo ga trazili (If-Modifiedu posta) pogreška na klijentu: 400 Bad Reguesta – pogreška u sintaksi zah 401 Unauthorized – zahljevu nedostaje adoto posta zadaje posta kada semo ga trazili (If-Modifiedu posta) pogreška na klijentu: 400 Bad Reguesta – pogreška u sintaksi zah 401 Unauthorized – zahljevu nedostaje adoto posta posta</td><td>div atyle="font-aire: 30px;" <pre> <pre> <pre></td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula- exports = { createCart, addIteamToCart} ili Ovako - modula- exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu korištili varjiable navedene u datoted "routes, ja na sijedeći način: sako je u "routes; ša datote navedeno: user: req.session.user, cart: req.session.user, cart: req.session.user, condru "ejs-u možemo pisali sijedeće:
 < for (tiem of Object.values(cart.items)) { \$>\$ cdiv class" entry" data-id="Cs-item.id</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja prv. svm fer.unizg.nh., jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđivanje iz drugih domena - kolačićo stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo pohranjuje Kiljent, ako je kolačići solaj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačićo šejlu i klijenti poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Pather/nastava; Domain=wnw. fer. unizg. hr; Secure: iRttpOnly; Expires=Med. 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee → sadržaj je OBAVEZAN !!! Putza koj kolačić vrijedi: Path-inastava Domena za koju kolačić vrijedi: Domain=www.fer.uniz1nr Kolačić se ne prosljedival smo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokalno(HtpOnly) Kolačić sietjece: Expires=Wed. (7 May 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju kolačić veljedi: patina za kolačića pripale prosladomeni za koju kolačić selece: Expires=Wed. (7 May 2014 10:18:21</td><td>asveUnitialised: true, coxóte: (asxAge: 60+60+24+1000) // 1 dan))); Kako (zvršavatí query-e²) Jedan primjer kako: (async() » Jet rows = awatt dbquery('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno DIOS HITE ZANTJEW WHONDS: informizanje o mogućnostima poslužitelja i komunikacijskim Orbitologi, informizanje o mogućnostima poslužitelja i</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage.getTeen ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new Score for user (and remove old one), and sort the scores for the list static updatemighScore(score, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni
certifikat 3. Preglednik provjerava valjanost javnog kij
certifikata 4. Ako su certifikat i javni ključ poslužitelja vz
preglednik generira jednokratni simetiřich klju
5. Preglednik šalje u odgovoru ključ sjednice
javnim ključem poslužitelja
6. Poslužitelja prina i desifiria ključ sjednice
svog privatnog ključa i šalje u odgovoru ključ
slifitran samim sobom
7. Ključ sjednice koristi se za daljnje šlifitranj
zimeđu preglednika i poslužija Identifikacija resursa
URI — Uniform Resource Identifier
URI — Uniform Resource Name (npr. um:sef.
URI — Uniform Resource Locator</td><td>dvije cacaje — incr-Marko%20Horvat8jmbag~32423442342 2 u odgovoru se ne nalaž ovo Uzoran stude jer checktox nema arrbut namet de ima pis uzoranstudent-checked HTP Odgovori: Uspješno obavljeni zahtjevi: 200 Kr. uspješno obavljeni zahtjevi: 200 Kr. uspješno obavljeni zahtjevi: 200 Kr. uspješno obavljeni zahtjevi: 301 Moved permanently - resurs je TRAIN sifriran omodu. 3 promodu sijednice a pruka drugi poslužitelj 302 Found - resurs je PRINKEMENO premanently - resurs je prodiciran - do prodiciran - resurs je pravidanje do prodiciran - resurs je pravidan, ali serve do premanently - resurs je pravidanje do premanently - resurs je pra</td><td>div style="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> cspan style="font-size: 1 cspan st</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraju to "exportamo" za dalinje korištenje pomoću: (primjer) moduže. «sports = (rosateCart, addIteamToCart) ili lovako — moduže. «sports = class User{} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, smanjim promjenama - u EJS-u se mogu koristiti varijable navedene u datoteci "routes, sin as ljedeći način: sako je u "routes, še datote routes, oda vi "ejsu moženo" oda vi "ejsu moženo" oda vi "ejsu moženo postal šledeće: <\$ for (item of object.valuse) (\$> cdito class"entry" data-id="Cism id.</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja prv. zwm fer. unizg hr.) nij (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje z krugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje klijeni, ako je kolačići saloj imena s istog posluživelja prethodna informacija bit če zamljenjena novom - kolačiće šaljuži i klijeni ti poslužicij, zahtijeva in nitko Primjer. set-Cookie: SID=4a57e40d0ee; Path=/nastava; Domain=www. fer. unizg. hr.; Secure: ilittponity; sprijeva-iliti i sadržaj kolačića: SID=4a57e40d0ee; Path=/nastava; Posluži klijeni ti posluži da 10:18:12; dett? Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Put za koli kolačića vijedi. Path-inastava w kret unizžir kolačića se može prosljedivati samo sigurnim konalima (Secure) Kolačića se dostava uniziti. 1. Poslužičila pripada domeni za koju je kolačića utjeti sekano rok tajanja ši im unizm puta dohvadenog resura, s. Kolačića unje istekao rok tajanja ši im unizm puta dohvadenog resura, s. Kolačića unje istekao rok tajanja ši im unizm</td><td>asveUnitialised: Erus, coxóic: (asaXpe: 60+60+24+1000) // 1 dan})); Kako izvásvati query-e/ Jodan primjer kako: (async(): > let rows = awatt dbquery('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows): // opcionalno DIO: DIO: METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogudonostima poslužitelja i komunikacijskim zahlevima povezamina resursom GET - 2a dohvačanje sodatka o resursu (npr. provjera da postoji, HEDD - 2a dohvačanje podatka o resursu (npr. provjera da postoji, POST. aktiviranje resursa (npr. slanje podataka obrazaca) PUT - postavjenje entileta (npr. provjera da postoji, RTAGE: 2a dilgnostiku CONNECT - pokreće dvosmjernu komunikaciju s traženim resursom putem prove, i perma web-sleditu koje korist HTTPS</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores) return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni
certifikat 3. Preglednik provjerava valjanost javnog klj
certifikata 4. Ako su certifikat i javni ključ poslužitelja vz
preglednik generira jednokratni simetrični klju-
5. Preglednik šalje u odgovoru ključ sjednice
javnim ključem poslužitelja 6. Poslužitelja prima i desifirira ključ sjednice
svog privatnog ključa i šalje u odgovoru ključ
slifiran samim sobom 7. Ključ sjednice koristi se za daljnje šlifiranji
zimeđu preglednika i poslužija identifikacija resursa
URI — Uniform Resource Identifier
URI — Uniform Resource locator
Analiza URI-ja HTTP-g:
http://www.fer.unizg.hr/predmet/ppzw</td><td>dvije acaje po dvaje acaje po dvaje acaje po dvaje acaje po dvaje acaje po dvajeva se ne nelazi vo u Uzaras Ujeva i u odavani po dvajeva se ne nelazi vo u Uzaras Ujeva i u odavani po dvajeva i u odavani u o</td><td>div atyle="font-size: 30px;" div atyle="font-size: 30px;" den":>emc/span> //fs1 den":>emc/span> //fs2 den": d</td><td>- pišemo
neke funkcije, definiramo klase, konstruktore Ina kraju to "exportamo" za dalinje korištenje pomoću: (primjer) moduže. «sports = (createCart, addIteamCCart) ili lovako — moduže. «sports = class User{} MVC-VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, smanjim promjenama - u EJS-u se mogu koristiti varijable navedene u datoteci "routes ja na sijedeci način: sako je u "routes ja datavica moderno user : req. session. user, conder "eq. session. user, conder</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime posluželjali prv. wme frunizgi nij. nij (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje klijeni, ako je kolačići saloj imena s istog posluživelja prethodna informacija bit če zamljenjena novom - kolačiće šalju i klijeni ti poslužicij, zahtijeva in nitiko Primjer. sat-cookie: SID=4a57e40d0ee; Path=/nastava; Domain=www. Zer. uni zg. hz; Secure: ilttponiy; sat-tookie: SID=4a57e40d0ee; Path=/nastava; Domain=www. Zer. uni zg. hz; Secure: ilttponiy; Satrizes-Med. 07 May 2014 10:18:12: datr; Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Put za koli kolačić vijed: Path-mastava w Meter unizfir Domena za koju kolačić vijed: Path-mastava v Meter unizfir Nošlačiću se ne može pristupsti seboslačiću silječne. Evpires-Ved. 07 May 2014 10:18:21 Prosljeđivanje kolačića: UVIET. 1. Poslužičelj pripada domeni za koju je kolačić defeiriran. 2. Put definiran za kolačiću jejelinje padadržan unutar definiran rok trajanja ili. Acidačić koji imija definiran osvojstvo secure mogu bili prospediovani samo sigurnim kanalina (thes, ne http). S. Samo mogu liži prospediovani samo sigurnim kanalina (thes, ne http). S. Samo mogu bili prospediovani samo sigurnim kanalina (thes, ne http). S. Samo mogu bili prospediovani samo sigurnim kanalina (thes, ne http). S. Samo</td><td>asveUnitialized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kako (zvršavati query.e²) Jedan primjer kako: (async()*) Let rows = awalt db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje sodatako o resursu (npc. provjera da postoj, veličina) POST - aktivanje resursa (npc. slanje podataka obrazaca) PUT - postavljanje entitela (npc. promjena podataka - kod REST-a) DELETE - brisanje resursa (npc. slanje podataka - kod REST-a) DELETE - brisanje resursa (npc. kod REST-a) TRACE - za dijagnostiku CONNECT - pokreće dvosnjemu komunikaciju s traženim resursom putem prosv-ja prema web-sjedištu koje koristi HTTPS Razlika izmedu GET-a i HEAD-a:</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, gettem ("highscores"); if (!highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new Score for user (and remove old one), and sort the scores for the list static updatemighscore(score, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Opis tileka komunikacije: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja va preglednik generira jednokratni simetričn klju- 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i destifirak ključ sjednice javnim ključem poslužitelja 5. Poslužitelja 6. Poslužitelja 7. Ključ sjednice kompo polužitelja identifikacija ce kljem poslužitelja ključitelja ključ</td><td>dvije acaje po dvaje acaje po dvaje acaje po dvaje acaje po dvaje acaje po dvajeva se ne nelazi vo u Uzorazi po dvajeva se ne nelazi vo u Uzorazi po dvajeva se nelazi vo u Uzorazi u se nelazi vo u Septimo ne nelazi vo u se nelazi vi se nelazi vi se nelazi vi se nelazi vi se n</td><td>div style="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> cspan style="font-size: 1 cyles" cyles" HTML Some span style="font-size: 1 cspan style="</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) module.exports = (createCart, addIteamToCart) ili Ovako - module.exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoteci "routes ja na sijedeći način: ako je u "routes, is datote navedeno: user : req.session.user, cart : req.session.user, cart : req.session.cart, onda u "ejs-u možemo pisati sjedeće: <\$ for (tiem of Object.values(cart.items)) { \$>\$ (div class" embry "data-id="cs" item.id \$>" () lako dalje <ddiv><= item.price \$></div><di>√div><= item.price \$></div></td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja prv. zwm fer. unizg. rh., jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranjuje klijent, ako je kolačići stoja (jima si sistog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće splu i klijenti poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Pather/nastava; Domain=wnw. fer. unizg. hr; Secure: iRttpOn1y; Expires=-Med. 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putza koji kolačić vrijedi: Path-inastava Domena za koju kolačić vrijedi: Pomain=www.fer.uniz/hr Kolačić se nebe prosljedival smo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolačiću se ne se zavenima se pristupati kokalno(HttpOnly) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolači</td><td>asveUnitialized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kako (zvršavati query.e²) Jedan primjer kako: (async()*) Let rows = awalt db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje sodatako o resursu (prp. provjera da postoj, veličina) POST - aktivanje resursa (nrp. slanje podataka obrazaca) PUT - postavljanje entitela (prp. promjena podataka - kod REST-a) DELETE - brisanje resursa (nrp. slanje podataka - kod REST-a) DELETE - brisanje resursa (nrp. skod REST-a) DELETE - brisanje resursa (nrp. skod REST-a) DELETE - brisanje resursa (nrp. skod REST-a) TRACE - za dijagnostiku CONNECT - pokreće dvosnjemu komunikaciju s traženim resursom putem proxy-ja prema web-sjedištu koje koristi HTTPS Razlika izmedu GET-a i HEAD-s - Veličina - GET je puno veći - Zaglavlja - letenična su im</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, gettem ("highscores"); if (highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); l/Add new Score for user (and remove old one), and sort the scores for the list static updatedlighScores(core, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni
certifikat 3. Preglednik provjerava valjanost javnog klj
certifikata 4. Ako su certifikat i javni ključ poslužitelja vz
preglednik generira jednokratni simetrični klju-
5. Preglednik šalje u odgovoru ključ sjednice
javnim ključem poslužitelja 6. Poslužitelja prima i desifirira ključ sjednice
svog privatnog ključa i šalje u odgovoru ključ
slifiran samim sobom 7. Ključ sjednice koristi se za daljnje šlifiranji
zimeđu preglednika i poslužija identifikacija resursa
URI — Uniform Resource Identifier
URI — Uniform Resource locator
Analiza URI-ja HTTP-g:
http://www.fer.unizg.hr/predmet/ppzw</td><td>dvije com da sie de sie</td><td>div style="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> cspan style="font-size: 1 cspan sty</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraja to "exportamo" za daljnje korištenje pomoću: (primjer) module. exports = { createCart, addIteamToCart} ili Ovako - module. exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoteci "routes ja na sijedeći način: ako je u "routes js databete navedeno: user: req.session.user, cart: req.session.cart, onda u "eis-u možemo pisali siledeće: <\$ for (tiem of Object.values (cart.items)) { \$>\$ cdit class" embry data-id="cs" item.id \$>>" () I tako dalje <ddiv>/div><a href="data" tie</td><td>Meta-podací (svystva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg. rh. ju (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđivanje iz drugih domena - kolačićo stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvarajuje klijent, ako je kolačići solaj imena s
istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačićo spluti viljeni poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain=www.fer. unizg. hr.; Secure: iRttpOn1y; Expires=#ed., 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putz a koji kolačić vrijedi: Parli-mastava Domena za koju kolačić vrijedi: Domain=www.fer. unizž. hr. Kolačić se nete prosljedival samo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolačić stječe: Expires=Wed. O' 70 May 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić defirinar, 2. Put definiran za kolačić u jelni je sadržan unutar puta dohvaćenog resursa, 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji manja definiran ok vignaja, 4. Kolačić koji manja definiran ok trajanja, 4. Kolačić koji minja definiran os vigtovo secure mogu biti prosljedivani samo sigurnim kanalima (https., ne http), 6. Samo ako je to izravno zabranjeno, kolačić neće biti proslijeden iz druge domene. **Trajnost kolačića: Trajni (persisten) kolačić (imaju definirano vvijemo evijemo edinirano definirano okrajova ozbranjeno, kolačić (imaju definirano vijemo edinirano okrajova ozbranjeno, kolačić (imaju definirano vvijemo edinirano koje kolačića: Trajni (persisten) kolačića (imaju definirano vvijemo edinirano svojetova edinirano svojetova edinirano kojetova edinirano koje</td><td>asveUnitialised: Erus, coxós: (asxAge: 60+60+24+1000) // 1 dan})); Kako izvásvati query-e² Jodan primjer kako: (async(): > let rows = awatt dbquery('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows): // opcionalno METODE HTTP ZAHTJEVA OPTIONS: informiranje o mogudonostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje podataka o resursu (npr. provjera da postoji, velična) UTI: poztavljenje relutisel (npr. slanje podataka obrazaca) UTI: poztavljenje relutisel (npr. vonjena podataka e Korazoca) DELETE: brisanje resursa (npr. kod REST-a) Velična—GET pokreće dvosmjernu komunikaciju s traženim resursom putem prov.)- prema web-sjeditu koje koristi HTTPS Razilka između GET-a i HEAD-a: - Veličina—GET je puno već - Zaglavlja – identična su im -</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores) return JSON.parse(highscores); } //Add new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (score, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tijeka komunikacije</i>: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja vz preglednik generira jednokratni simetrični ključ 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja ci privatnog ključa i šalje u odgovoru ključ sjednice javnim ključem poslužitelja ci privatnog ključa i šalje u odgovoru ključ sidiriran simm sobom 7. Ključ sjednice koristi se za daljnje šifriranji zimeđu preglednika i poslužitelja identifikacija resursa URI – Uniform Resource Identifier URN – Uniform Resource Locator Analiza URI-ja HTTP-g: 1. Numk erunizg Irr / predmet/rppzw 1. http // www.fer.unizg Irr / predmet/rppzw 1. http // shema 2. www.fer.unizg Irr / praziv poslužitelja (host 3. predmet/rppzw → put I oznaka resursa Opća sintaksa URI-ja</td><td>dvije care da since d</td><td>div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size:</td><td>- pišemo neke funkcije, definiramo klase, konstruktore in a kraju to "exportamo" za dalinje korištenje pomoću: (primjer) module. exports = (createCart, addIteamToCart) ili ovako — module. exports = class User{-} MVC-VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sinitaksa gotovo identična kao HTML-ova, smanjim promjenama - u EJS-u se mogu koristiti varijable navedene u datoted "routes ja na sijedeci način: sako je u "routes ja data toda način: sako je u "routes" (sako ja sako j</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime posituželnia prv. swnfe-runizgi, nh. jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosjedivanje, iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće sprvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće sprvarajuje kiljeni, ako je kolačići solaj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće šalju i kiljeni ti poslužilej, zahtljeva in nitiko Primjer. Set-Cookie: SID=4a57e40d0ee; Path=/nastava; Domain=wnw. čer. uni zg. hr.; Secure: išttponīy; Explice=Medaci or jedici Path-inastava Domena za koju kolačić vnjedi: Path-inastava Domena za koju kolačić vnjedi: Path-inastava Domena za koju kolačić vnjedi: Domain=www. fer. uni zl. hr.; Kolačić se ne prosjedivala smo sigurnim knaalima (Secure) Kolačiću se ne može pristupati lotalno(HipOnly) Kolačiću se ne može pristupati lotalno(HipOnly) Kolačiću se ne može pristupati lotalno(HipOnly) Kolačiće fefiniran, 2. Put definiran za kolačić u jeleni je sadržan unutam definiran rok trajanja ila rutam definiran os vojstvo secure mogu biti prosdjedivani samo sigurnim knaalima (Hote, ne http). S. Samo ako je to izravno zabranjeno, kolačić neće biti prosiljeden iz druge domene. Trajnost kolačića: Trajni (persisten) kolačić inje isteka or krajnina ila mesento niča siteka valjanosti) Prevmenni ((ranišent).</td><td>asveUnitialized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kako izvršavati query-e? Jedan primjer kako: (async()*) - let rows = awati db. query ('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje podataka o resursu (prp. provjera da postoji, veličina) POST - aktivranje resursa (ng. slanje podataka o brazaca) PUT - postavljanje entiteta (prp. promjena podataka - kod REST-a) DELETE - brisanje resursa (ngr. kod REST-a) DELETE - brisanje resursa (ngr. kod REST-a) TRACE - za dijagnostiku CONNECT - potreće dvosnjemu komunikaciju s traženim resursom putem proxy-ja prema web-sjedištu koje koristi HTTPS Razlika izmedu GET-a i HEAD-s Veličina - GET je puno veći - Zaglavlja - identična su im - Response - HEAD nema response</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores) return JSON.parse(highscores); } //Add new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores = HighScores, getHighScores(); let foundEntry = highscores, findIndex(oneEntry => oneEntry, username = user); if(foundEntry != -1) { highscores.splice(foundEntry, 1); highscores.</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Opis tileka komunikacije: 1. Preglednik šalje zahtjev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja ve preglednik generira jednokratni simetričn ključ 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i destifirak ključ sjednice javnim ključem poslužitelja 6. Poslužitelje prima i destifirak ključ sjednice javnim ključem poslužitelja 7. Ključ sjednice koristi se za daljnje šifiranj tranedu preglednika i poslužitelja URI — Uniform Resource Identifier URI — Uniform Resource Identifier URI — Uniform Resource Locator Analiza URI±a HTTP-a: http://www.fer.unizg.hr/predmet/rppzw 1. http.⇒ shema 2. www.fer.unizg.hr/p naziv poslužitelja (host 3. predmet/typzw → put I oznaka resursa Opća sintaksa URI-ja 2. predmet/prpzw → put I oznaka resursa Opća sintaksa URI-ja scheme // userinfo@ host: port path? quer scheme // userinfo@</td><td>dvije igredikarkor/SQUHovnstikimbase/34/2344/2342 sacaje - 9 - Jo olgovoru se ne nalazi voo Uzbaro sukului jer checktor nema arribut namel da ima pisudujer checktor sukulia in ili ili ili ili ili ili ili ili ili</td><td>div style="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> cspan style="font-size: 1 cspan sty</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula-exports = (createCart, addIteamToCart) ili Ovako - modula-exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu korištil varjiable navedene u datoteci "routes, ja na sijedeći način: ako je u "routes, ša datote navedeno: user : reg.session.user, cart : red.session.user, cart : red.session.user</td><td>Meta-podací (svystva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg. rh. ju (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđivanje iz drugih domena - kolačićo stvara poslužitelj, definira sadržaj i svojstva kolačíca -
kolačićo sprvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačićo sprvarajuje klijent, ako je kolačići solaj imena s istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačićo spluti viljeni poslužitelj. zahtljeva in nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain=www.fer. unizg. hr.; Secure: iRttpOn1y; Expires=#ed., 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putz a koji kolačić vrijedi: Parli-mastava Domena za koju kolačić vrijedi: Domain=www.fer. unizž. hr. Kolačić se nete prosljedival samo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolačić stječe: Expires=Wed. O' 70 May 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić defirinar, 2. Put definiran za kolačić u jelni je sadržan unutar puta dohvaćenog resursa, 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji manja definiran ok vignaja, 4. Kolačić koji manja definiran ok trajanja, 4. Kolačić koji minja definiran os vigtovo secure mogu biti prosljedivani samo sigurnim kanalima (https., ne http), 6. Samo ako je to izravno zabranjeno, kolačić neće biti proslijeden iz druge domene. **Trajnost kolačića: Trajni (persisten) kolačić (imaju definirano vvijemo evijemo edinirano definirano okrajova ozbranjeno, kolačić (imaju definirano vijemo edinirano okrajova ozbranjeno, kolačić (imaju definirano vvijemo edinirano koje kolačića: Trajni (persisten) kolačića (imaju definirano vvijemo edinirano svojetova edinirano svojetova edinirano kojetova edinirano koje</td><td>asvetNattalized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kāko izvršavati query-e? Jedan primjer kako: (async)(*) - let rows = awatt db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanima resursom GET - za dohvaćanje podataka o resursu (npr. provjera da postoji, veličina) POST - aktivranje resursa (npr. slanje podataka o brazaca) PUT - postavljanje enliteta (npr. promjena podataka - kod REST-a) DELETE - brisanje resursa (npr. kod REST-a) TRACE - za dijagnostiku DELETE - brisanje resursa (npr. kod REST-a) TRACE - za dijagnostiku - CONNECT - potreće dvosnjenu komunikaciju s traženim resursom putem provy-ja prema web-sjedištu koje koristi HTTPS Razlika izmedu GET-a i HEAD-s Veličina - GET je puno vedi - Zaglavlja - identična su im - Response - HEAD nema response - HEAD - metodrična su im - READ - remetodra su drivadnje podataka o resursu - razlika u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora</td><td>export class HighScores() { const highscores = localStorage, gettem ("highscores"); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); if (or the list static updatedlighscores (core, user) { const highscores =</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
Opis tileka komunikacije: 1. Preglednik šalje zatijev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja vz preglednik generira jednokratni simetričn ključ 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja vs. Preglednik salje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifirra ključ sjednice psvog privatnog ključa šalje u odgovoru ključ: slifitran samim sobom 7. Ključ sjednice koristi se za daljnje šifirani zumedu preglednika i poslužitelja klemifikacija resusen. Javneme (pr. umieti. UBN. – Uniform Resource lomane (pr. umieti. UBN. – Uniform Resource lomane (pr. umieti. UBN. – Uniform Resource kame (pr. umieti. umieti. umie</td><td>igr-diarkor/S/2046-yotk/Birbaso-24/2344/2342 acaigi- 9 dogovoru se ne nelazi voo Uzorasi jer dogovoru se ne nelazi voo Uzorasi jer dopovori Usorasi Usoles ne nelazi voo Uzorasi Usoles nelazi voo Nelazi voo Voo Voo Voo Voo Voo Voo Voo Voo Voo</td><td>div style="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> <pre> cspan style="font-size: 1 cspan sty</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula- exports = { createCart, addItemToCart} ili Ovako - modula- exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - srintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varjiable navedene u datoted "routes js na sijedeći način: ako je u "poukes js datoted nacin: ako je u "poukes js datoted nacin: ako je u "poukes js datoted nacin: cart; cnda u "eise-u možemo pisali sijedeće: <\$ for (tiem of Object.values (cart.items)) { \$>\$ cditv class" emery" data-id="<\$ item.id \$>>" () I tako dalje <ddiv><= item.price \$></div>cdiv><= item.price \$></div> * (if (user != undefined) (%> MVC - CONTROLLER * (NOTROLLER * Pouter = controller > objašnjeno gore u express routing * (Protaje ga u *) prima zalnikev > pribavlja model > predaje ga u * (Protaje ga u *) prima zalnikev > pribavlja model > predaje ga u * (Protaje ga u *) prima zalnikev > pribavlja model > predaje ga u * (Protaje ga u *) prima zalnikev > pribavlja model > predaje ga u * (Protaje ga u *) prima zalnikev > pribavlja model > predaje ga u * (Protaje ga u * (Protaj</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg.nh. jul (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje klijent, ako je kolačić istoja (jimas a istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće Stranjenjeni informacija bit če zamljenjena novom - kolačiće Stranjenjeni informacija bit če zamljenjena novom - kolačiće Salju i klijeni ti poslužitelj. zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain*movn. Eer. unizg. hr.; Secure: iRttpOn1y; Expires**edd. 70 Nay 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee → sadržaj je OBAVEZAN !!! Putz a koji kolačić vrijedi: Domain*movn. Ver. unizg. hr.; Secure: iRttpOn1y; Kolačić sa pete prosljedival samo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokainc(HtpOn1y) Kolačić sitejece: Expires**Ved. (70 Nay 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić definiran z. Put definiran za kolačiću cijelni je sadržan unutar puta dohvaćenog resursa. 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji imsilų definirano svijstvo secure mogu bili prosljedivani samo sigurnim kanalima (https., ne http), 5. Samo ako je to izavno zabranjeno, kolačića čne biti prosljedeni z duge domene. Trajnost kolačića: Trajni (persistent) kolačići (imaju definirano vrijeme isteka, bršu se nakon roka isteka valjanosti, niječeće imaju maxAge, bršu se nakon rokrisink putmo pojeju u pregledniku),</td><td>asvetīnātālisēd : Erus, coxicis : (asaxāge : 60+60+24+1000) // 1 dan})); Kāko izvīsāvaid query-e² Jodan primjer kako: (async (1) = Jet rows = awatt dbquery ('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno)).0; METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogudonostima poslužitelja i komunikacijskim zahigevima povezanim s resursom GET - 2a dotivadranje sodatako o resursu (npr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako o resursu (npr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako o resursu (npr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako o resursu (npr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako (pr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako (pr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako (pr. provjera da postoji, veličina) FLAD - 2a dotivadranje podatako (PEST-a) FLAD - 2a dotivadranje podatako (PEST-a) Veličina - GET je puno već Zaglavlja - identična su im - Response - HEAD nema response - HEAD - metoda za odivadranje podataka o resursu - ražika u domosu na GET: poslužitelju e vraca sadržaj resursa u tijelu odgovora - najčešće se koristi kako bi se provjerilo postoji li entitet na poslužitelju</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores) return JSON.parse(highscores); } //Add new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores = HighScores, getHighScores(); let foundEntry = highscores, findIndex(oneEntry => oneEntry, username = user); if(foundEntry != -1) { highscores.splice(foundEntry, 1); highscores.</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni
<i>Opis tileka komunikacije</i>: 1. Preglednik šalje zahljev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog kij certifikata 4. Ako su certifikat i javni ključ poslužitelja vz preglednik generira jednokratni simetrični ključ 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja os privatnog ključa i šalje u odgovoru ključ sjednice je svog privatnog ključa i šalje u
odgovoru ključ sjednice je svog privatnog ključa i šalje u odgovoru ključ i slifiran samim sobom 7. Ključ sjednice koristi se za daljnje šifriranji između preglednika i poslužija identifikacija resursa URI – Uniform Resource Locator Analiza URI-ja HTTP-g: 1. What Furunizg hr / predmet/rppzw 1. http // www.fer.unizg hr / predmet/rppzw 1. http // shema 2. www.fer.unizg hr / praziv poslužitelja (host 3. predmet/rppzw → put I oznaka resursa Opća sintaksa URI-ja scheme // userinfo @ host: port path ? query - userinfo → obično se ne koristi - redoslijed je uglavnom drukčija – host se nali patha, a pot može bit lizostavjne. Primjer: -</td><td>igr=Markoris/Coll-dovatikirhasa-23/42/34/2342 acaigis — 3 optoporeu se ne nalazi voo Uzorasi
kijuč i Usoporeu se ne nalazi voo Uzorasi
uča i Usopison obavijen zahtjevi:
200 OK – usopison obavijen zahtjeva:
300 Multiple Choices
300 Multiple Choices
300 Multiple Choices
301 Multiple Choices
302 Multiple Choices
303 Multiple Choices
304 Multiple Choices
305 Multiple Choices
305 Multiple Choices
306 Multiple Choices
307 Multiple Choices
308 Portuka
408 Orova – resurs nije modificiran or
25 Zadnjeg puta kad smo ga tražili (If-Modified-
u deaderu je false)
Pogreška na klijentu:
400 Bad Request – pogreška u sintaksi zah
401 Unauthorized – zahtjevu nedostaje
401 Modified- resurs nije dostupna, ali nulazi u detalje zašto – obično je greška pri
25 zadavaju URFlaj
26 Pogreška na poslužitelju:
27 Mod Found – resurs nije dostupna, ali nulazi u detalje zašto – obično je greška pri
28 zadavaju URFlaj
28 Pogreška na poslužitelju:
30 Mortupa – resurs prije dostupna, ali nulazi u detalje zašto – obično je greška pri
26 zadavaju URFlaj
28 Sevrice Unavallable – prostužitelj pe</td><td>div atyle="font-size: 30px;" <pre> <pre> <pre> <pre> div atyle="font-size:</td><td>- pišemo neke funkcije, definiramo klase, konstruktore I na kraja to "exportamo" za dalinje korištenje pomoću: (primjer) module. exports = { createCart, addIteamCoCart} ili ili ovako - module. exports = class User{-} MVC-VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sinitaksa gotovo identična kao HTML-ova, smanjim promjenama - u EJS-u se mogu koristili varijable navedene u datoted "routes ja na sijedeci način: sko je u "routes ja datotei način: sko je u "routes ja datotei način: sko je u "routes ja datotei način: oza re reg. seasion. user, cart : reg. seasion. user, cart : reg. seasion. user, cart : reg. seasion. user, coli vi i i i i i i i i i i i i i i i i i</td><td>Meta-podací (svojstva kolačića) → domenajako nije definirana onda je uzima ime posituželjali prv. svom fer. unizg. hr.) ni (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosjedivanje z idrugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića - kolačiće potranuje klijeni, ako je kolačića (soji imena s istog poslužitelja prethodna informacija bit če zamijenjena novom - kolačiće šalju i klijeni ti poslužilej, zabiljeva in initko Primjer. kolačiće šalju i klijeni ti poslužilej, zabiljeva in initko Primjer. Set-Cookie: SID=4a57e40d0ee; Path=/nastava; Domain=wnw. čer. unizg. hr.; Secure: išttponīty; Explires=*Med. 07 May 2014 10:18:12 dett* Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Put za koji kolačić vinjedi: Brahmastava Domena za koji kolačić vinjedi: Brahmastava Domena za koji kolačić vinjedi: Bromainavewy fer. uniz1/hr Kolačića se ne Express=*Wed. (7 May 2014 10:18:12 dett* Kolačića septe Express=Wed. (7 May 2014 10:18:12 dett* Kolačića septe presigledival sadrža in jesteka or krajanja ila unem definiran rok krajanja. 4. Kolačić koji imaju definirano svojstvo secure mogu bili prospledaval samo sigurnim kanalima (Huts, ne http). S. Samo ako je to izravno zabranjeno, kolačić neće bili prosiljeden iz druge domene. Trajnost kolačića: Trajni (persisten) kolačić inje isteka valjanosti, najčešće imaju maxAge. brišu se nakon proka isteka valjanosti, najčešće imaju maxAge. brišu se nakon proka isteka valjanosti, prevenema (transient) - session cookies (nemaju definirano vvijeme siteka valjanosti, najčešće imaju maxAge. brišu se nakon proka isteka valjanosti, prevenemo (transient) - session cookies (nemaju definirano vvijeme isteka valjanosti, prevenano reza</td><td>asvetNattalized: true, cookie: (asxAge: 60+60+24+1000) // 1 dan})); Kāko izvršavati query-e? Jedan primjer kako: (async)(*) - let rows = awatt db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanima resursom GET - za dohvaćanje podataka o resursu (npr. provjera da postoji, veličina) POST - aktivranje resursa (npr. slanje podataka o brazaca) PUT - postavljanje enliteta (npr. promjena podataka - kod REST-a) DELETE - brisanje resursa (npr. kod REST-a) TRACE - za dijagnostiku DELETE - brisanje resursa (npr. kod REST-a) TRACE - za dijagnostiku - CONNECT - potreće dvosnjenu komunikaciju s traženim resursom putem provy-ja prema web-sjedištu koje koristi HTTPS Razlika izmedu GET-a i HEAD-s Veličina - GET je puno vedi - Zaglavlja - identična su im - Response - HEAD nema response - HEAD - metodrična su im - READ - remetodra su drivadnje podataka o resursu - razlika u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, gettem ("highscores"); if (highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); if the was score for user (and remove old one), and sort the scores for the list //Add new score for user (and remove old one), and sort the scores for the list onath highscores: else (and remove old one), and sort the scores (list) let foundfattry = highscores (secore, user) { let foundfattry = highscores.publ(highscores indiance (oneEntry</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Opis tileka komunikacije: 1. Preglednik šalje zatijev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja vz preglednik generira jednokratni simetrični ključ 5. Preglednik salje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešfirira ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešfirira ključ sjednice javnem ključem poslužitelja 6. Poslužitelja prima i dešfirira ključ sjednice javnem ključejednika i poslužitelja kletnifikacija resurea 10. Uniform Resouce Identifier 10. Uniform Identifier 10. Uniform Identifier 10. Uniform Identifier 10. Uniform Iden</td><td>igredikarkor/SQUHoyndikimbacs/34/2344/2342 acaige 9 di olgopore se ne halizi rov Uzorasi Villori III. kijuč i Uzorasi Villori Villori</td><td>div style="font-size: 30px;" <pre> <pre> <pre></td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula-exports = { createCart, addItemToCart} ili Ovako - modula-exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoteci "routes ja na sijedeći način: ako je u "routes, is datoteci navedeno: user : req.session.user, cart : req.session.cart, onda u "ejs-u možemo pisati sjedeće: <\$ for (tam of Object.values(cart.items)) { \$>\$ cdit v class" embry data-id="Cart" embr</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg.nh. jul (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće potranuje klijent, ako je kolačić istoja (jimas a istog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće Stranjenjeni informacija bit če zamljenjena novom - kolačiće Stranjenjeni informacija bit če zamljenjena novom - kolačiće Salju i klijeni ti poslužitelj. zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Path*/nastava; Domain*movn. Eer. unizg. hr.; Secure: iRttpOn1y; Expires**edd. 70 Nay 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee → sadržaj je OBAVEZAN !!! Putz a koji kolačić vrijedi: Domain*movn. Ver. unizg. hr.; Secure: iRttpOn1y; Kolačić sa pete prosljedival samo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokainc(HtpOn1y) Kolačić sitejece: Expires**Ved. (70 Nay 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić definiran z. Put definiran za kolačiću cijelni je sadržan unutar puta dohvaćenog resursa. 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji imsilų definirano svijstvo secure mogu bili prosljedivani samo sigurnim kanalima (https., ne http), 5. Samo ako je to izavno zabranjeno, kolačića čne biti prosljedeni z duge domene. Trajnost kolačića: Trajni (persistent) kolačići (imaju definirano vrijeme isteka, bršu se nakon roka isteka valjanosti, niječeće imaju maxAge, bršu se nakon rokrisink putmo pojeju u pregledniku),</td><td>asvet/natialized : Eruse, cookie : (asax/ae : 60+60+24+1000) // 1 dan})); Kako izvršavati query-e² Jedan primjer kako: (async) (*) - Let rows = awalt db. query (*) SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućnostima poslužitelja i
komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje podataka o resursu (prp. provjera da postoji, veličina) POST - aktivanje resursa (nrp. slanje podataka obrazaca) PUT - postavljanje entiteta (nrp. promjena podataka - kod REST-a) DELETE - brisanje resursa (nrp. kod REST-a) Veličina - GET je puno veći - Veličina - GET je puno veći - Zaglavlja - lednična su im - Response - HEAD nema response - HEAD - metodična su destruje podataka o resursu - razilka u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora - naješeće se koristi kako bi se provjerilo postoj li entitet na poslužitelju - provjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o entitetu na poslužitelju - provjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o entitetu na poslužitelju - provjera veličine datoteke prije dohvaćanja</td><td>export class HighScores() { const highscores = localStorage, gettem("highscores); if ('highscores) { throw new Error("Highscores); if ('highscores) { throw new Error("Highscores); } //Add new Error("Highscores); } //Add new Score for user (and remove old one), and sort the scores for the list static updateHighScores (ore, user) { const highscores = HighScores(); let FoundEntry = highscores; findIndex(oneEntry **) consEntry.username ** user); if(FoundEntry != -1) { highscores.splice(FoundEntry, 1); localStorage.selflee("highscores"); JOSON.string(fy (highscore)); localStorage.selflee("highscores"); JOSON.string(fy (highscore)); constructor="milligupdateTimeTU" = null; updateHighScoreListUI = null; restCameUI = null; updateHighScoreListUI, resetGameUI) { this.updateTimerUI, updateHighScoreListUI, resetGameUI) { this.updateTimerUI, updateHighScoreListUI, resetGameUI) { this.updateTimerUI = updateTimerUI; }</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni <i>Qost tileka komunikacije</i>: 1. Preglednik šalje zahljev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja ve preglednik generira jednokratni simetričn ključ 5. Preglednik šalje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešfirra ključ sjednice javnim ključem poslužitelja 5. Proslužitelja prima i dešfirra ključ sjednice javned preglednika i poslužitelja klitran samim sobom 7. Ključ sjednice koristi se za daljnje šifirrani zmeđu preglednika i poslužitelja klitrani Luritorm Resource Identifier LURI — Uniform Resource Iden</td><td>igr-diarkors/Coll-dovatikinhaso-24/2344/2342 acaigie 3 dogoveru sen enalizi ovo Uzzorati jer checkbox nema airibut namel dia ima pis udia i Uspiešno obavijen zahtjevi: 200 OK – uspiešno obavijen zahtjevi: 200 OK – uspiešno obavijen zahtjevi: 200 OK – uspiešno obavijen zahtjevi: 300 Multiple Choices 300 Multiple Choices 301 Multiple Choices 301 Multiple Choices 302 Found – resurs pie PRIVEEMENO prem na drugi poslužitelj 303 Multiple Choices 403 Portividen – resurs nije modificiran or zadnjeg puta kad smo ga trazili (If-Modified- udadru je false) 403 Fortividen – drugi poslužitelj 404 Fortividen – drugi poslužitelj 405 Fortividen – drugi poslužitelj 406 Fortividen – drugi poslužitelj 407 Fortividen – drugi poslužitelj 408 Fortividen – drugi poslužitelj 409 Fortividen – drugi poslužitelj 409 Fortividen – drugi poslužitelj 400 Fortividen – drugi poslužitelj 401 Fortividen – drugi poslužitelj 402 Fortividen – drugi poslužitelj 403 Fortividen – drugi poslužitelj 404 Not Found – resurs nije dostupan, ali nu
21 NAKON 503 Service Unavailable – proslužitelj ju 500 Internal Server Error – obično progran 500 Internal Server Error – obično progran 500 Fortividen – proslužitelj ju 500 Fortividen – prospoven može sadržavati po 600 Fortividen – proslužitelj ju 500 Fortiviten – prosl</td><td>div style="fort-size: 30px;" <pre> <pre> <pre> div style="fort-size: 30px;" <pre> <pre> <pre> <pre> div style="fort-size: 1</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula- exports = { createCart, addItemToCart, ili Ovako - modula- exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varjiable navedene u datoted "routes ja na sijedeći način: ako je u "zoukejs is datoteći nacin: ako je u "zoukejs is datoteći nacin: ako je u "zoukejs is datoteći nacin: cart ; eng. session. user, cart : req. session. cart, onda u "eis-u možemo pisali siledeće: <\$ for (tam of Object. values (cart. items)) { \$>\$ cditv class" emergravi data-id="<\$ item.id \$>>" () I tako dalje <ddi>>>di.v>cia item.price \$></div> - cdiv>- item.price \$></di> - cdiv>- item.price \$></div> - cdiv>- it</td><td>Meta-podací (svojstva kolačka) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg. nh. jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđuvanje z drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića - kolačiće potranjuje klijent, ako je kolačić istoja (jimas a istog poslužitelja prethodna informacija bit če zamijenjena novom - kolačiće aplica i klijenti poslužitelj, zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Pather/nastava; Domain=mow. Eer. unizg. hr.; Secure: iRttpOn1y; Expires=Med. 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putz ako jik kolačić vrijedi: Path-inastava Domena za koju kolačić vrijedi: Domain=www.fer.uniz1r Kolačić se ne može pristupati kokano(HttpOnly) Kolačić sieptec: Expires=Wed. 07 May 2014 10:18:21 Prosljeđivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić defirinar, 2. Put definiran za kolačić ujelni je sadržanj unutar puta dohvaćenog resursa. 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji maju definirano svijstvo secure mogu biti prosljeđivanja samo sigurnim kanalima (kttps, ne http), 6. Samo ako je to izavno zabranjeno, kolačića dreće biti proslijeden iz druge domene. Trajnost Kolačića: Trajni (persistent) kolačić (imaju definirano vrijeme isteka, brišu se nakon roka isteka valjanosti, naječeće imaju max/ąe, brišu se nakon roka isteka valjanosti, naječeće imaju max/ąe, brišu se nakon roka isteka valjanosti, propiemom sadržaja kolačića – rutenči (koriski putem pogie u pregledniku), automatsko brisanje istekom roka valjanost, propiemom sadržaja kolačića. Prafist (propiemom sadržaja kolačića. Prafist (propiemom sadržaja kolačića.)</td><td>asvetīnātālisēd : Erus, cockie : (asaĀge : 60+60+24+1000) // 1 dan})); Kāko izvīsāvatī query-e² Jedan primjer kako: (async () = Jet rowe = awatī db. query ('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rowe); // opcionalno)).0; METODE HTTP ZAHTJEVĀ OPTIONS - informiranje o mogućostima poslužitelija i komunikacijskim zahijevima povezamin s resursom GET - za dohvačanje podataka o resursu (npr. provjera da postoji, veličina) POST - aktivranje resursa (npr. slanje podataka obrazaca) PUT - postavljanje emiteta (npr. promjena podataka - kod REST-a) DELETE - brasanje resursa (npr. kod REST-a) Veličina - GET- i HEAD-a veličina - GET- i HEAD-a veličina - GET- i puno ved - zaglavja – identična su in - Response - HEAD nema response - HEAD - metoda za dohvaćanje podataka o resursu - razika u odnosu na GET- poslužitelji ne vraća sadržaj resursa u tijelu odgovora - najčešće se koristi kako bi se provjerilo postoji li entitet na poslužitelju - druge uporabe: - provjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o entitetu na poslužitelju se zahtjev HEAD obraduje jednaka koa i GET</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores); } if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new Score for user (and remove old one), and sort the scores for the list static updateHighScores (core, user) { const highscores = HighScores (getHighScores); let foundEntry = highscores, findIndex(oneEntry *) voneEntry username = user); if(FoundEntry != -1) { highscores.splice (foundEntry, 1); localStorage user ("highscores"); JOSN stringfy(highscore); JOSN stringfy(highscore); constructed = false; cookieCount = 0; gemeStated = false; cookieCount = 0; gemeStated = false; cookieCount = unll; updateHighScoreListUI = unll; resetCameUI unll; constructor(updateTimerUI, updateHighScoreListUI, resetGameUI) { this.updateHighScoreListUI; restated updateHighScoreListUI; restated updateHightScoreListUI; restated updateHightScoreListUI; restated updateHi</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Obsi tileka komunikacije: 1. Preglednik šalje zahljev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja va preglednik generira jednokratni simetrični ključ 5. Preglednik salje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnog privatnog ključa i
šalje u odgovoru ključ sifriran samim sobom 7. Ključ sjednice koristi se za daljnje šifrirani zmedu preglednika i poslužitelja identifikacija resursa 1. Preglednika proslužitelja identifikacija resursa 1. Pred 1. Uniform Resource Name (npr. umieti: URL — Uniform Resource Name (npr. umieti: URL — Uniform Resource socator Analizu Ulejla HTTP-s. http: // www.fer.unizg.hr / predmet/rppzw 1. http -> bresim / predmet/rppzw 2. www.fer.unizg.hr / predmet/rppzw 2. www.fer.unizg.hr / poslužitelja (host 3. predmet/prpzw -> put I cznaka resursa Opća sintaksa URI-ja 2. www.fer.unizg.hr / poslužitelja (host 3. predmet/prpzw -> put I cznaka resursa Opća sintaksa URI-ja 1. redoslijel oj su ujavnom drukčji - nost se nati path-a, a pott može biti izostavljen. Primjeri: https://www.fer.unizg.hr/rossi@-zlezujiekapath-a path-a, a pott može biti zostavljen. Primjeri: https://www.fer.unizg.hr/rossi@-zlezujiekapath-a path-a, a pott može biti zostavljen. Primjeri: http</td><td>igredikarkor/SQUHoyndikimbacs/34/2344/2342 acaige 9 di olgopore se ne halizi rov Uzorasi Villori III. kijuč i Uzorasi Villori Villori</td><td>div style="font-size: 30px;" <pre> <pre> <pre></td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za dalinje korištenje pomoću: (primjer) modulie.exports = (createCart, addIteamToCart) ili Ovako — modulie.exports = class User{-} MVC-VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoted 'routes, ja na sijedeci način: sako je u 'routes, ja datoje navedeno: user : req. session. user, oat : req. session. user, oat : req. session. user, oat : req. session. user, odi v ilasse "amendeno" in the composition of the compo</td><td>Meta-podací (svojstva kolačíca) → domenajako nije definirana onda je uzima ime posituželnia pro. svom fer. unizg. nh. jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosijedivanje z krugih domena - deložiće stvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće sptvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće sptvara poslužitelj, definira sadržaj i svojstva kolačíca - kolačiće sptvara jud kilinira sadržaj svojstva kolačíca sptvara poslužitelj, zabriljeva ni nitiko Primjer. kolačiće sptvari i kljenit rjosužitelj, zabriljeva ni nitiko Primjer. Set-Cookie: SID=4a57e40d0ee; Path=/nastava; Domain=wnov. žer. unizg. hr.; Secure: ilttponity; Expires=*ded, 07 May 2014 10:18:21 GMT Sadržaj kolačica: SID=4a57e40d0ee → sadržaj je OBAVEZAN !!! Puti za koji kolačić vinjedi: Enhamastava Domena za koji koji koji koji koji koji koji koji</td><td>asvetīnātālisēd : Erus, cockie : (asaĀge : 60+60+24+1000) // 1 dan})); Kāko izvīsāvatī query.e² Jedan primjer kako: (async.() = Jet rowe = awatī db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rowe); // opecionalno)).01; METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućonostima poslužitelija i komunikacijskim zahijevima povezamina resurusom GET - za dohvačanje podataka o resurus (npr. provjera da postoji, veličina) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka - kod REST-a) DELETE - brisanje resursa (npr. kod REST-a) Veličina - GET [1] punn ved Zaglavija - identična su in - Response - HEAD nema response - HEAD - metoda za dohvaćanje podataka o resursu - razika u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora - najčešće se koristi kako bi se provjerino postoji li entitet na poslužitelju - druge uporabe: - provjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o entitetu na poslužitelju se zahtjev HEAD obraduje jednaka koa i GET - POST - metoda za "aktiviranje" resursa znad "pomoću adresiranog</td><td>export class HighScoreManager { static getHighScores { const highscores = localStorage, gettem ("highscores"); if (highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); if throw score for user (and remove old one), and sort the scores for the list } //Add new score for user (and remove old one), and sort the scores for the list } //Add new score for user (and remove old one), and sort the scores for the list</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Obsi tileka komunikacije: 1. Preglednik šalje zahljev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja va preglednik generira jednokratni simetrični ključ 5. Preglednik salje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnog privatnog ključa i šalje u odgovoru ključ sifriran samim sobom 7. Ključ sjednice koristi se za daljnje šifrirani zmedu preglednika i poslužitelja identifikacija resursa 1. Ru Uniform Resouvce Name (npr. umieti: URL — Uniform Resouvce Name (npr. umieti: URL — Uniform Resouvce otentifer URL — Uniform Resouvce Name (npr. umieti: U</td><td>igredikarkor/SQUHoyndikimbago-324/3244/3242 kacaje by Jodgoveru sen benaltzi öru Üzerasi vi Jodgoveru sen benaltzi öru Üzerasi vi Jodgoveru sen benaltzi öru Üzerasi vi Libi v</td><td>div atyle="font-aire: 30px;" div atyle="font-aire: 30px;" div atyle="font-aire: 1 em;">em</pre> dip atyle="font-aire: 1 em;">em</pre> dip atyle="font-aire: 1 expace-between, space-around, space-between, s</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) modula- exports = { createCart, addItemToCart, ili Ovako - modula- exports = class User{-} MVC - VIEW - EJS file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varjiable navedene u datoted "routes ja na sijedeći način: ako je u "zoukejs is datoteći nacin: ako je u "zoukejs is datoteći nacin: ako je u "zoukejs is datoteći nacin: cart ; eng. session. user, cart : req. session. cart, onda u "eis-u možemo pisali siledeće: <\$ for (tam of Object. values (cart. items)) { \$>\$ cditv class" emergravi data-id="<\$ item.id \$>>" () I tako dalje <ddi>>>di.v>cia item.price \$></div> - cdiv>- item.price \$></di> - cdiv>- item.price \$></div> - cdiv>- it</td><td>Meta-podací (svojstva kolačka) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem ker unizg. nh. jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljeđuvanje z drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića - kolačiće potranjuje klijent, ako je kolačić istoja (jimas a istog poslužitelja prethodna informacija bit če zamijenjena novom - kolačiće aplica i klijenti poslužitelj, zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Pather/nastava; Domain=mow. Eer. unizg. hr.; Secure: iRttpOn1y; Expires=Med. 07 May 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putz ako jik kolačić vrijedi: Path-inastava Domena za koju kolačić vrijedi: Domain=www.fer.uniz1r Kolačić se ne može pristupati kokano(HttpOnly) Kolačić sieptec: Expires=Wed. 07 May 2014 10:18:21 Prosljeđivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić defirinar, 2. Put definiran za kolačić ujelni je sadržanj unutar puta dohvaćenog resursa. 3. Kolačiću nije istekao rok trajanja ili nema definiran rok trajanja, 4. Kolačić koji maju definirano svijstvo secure mogu biti prosljeđivanja samo sigurnim kanalima (kttps, ne http), 6. Samo ako je to izavno zabranjeno, kolačića dreće biti proslijeden iz druge domene. Trajnost Kolačića: Trajni (persistent) kolačić (imaju definirano vrijeme isteka, brišu se nakon roka isteka valjanosti, naječeće imaju max/ąe, brišu se nakon roka isteka valjanosti, naječeće imaju max/ąe, brišu se nakon roka isteka valjanosti, propiemom sadržaja kolačića – rutenči (koriski putem pogie u pregledniku), automatsko brisanje istekom roka valjanost, propiemom sadržaja kolačića. Prafist (propiemom sadržaja kolačića. Prafist (propiemom sadržaja kolačića.)</td><td>asvetīnātālisēd : Exue, cookie : (asaxāg : 60+60+24+1000) // 1 dan})); Kāko izvīšavatī query.e². Jedan primjer kako: (async) (*) - Let rows = awalt db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rows); // opcionalno))(); METODE HTTP ZAHTJEVĀ OPTIONS - informiranje o mogućnostima poslužitelja i komunikacijskim zahljevima povezanim s resursom GET - za dohvaćanje sodatka o resursu (prp. provjera da postoji, veličina) POST - aktivranje resursa (nrp. slanje podataka obrazaca) PUT - postavljanje entiteta (prp. promjena podataka - kod REST-a) DELETE - brisanje resursa (nrp. kod REST-a) Veličina - GET je puno vedi - Veličina - GET je puno vedi - Zaglavlja - ledentična su im - Response - HEAD nema response - HEAD - metolična su im - Response - HEAD nema response - HEAD nemotična su im - Response - HEAD nema response - HEAD nemotična su im - razilka u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora - naješeće se koristi kako bi se provjerilo postoji li entitet na poslužitelju - privovjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o i GET - POST - metoda za *aktivranje' resursa znad *pomoću adresiranog</td><td>export class HighScoreManager { static getHighScores { const highscores localStorage, getItem ("highscores"); if
('highscores) { throw new Error("Highscores); } if ('highscores) { throw new Error("Highscores not set!") } return JSON.parse(highscores); } //Add new Score for user (and remove old one), and sort the scores for the list static updateHighScores (core, user) { const highscores = HighScores (getHighScores); let foundEntry = highscores, findIndex(oneEntry *) voneEntry username = user); if(FoundEntry != -1) { highscores.splice (foundEntry, 1); localStorage user ("highscores"); JOSN stringfy(highscore); JOSN stringfy(highscore); constructed = false; cookieCount = 0; gemeStated = false; cookieCount = 0; gemeStated = false; cookieCount = unll; updateHighScoreListUI = unll; resetCameUI unll; constructor(updateTimerUI, updateHighScoreListUI, resetGameUI) { this.updateHighScoreListUI; restated updateHighScoreListUI; restated updateHightScoreListUI; restated updateHightScoreListUI; restated updateHi</td></tr><tr><td>osnovne faze: Faza rukovanja I Faza komuni Obsi tileka komunikacije: 1. Preglednik šalje zahljev poslužitelju 2. Poslužitelju odgovoru dostavlja svoj javni certifikat 3. Preglednik provjerava valjanost javnog klj certifikata 4. Ako su certifikat i javni ključ poslužitelja va preglednik generira jednokratni simetrični ključ 5. Preglednik salje u odgovoru ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnim ključem poslužitelja 6. Poslužitelja prima i dešifrira ključ sjednice javnog privatnog ključa i šalje u odgovoru ključ sifriran samim sobom 7. Ključ sjednice koristi se za daljnje šifrirani zmedu preglednika i poslužitelja identifikacija resursa 1. Ru Uniform Resouvce Name (npr. umieti: URL — Uniform Resouvce Name (npr. umieti: URL — Uniform Resouvce otentifer URL — Uniform Resouvce Name (npr. umieti: U</td><td>igredikarkor/SQUHoyndikimbago-324/3244/3242 kacaje by Jodgoveru sen benaltzi öru Üzerasi vi Jodgoveru sen benaltzi öru Üzerasi vi Jodgoveru sen benaltzi öru Üzerasi vi Libi v</td><td>div atyle="font-aire: 30px;" div atyle="font-aire: 30px;" div atyle="font-aire: 1 em;">em</pre> dip atyle="font-aire: 1 em;">em</pre> dip atyle="font-aire: 1 expace-between, space-around, space-between, s</td><td>- pišemo neke funkcije, definiramo klase, konstruktore Ina kraja to "exportamo" za daljnje korištenje pomoću: (primjer) module. exports = (createCart, addItemToCart) ili Ovako - module. exports = class User(-) MVC - VIEW - LSI file - Embedded JavaScript Template - zaduženje: prezentacija dostavljenih podataka - sintaksa gotovo identična kao HTML-ova, s manjim promjenama - u EJS-u se mogu koristili varijable navedene u datoteci "routes ja na sijedeći način: ako je u "zoukejs is datoteci naveneco: user: req.session.user, cart: req.session.cart, onda u "ejs-u možemo pisati sijedeće: <\$ for (tiem of Object.values (cart.items)) { \$>\$ (tim of Object.values) { \$>\$ (tim of Object.values</td><td>Meta-podací (svojstva kolačka) → domenajako nije definirana onda je uzima ime poslužtelja pri, vem kr-unizg.nh, jud (domena + put = doseg), rok valjanosti, ograničenje pristupa, ograničenje na sigurnost prosljedivanje iz drugih domena - kolačiće stvara poslužitelj, definira sadržaj i svojstva kolačića - kolačiće optranuje klijent, ako je kolačići stoja (jima si stog poslužitelja prethodna informacija bit če zamljenjena novom - kolačiće aplitu i klijenti poslužitelj, zahtljeva ih nitko Primjer. Set-Cookie: SID=4a57e40d0ee; Pather/nastava; Domainsmow. Eer. unizg. hr; Secure: iRttpön1y; Expires=Med. O7 Nay 2014 10:18:21 OMT Sadržaj kolačića: SID=4a57e40d0ee > sadržaj je OBAVEZAN !!! Putz ako jik kolačić vrijedi: Palm-inastava Domena za koju kolačić vrijedi: Domain-sww.fer.uniz/hr Kolačić se nebe prosljedival samo sigurnim kanalima(Secure) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolačiću se ne može pristupati kokalno(HttpOnly) Kolačići selece: Expires=Wed, O'7 May 2014 10:18:21 Prosljedivanje kolačića: UVJETI: 1. Poslužitelj pripada domeni za koju je kolačić definiran ok trajanja, 4. Kolačić koji misaj definirano svojstvo secure mogu biti prosljedivani samo sigurnim kanalima (https, ne http), 5. Samo ako je to izavno zabranjeno, kolačića ebe biti proslijeden iz druge domene. Trajnost kolačića: Trajni (persistent) kolačić (imaju definirano vrijeme isteka, brišu se nakon roka isteka valjanosti, nješeće imaju max/ąe, brišu se nakon roka isteka valjanosti, nješeće imaju max/ąe, brišu se nakon roka isteka valjanosti, nješeće imaju max/ąe, brišu se nakon roka isteka valjanosti, projedoniku), automatsko brisanje istekom roka valjanost, projedovanja kolačića – utudni ji lokalni (klijent, programski) -ako je postavljeno svojskvo Http.Only nije moguć lokalni pristup kolačića - ograničenje prosljedivanja samo na sigurne komunikacijske kanale</td><td>asvetīnātālisēd : Erus, cockie : (asaĀge : 60+60+24+1000) // 1 dan})); Kāko izvīsāvatī query.e² Jedan primjer kako: (async.() = Jet rowe = awatī db. query('SELECT id, name, description FROM place_type WHERE id BETWEEN \$1 AND \$2', [1, 3]); console.log(rowe); // opecionalno)).01; METODE HTTP ZAHTJEVA OPTIONS - informiranje o mogućonostima poslužitelija i komunikacijskim zahijevima povezamina resurusom GET - za dohvačanje podataka o resurus (npr. provjera da postoji, veličina) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka obrazaca) POST - aktiviranje resursa (npr. slanje podataka - kod REST-a) DELETE - brisanje resursa (npr. kod REST-a) Veličina - GET [1] punn ved Zaglavija - identična su in - Response - HEAD nema response - HEAD - metoda za dohvaćanje podataka o resursu - razika u odnosu na GET: poslužitelj ne vraća sadržaj resursa u tijelu odgovora - najčešće se koristi kako bi se provjerino postoji li entitet na poslužitelju - druge uporabe: - provjera veličine datoteke prije dohvaćanja - pribavljanje metapodataka o entitetu na poslužitelju se zahtjev HEAD obraduje jednaka koa i GET - POST - metoda za "aktiviranje" resursa znad "pomoću adresiranog</td><td>export class HighScoreManager { static getHighScores { const highscores const highscores localStorage, getItem ("highscores"); if ('highscores) { throw new Error("Highscores); } if ('highscores); if throw new Error("Highscores not set!") } return JSON.parse(highscores); lotal throw score for user (and remove old one), and sort the scores for the list static updateHighScores (core, user) { const highscores = HighScores (getHighScores); litfoundEntry = highscores, findIndex(oneEntry *</td></tr></tbody></table></script></h3></h2></h1></legend> | | | | |

```
<u>UserModel.js</u> - iz 4. Labosa
const db = require('../db')
                                                                                                                                                                                                                                                                                                                                                             Preglednik šalje sljedeći zahtjev poslužitelju www.fer.unizg.hr na
                                                               Primjer padajućeg izbornika u ejs-u:
                                                                                                                                      CartModel.js
                                                                                                                                                                                                                                                                                                2. Labos – primjer dodavanja proizvoda na stranicu orde
→ "Napiši osnovne dijelove HTML-a I JS-a da bi se ostvario
                                                                                                                                         onst_db = require('.../db')
                                                                                                                                                                                                                                                                                                putem DOM-ar:
<template id="category template"> ...
</template id></template</pre>
                                                                                                                                                                                                                                                                                                                                                             vrata 80
                                   - Raspon su mu
                                                                  - 0
                                                                                                                                       function createCart() {
                                                                                                                                                                                                                 //razred User enkansulira korisnika web troovine
                                                                                                                                                                                                                                                                                                                                                            GET /logo.html HTTP/1.1
                                                                                                                                                                                                                 module.exports = class User {
ovakav brojač:
                                                                                                                                         let newCart = {}
                                                                                                                                                                                                                                                                                                                                                            Host: www.fer.unizg.hr
                                                                  type:</label>
brojevi 0-10. Kada se stranica učita, vrijednost mora biti 0."
                                                                                                                                          newCart.totalAmount = 0
                                                                                                                                                                                                                                                                                                 (script>
                                                                                                                                                                                                                                                                                                                                                            Traženi resurs logo.html prikazan je:
<!DOCTYPE html>
                                                                   "select class="form-component" id="promo-type"
                                                                                                                                                                                                                   constructor(username, first name, last name, email, password,
                                                                                                                                                                                                                                                                                                   let getData = async function () {
                                                                                                                                         newCart.items = {}
                                                                                                                                                                                                                 role="user")
<button onclick="decrement()">-</button>
                                                                  name="promo-type">
                                                                                                                                         return newCart
                                                                                                                                                                                                                                                                                                    let response = await
                                                                                                                                                                                                                                                                                                                                                             <html>
                                                                                                                                                                                                                     this.id = undefined
                                                                    <%for (const opt of promoOptions) { %>
<input id=numInput type=number min=0 max=10</pre>
                                                                                                                                         nction update(cart)
                                                                                                                                                                                                                                                                                                     fetch("https://web1lab2.azurewebsites.net.
                                                                    <option <%= promoType == opt ? 'selected' : '' %>
                                                                                                                                                                                                                     this.user_name = username
this.first name = first name
                                                                                                                                                                                                                                                                                                                                                                 k rel="stylesheet" href="logo.css">
value='0'>
                                                                                                                                         cart.totalAmount = 0
                                                                                                                                                                                                                                                                                                     categories");
<button onclick="increment()">+</button>
                                                                                                                                         for(let item of Object.values(cart.items))
  cart.totalAmount += item.price * item.quantity}
                                                                                                                                                                                                                                                                                                     let data = await response.ison();
                                                                       <%= opt%>
                                                                                                                                                                                                                                                                                                                                                               </head>
                                                                                                                                                                                                                     this.last_name = last_name
this.email = email
                                                                                                                                                                                                                                                                                                   addCategories (data); }
let addCategories = async function
JS (unutar istog HTML-a):
                                                                    </option>
                                                                                                                                                                                                                                                                                                                                                               <body>
                                                                                                                                        sync function addItemToCart(cart, id, quantity)
<script>
                                                                    <818>
                                                                                                                                                                                                                                                                                                                                                                 <div>
                                                                                                                                                                                                                                                                                                   (categories) {
  let main = document.querySelector('main');
  function increment() {
                                                                  </select>
                                                                                                                                        let itemObject = cart.items[id]
if( itemObject === undefined ) {
                                                                                                                                                                                                                     this.password = password
                                                                                                                                                                                                                                                                                                                                                                  <img class="left"</pre>
                                                                                                                                                                                                                      this role = role
                                                                                                                                                                                                                                                                                                                                                                   src = "https://akmuz.fer.hr/
   document.getElementBvId('numInput').stepUp();}
                                                                                                                                                                                                                 //dohvat korisnika na osnovu korisničkog imena
  function decrement() {
                                                               Primier submit-ania obrasca (EJS - promotional.eis):
                                                                                                                                           itemObject = {id: id, name: undefined, price:
   undefined, quantity: 0, imageurl: undefined}
                                                                                                                                                                                                                                                                                                     let categoryTemplate = document
                                                                                                                                                                                                                                                                                                                                                                   images/FER_logo_old.png">
<h2>Stari logo_vs. novi logo</h2>
 document.getElementById('numInput').stepDown();
                                                                                                                                                                                                                 static async fetchByUsername(username) {
                                                                                                                                                                                                                                                                                                      .querySelector('#category-template');
                                                                <input class="btn" type="submit"
formaction="./promotional/save" value="Save and return</pre>
                                                                                                                                                                                                                   let results = await dbGetUserByName(username)
                                                                                                                                                                                                                                                                                                     for (let index = 0; index < categories.
</script>
                                                                                                                                           cart.items[id] = itemObject }
                                                                                                                                                                                                                                                                                                                                                                  Koji vam se više svida?
                                                                                                                                         if(itemObject.price === undefined) {
  if( itemData = await gettemData(id) ) {
  Prilikom slanja podatka iz obrasca na web-poslužitelj nazivi I
                                                                                                                                                                                                                   let newUser = new User()
                                                                                                                                                                                                                                                                                                      length: index++) {
                                                                                                                                                                                                                                                                                                                                                                   <img class="right" src="./img/FER logo.png">
                                                                 input class="btn" type="submit"
                                                                                                                                                                                                                   if ( results.length > 0 ) {
                                                                                                                                                                                                                                                                                                     let category = categoryTemplate.content.cloneNode(true);
vrijednosti elemenata obrasca će se u HTTP-zahtjevu prenijeti:
                                                                                                                                                                                                                     newUser = new User(results[0].user_name,
metodom GET u URI/URL-u HTTP-zahtjeva kao parametri upita
                                                                formaction="./promotional/reset" value="Reset">
                                                                                                                                              itemObject.name = itemData.name
itemObject.price = itemData.price
                                                                                                                                                                                                                                                                                                                                                               </body>
                                                                <input class="btn push-right" type="submit"
formaction="./promotional" value="Order now!">
                                                                                                                                                                                                                 results[0].first_name, results[0].last_name, results[0].email,
                                                                                                                                                                                                                                                                                                    let categoryTitleElement = category.
querySelector('.decorated-title > span');
metodom POST u tijelu HTTP-zahtieva
                                                                                                                                                                                                                 results[0].password, results[0].role)
- Za upravljanje ponašanjem priručnog spremišta koristi se polje
Cache-Control (pojavljuje se u zahtjevu I odgovoru)
                                                                                                                                                                                                                                                                                                                                                             Skicirajte slijedni dijagram i označite sve akcije koje će preglednik
                                                                                                                                              itemObject.imageurl = itemData.image }}
                                                                                                                                        itemObject.quantity =
                                                                Primjer submit-anja obrasca (cijeli promotional.routes.js):
                                                                                                                                                                                                                 newUser.id = results[0].id }
                                                                                                                                                                                                                                                                                                      categoryTitleElement.textContent =
                                                                                                                                                                                                                                                                                                                                                             napraviti vezano uz poslužitelje ili priručnu memoriju te odgovore
                                                                                                                                            itemObject.quantity + quantity
                                                                                                                                                                                                                   return newHeer |
                                                                                                                                                                                                                                                                                                      categories[index].name;
→ Koja je KONKRETNA razlika izmođu  (uređene liste)
                                                                const express = require('express');
const router = express.Router();
                                                                                                                                                                                                                                                                                                                                                            poslužitelia. Pretpostavite da je poslužiteli www.fer.unizg.hr
                                                                                                                                                                                                                 //dohvat korisnika na osnovu email adrese
update(cart)
                                                                                                                                                                                                                                                                                                     let gallery =
                                                                                                                                                                                                                                                                                                                                                            konfiguriran tako da koristi protokol https te na prvi navedeni
zahtjev preglednika odgovara s 307 Temporary Redirect. Svi
                                                                 onst cart = require('../models/CartModel');
                                                                                                                                        sync function removeItemFromCart(cart, id, quantity) {
                                                                                                                                                                                                                static async fetchByEmail(email) {
                                                                                                                                                                                                                                                                                                    category.querySelector(".gallery");
main.appendChild(category);
predstavlia natuknice. Ú čistom HTML-u  će dati brojeve (1.
                                                                                                                                                                                                                   let results = await dbGetUserByEmail(email)
let newUser = new User()
2., 3., ...) a 
    će dati bulletpoints (* natuknica \n * druga

                                                                const authHandler = require('./helpers/auth-handler')
                                                                                                                                         let itemObject = cart.items[id]
                                                                                                                                                                                                                                                                                                                                                             poslužitelii su dostupni i mogu dostaviti sve tražene resurse, dok ie u
                                                                                                                                                                                                                                                                                                    let ProductResponse = await
fetch("https://web1lab2.azurewebsites.net.
                                                                                                                                         if( itemObject ) {
                                                                                                                                                                                                                                                                                                                                                             priručnoj memoriji preglednika pohranjena ažurna kopija navedenog
natuknica )
                                                                router.get('/', authHandler, function (req, res, next)
                                                                                                                                                                                                                   if ( results.length > 0 ) {
                                                                                                                                            let newQuantity =
→ Primjer opisne liste <dl>:
                                                                                                                                                                                                                                                                                                                                                            CSS-dokumenta logo.css.
                                                                                                                                                                                                                     newUser = new User(results[0].user_name,
                                                                  res.render('promotional',
                                                                                                                                              Math.max(0, itemObject.quantity - quantity)
                                                                                                                                                                                                                                                                                                     products?categorvId=" + (index + 1));
                                                                                                                                                                                                                                                                                                                                                             preglednik
                                                                                                                                                                                                                                                                                                                                                                                     www.fer.unizg.hr
                                                                                                                                                                                                                     results[0].first_name, results[0].last_name,
results[0].email, results[0].password, results[0].role)
newUser.id = results[0].id )
                                                                    title : 'Enter promo code',
linkActive : 'cart',
                                                                                                                                           if ( newQuantity )
                                                                                                                                                                                                                                                                                                      let ProductData = await
 <dt>name:</dt><dd>John Doe</dd>
  <dt>tel:</dt><dd>01=2345678</dd></dd></dr>
<dt>fax:</dt><dd>02=3456789</dd></dd></dr>
                                                                                                                                              itemObject.quantity = newQuantity
                                                                                                                                                                                                                                                                                                      ProductResponse.json();
                                                                                                                                                                                                                                                                                                                                                                     http://www.fer.unizg.hr/logo.html
                                                                    cart : req.session.cart,
user : req.session.user,
                                                                                                                                                                                                                                                                                                     let productTemplate = document
                                                                                                                                                                                                                   return newUser }
                                                                                                                                                                                                                                                                                                      .querySelector("#product-template");
                                                                                                                                              delete cart.items[id] }
  <dt>cdt>amail:</dt></dt></d>
                                                                                                                                                                                                                                                                                                                                                                     307 Temporary Redirect
https://www.fer.unizg.hr/logo.html
                                                                                                                                                                                                                 //dohvat korisnika na osnovu id korisnika (tablica users)
                                                                    promoType : reg.session.promotionType,
                                                                                                                                         update(cart) }
                                                                                                                                                                                                                                                                                                     for (let i = 0;
</d1>
                                                                    promoCode : req.session.promotionCode
                                                                                                                                        sync function getItemData(id) {
                                                                                                                                                                                                                 static asvnc fetchBvUserId(id) {
                                                                                                                                                                                                                                                                                                       i < ProductData length: i++) /
"Napišite HTML kôd obrasca za unos registracijskih podataka
                                                                                                                                                                                                                   let results = await dbGetUserById(id)
let newUser = new User()
                                                                                                                                        const sql = `SELECT name, price, imageurl FROM
inventory WHERE id = ` + id;
                                                                                                                                                                                                                                                                                                       let product = productTemplate
                                                                err : undefined }); });
router.post('/', function (req, res, next) {
(HTMI ili eis)"
                                                                                                                                                                                                                                                                                                                                                                     https://www.fer.unizg.hr/logo.h
create table user (
                                                                                                                                                                                                                                                                                                        .content.cloneNode(true)
                                                                  req.session.promotionType = undefined;
                                                                                                                                                                                                                   if( results.length > 0 ) {
                                                                                                                                                                                                                                                                                                        let productTitle = product
   email text primary key,
                                                                                                                                                                                                                     newUser = new User(results[0].user_name,
results[0].first_name,results[0].last_name,
                                                                                                                                            const result = await db.query(sql, []);
                                                                                                                                                                                                                                                                                                        querySelector(".photo-box-title");
                                                                                                                                                                                                                                                                                                                                                                       200 OK + logo.html
                                                                  req.session.promotionCode = undefined;
   ime text not null,
prezime text not null )
                                                                res.redirect('/checkout'); })
router.post('/save', function (req, res, next) {
                                                                                                                                                                                                                                                                                                        productTitle.textContent =
                                                                                                                                                                                                                     results[0].email, results[0].password, results[0].role)
newUser.id = results[0].id }
                                                                                                                                              name: result.rows[0].name,
                                                                                                                                                                                                                                                                                                                                                                     dohvati logo.css iz priručnog
<form action="/register" method="POST">
                                                                  reg.session.promotionType = reg.body['promo-type'];
                                                                                                                                              price: result.rows[0].price.
                                                                                                                                                                                                                                                                                                       let productImage = product
   <fieldset>
                                                                  req.session.promotionCode = req.body['promo-code'];
                                                                                                                                              image: result.rows[0].imageurl }
                                                                                                                                                                                                                   return newUser
                                                                                                                                                                                                                                                                                                        .guervSelector(".photo-box-image");
                                                                                                                                                                                                                                                                                                                                                                           GET https://akmuz.fer.hr/
                                                                                                                                                                                                                                                                                                                                                                                                nages/FER_logo_old.png
     <legend>Registration data</legend>
                                                                                                                                                                                                                 //da li je korisnik pohranjen u bazu podataka?
                                                                res.redirect('/cart'); })
router.post('/reset', function (req, res, next) {
                                                                                                                                         } catch (err) {
                                                                                                                                                                                                                                                                                                       productImage.src =
     <vib>
                                                                                                                                              console.log(err);
throw err }
                                                                                                                                                                                                                 isPersisted() {
  return this.id !== undefined }
//provjera zaporke
                                                                                                                                                                                                                                                                                                        ProductData[i].imageUrl;
                                                                                                                                                                                                                                                                                                                                                                           200 OK + FER logo old.png
        <label for="ime">Ime:</label>
                                                                  req.session.promotionType = undefined;
                                                                                                                                                                                                                                                                                                        let cartButton = product
       <input type="text" name="ime" id="ime'</pre>
                                                                  reg session promotionCode = undefined:
                                                                                                                                         return undefined }
                                                                                                                                                                                                                                                                                                        .querySelector(".cart-btn");
eize="30">
                                                                       redirect('/promotional'); })
                                                                                                                                                                                                                 checkPassword(password)
                                                                                                                                                                                                                                                                                                        cartButton.onclick = function() {
  addToCart(ProductData[i].id);
                                                                                                                                                                                                                                                                                                                                                                      attps://www.fer.unizg.hr/img/FER_logo.png
                                                                                                                                          dule.exports =
     </div>
                                                                                                                                                                                                                 return this.password ? this.password == password : false 
//pohrana korisnika u bazu podataka
                                                                module.exports = router;
                                                                                                                                          createCart,
      <div>
                                                                Primjer router.post fje pri prijavi korisnika (login.routes.js):
                                                                                                                                          addItemToCart
                                                                                                                                                                                                                                                                                                                                                                        200 OK + FER_logo.png
         <label for="prezime">Prezime:</label>
                                                                router.post('/', function (req, res, next) {
                                                                                                                                          removeItemFromCart }
                                                                                                                                                                                                                 asvnc persist() {
                                                                                                                                                                                                                                                                                                     gallery.appendChild(product); }
         <input type="text" name="prezime</pre>
                                                                  (async () => { let user =
                                                                                                                                       <u>Server.js</u>
const express = require('express');
id="prezime" size="30">
                                                                                                                                                                                                                     let userID = await dbNewUser(this)
                                                                                                                                                                                                                                                                                                                                                            Napišite CSS izraz kojim ćete odabrati sve elemente neporedane
                                                                     await User.fetchByUsername(req.body.user);
                                                                                                                                                                                                                  this.id = userID

} catch(err) {

console.log("ERROR persisting user data: " +
     </div>
                                                                                                                                                                                                                                                                                                                                                            liste koji imaju id (bilo kakav) ili imaju klasu "zeleno" te im postaviti
                                                                    if(user.id === undefined || !user.checkPassword
                                                                                                                                                                                                                                                                                                   updateCart();
                                                                                                                                        onst app = express();
     <div>
                                                                                                                                        const path = require('path');
const pg = require('pg')
                                                                                                                                                                                                                                                                                                                                                             boju pozadine na zeleno.
                                                                     (req.body.password)){
                                                                                                                                                                                                                                                                                                   getData();
        <label for="email">Email:</label>
                                                                                                                                                                                                                                                                                                                                                            ul>li[id], ul>li.zeleno {
                                                                                                                                                                                                                                                                                                 </script>
        <input type="text" name="email" id="email"</pre>
                                                                                                                                                                                                                                                                                                                                                             background-color: green; }
                                                                                                                                                                                                                 JSON.stringify(this))
                                                                         title : 'Log in',
linkActive : 'login',
                                                                                                                                       const db = require('./db')
                                                                                                                                                                                                                                                                                                 Primier obećania:
eize="30">
                                                                                                                                                                                                                                                                                                 let promise = new Promise(function(resolve.
                                                                                                                                       const db = require('./db')

const session = require('express-session')

const pgSession = require('connect-pg-
                                                                                                                                                                                                                     throw err } }
                                                                                                                                                                                                                                                                                                                                                             Koristeći slijedni dijagram nacrtajte scenarij u kojem korisnik pristupa
     </div>
                                                                                                                                                                                                                                                                                                                                                            stranici za registraciju na sljedeći način: • prvo unosi neispravno
(prazno) ime (što rezultira neuspješnom registracijom) • zatim
                                                                                                                                                                                                                 //dohvat korisnika iz baze podataka na osnovu korisničkog im
                                                                         user : req.session.user,
                                                                                                                                                                                                                                                                                                  eject) {
        <input type="submit" value="Submit">
                                                                                                                                                                                                                 (stupac user_name)
                                                                         err : 'Incorrect username or password.'));
                                                                                                                                        imple') (session)
   </fieldset>
                                                                                                                                                                                                                 dbGetUserBvName = asvnc (user name) =>
                                                                                                                                        /middleware - predlošci (ejs)
                                                                                                                                                                                                                                                                                                   Math.floor(Math.random() * 10);
                                                                                                                                                                                                                                                                                                                                                            popravlia ime, ali unosi e-mail koji nije jedinstven (što rezultira
                                                                       return; }
</form>
                                                                                                                                                                                                                 const sql = 'SELECT id, user_name, first_name, last_name, email, password, role FROM users WHERE user name = '' +
                                                                                                                                                                                                                                                                                                                                                             neuspješnom registracijom) • konačno, unosi ispravne podatke
                                                                       reg.session.user = user;
                                                                                                                                       app.set('views', path.join( dirname, 'views'));
                                                                                                                                                                                                                                                                                                   console.log(randomNumber);
"Napišite kôd na poslužitelju (samo metodu kontrolera, nije
                                                                                                                                        app.set('view engine', 'ejs');
//middleware - statički resursi
                                                                                                                                                                                                                                                                                                   if(randomNumber%2==1){
    setTimeout(() => {
                                                                       res.redirect('/');
                                                                                                                                                                                                                                                                                                                                                             uspješno se registrira.
potrebno usmjeravanje/routing) koji obrađuje zahtjev koji se šalje
                                                                                                                                                                                                                 user_name + `';
iz zadatka (b), ti, obavlia opisanu validaciju i registrira korisnika
                                                                                                                                       app.use(express.static(path.join( dirname
                                                                                                                                                                                                                                                                                                      resolve ("An odd random number was
                                                                                                                                                                                                                                                                                                                                                                                                         Baza podataka
ako je validacija uspješna, pritom unoseći novi zapis u bazu
                                                                                                                                                                                                                     const result = await db.query(sql, []);
                                                                                                                                                                                                                                                                                                                                                             preglednik
                                                                                                                                                                                                                                                                                                                                                                                    poslužiteli
nodataka (strukturu jednostavne tablice korisnika osmislite sami)
                                                              Pretpostavimo da želimo provieriti iedinstvenost e-mail adrese bez
                                                                                                                                                                                                                     return result.rows;
                                                                                                                                                                                                                                                                                                                                                                        GET /register
                                                                                                                                const homeRouter = require('./routes/home.routes');
                                                                                                                                                                                                                                                                                                     }, 5000);
router.post('/', async function(req, res) {
                                                              predavanja cijelog obrasca i u tu svrhu dodajemo gumb "Provjeri
                                                                                                                                                                                                                   } catch (err) {
                                                                                                                                const orderRouter = require('./routes/order.routes');
                                                                                                                                                                                                                                                                                                                                                                       200: HTML stranica s
                                                               e-mail" koji treba korisniku dojaviti je li e-mail slobodan ili ne.
                                                                                                                                                                                                                      console.log(err);
                                                                                                                                                                                                                                                                                                     setTimeout(() => {
                                                                                                                                const loginRoute = require('./routes/login.routes');
const logoutRoute = require('./routes/logout.routes')
   if (reg.body.ime && reg.body.prezime &&
                                                               Objasnite što je potrebno napraviti u kojem sloju da bi se to
                                                                                                                                                                                                                throw err } );
//dohvat korisnika iz baze podataka na osnovu id korisnika
                                                                                                                                                                                                                                                                                                     reject(new Error("This is generated from
      req.body.ime.length > 2 &&
                                                               ostvarilo. Napišite kôd koji je potrebno dodati na klijentskoj strani
                                                                                                                                                                                                                                                                                                      the promise executor!"));
                                                                                                                                const signupRoute = require('./routes/signup.routes'); ...
      req.body.prezime.length > 2) {
                                                                                                                                                                                                                                                                                                                                                                        POST /register
                                                              (preglednik). Pretpostavite da poslužitelj vraća JSON s jednim
boolean podatkom (oznakom je li adresa valjana ili ne).
                                                                                                                                                                                                                 (stupac id)
                                                                                                                                app.use(session( {
                                                                                                                                                                                                                                                                                                     1, 1000);
    let rsp = await db.query(`SELECT COUNT(*) as
                                                                                                                                                                                                                 dbGetUserById = async (user_id) =>
                                                                                                                                   store : new pgSession(
                  cnt FROM users WHERE email = $1`,
                                                                                                                                                                                                                                                                                                                                                                       200: HTML stranica s
                                                                                                                                                                                                                  const sql = `SELECT id, user_name, first_name, last_name,
mail, password, role
                                                               <button id="btnCheck" type="button"> check
                                                                                                                                    { pool : db.pool,
  tableName : 'session'}),
  secret : '4. Labos',
                                                                                                                                                                                                                                                                                                                                                                       formom za registraciji
i porukom o grešci –
prazno ime
                  [reg.body.email]);
                                                                                                                                                                                                                                                                                                   function(error) { console log(error) :}) then(
    if (rsp[0].cnt > 0)
                                                                                                                                                                                                                   FROM users WHERE id = ` + user_id;
                                                               <script>
                                                                                                                                                                                                                                                                                                      function(result) { console.log("Resolve:"
      res.render('register', {
                                                                  document.getElementBvId("btnCheck").onclick =
                                                                                                                                       resave : false,
saveUninitialized : true,
                                                                                                                                                                                                                                                                                                     result) },
                                                                                                                                                                                                                                                                                                                                                                        POST /register
         msg: 'Email nije jedinstven.' });
                                                                  async function() {
   let response = await fetch
                                                                                                                                                                                                                     const result = await db.query(sql, []);
                                                                                                                                                                                                                                                                                                     function(result) / console log("Reject:" +
                                                                                                                                                                                                                                                                                                                                                                                               provjera je li email
                                                                                                                                cookie : {maxAge : 60 * 60 * 24 * 1000} } ));
app.use((req, res, next) => {
                                                                                                                                                                                                                     return result.rows;
                                                                                                                                                                                                                                                                                                     result) }
      await db.covery('INSERT INTO user(ime
                                                                       ('http://localhost:3000/register
                                                                                                                                                                                                                     catch (err) {
       prezime, email) VALUES ($1,$2,$3)`
[req.body.ime, req.body.prezime,
                                                                                                                                                                                                                     console.log(err);
                                                                                                                                                                                                                                                                                                OBEĆANJA
                                                                        /check?email=' +
                                                                                                                                                                                                                                                                                                                                                                       200: HTML stranica s
                                                                                                                                   if (req.session.cart === undefined) {
                                                                        document.getElementById("email").value);
                                                                                                                                                                                                                     throw erril
                                                                                                                                                                                                                                                                                                                                                                       formom za registraci
i porukom o grešci -
email nije jedinstven
                                                                                                                                                                                                                                                                                                  na njima "registriramo" metode:
                                                                                                                                       req.session.cart = cart.createCart();}
next(); });
         reg.body.email1);
                                                                      if (response.ok) {
                                                                                                                                                                                                                 //umetanje zapisa o korisniku u bazu podataka
                                                                                                                                                                                                                                                                                                   .then → aktivira se ako je izvršna funkcija vratila uspjeh ili
      res.render('ok'); }
                                                                        let jsonResp = await response.json();
                                                                                                                                                                                                                constsud; = async (user) >> {
    const sql = "INSERT INTO users (user name, first_name,
    last_name, email, password, role) VALUES ("" + user_user_name
", "" + user_first_name + "", "" + user_last_name + "", "" +
    user_cemail + "", "" + user_password + "", "" + user_role + "")
    RETURNING id";
                                                                                                                                                                                                                                                                                                          neuspjeh
                                                                                                                                //definicija ruta
                                                                        if (jsonResp.isUnique)
 l alea I
                                                                                                                                                                                                                                                                                                                                                                        POST /registe
                                                                                                                                                                                                                                                                                                  .catch → aktivira se samo ako ie izvršna funkcija vratila
                                                                                                                                app.use('/', homeRouter);
                                                                                                                                                                                                                                                                                                                                                                                               SELECT.
    res.render('register', {
                                                                          alert("You're fine");
      msg: 'Ime i prezime moraju imati barem dva
                                                                                                                                app.use('/order', orderRouter);
                                                                                                                                                                                                                                                                                                          neuspjeh
                                                                                                                                                                                                                                                                                                                                                                                               provjera je li emal
                                                                        } else {
                                                                                                                                                                                                                                                                                                   finally → aktivira se ako je izvršna funkcija vratila uspjeh ili
                                                                                                                                app.use('/login', loginRoute);
      znaka.' }); }
                                                                           alert("Not unique, sorry."); }
                                                                                                                                                                                                                                                                                                    neuspjeh (ali za razliku od then ne daje mogućnost obrad
                                                                                                                                app.use('/logout', logoutRoute);
                                                                                                                                                                                                                                                                                                                                                                                               INSERT...
                                                                      } else {
                                                                                                                                                                                                                                                                                                    uspjeha ili neuspjeha)
                                                                                                                                app.use('/signup', signupRoute);
   res.render('register', {
                                                                        alert("HTTP-Error: " + response.status); }
                                                                                                                                                                                                                                                                                                                                                                                               unos novog
                                                                                                                                                                                                                                                                                                 ASINKRONOST
      msg: 'Došlo je do pogreške prilikom
registracije.' }); }
                                                                                                                                                                                                                     const result = await db.query(sql, []);
                                                                                                                                                                                                                                                                                                                                                                                               korisnika
                                                                                                                                                                                                                                                                                                Što se sve obavlja asinkrono? → IO operacije (dohvat podat
                                                                                                                                app.listen(3000);
                                                               </script>
                                                                                                                                                                                                                     return result rows[0] id:
                                                                                                                                                                                                                                                                                                 preko mreže, čitánje i pisanje na disk, ...) ; ovisi i o
                                                                                                                                                                                                                                                                                                  peracijskom sustavu ; ovisi i što programer koristi
                                                                                                                                                                                                                                                                                                                                                                       stranica s porukom o
                                                                                                                                                                                                                     console.log(err);
```

throw err 1 1s