

## Spirala 3

Kreirajte projekat na sljedeći način:

- U folderu projekta, u komandnoj liniji ukucajte **npm init**, package name neka bude: wt2018, ostalo ostavite na početnim vrijednostima
- Sve dodatne pakete koje koristite instalirajte lokalno u folderu projekta (u komandnoj liniji u folderu projekta **npm install PAKET --save**. *Ukoliko neki paket nismo koristili na vježbama, a želite ga koristiti postavite pitanje u forum da li je dozvoljen.*
- Provjerite da li se paket nalazi u package.json fajlu
- Dodajte .gitignore fajl u vaš projekat [link](#)
- Pokretanjem **node index.js** komande sve funkcionalnosti koje implementirate u nastavku trebaju raditi, port na kojem oslušujete zahtjeve treba biti 8080

### Zadatak 1. (0.5B)

Odgovor servera na GET zahtjev koji je upućen na url <http://localhost:8080/nazivDokumenta.html> je html dokument kojeg ste pravili na prvoj spirali, a nazivDokumenta može biti svaki od naziva dokumenata sa prve spirale. Stranica se treba ispravno prikazati (sa slikom, css stilom i js skriptama koje ste imali ranije).

### Zadatak 2\*. (2.5B)

Ukoliko se uputi zahtjev sa forme za unos zadatka sa poljima [name=naziv, type=text], [name=postavka, type=file], opciju za odabir vježbe obrišite (neće vam trebati). Za enkodiranje podataka u formi odaberite multipart/form-data. Za obradu multipart podataka možete koristiti multer paket. Nakon što se ovakav POST zahtjev uputi na <http://localhost:8080/addZadatak>, server će odgovoriti sa novom stranicom greska.html (gdje ćete ispisati navedenu poruku i omogućite korisniku da se vrati na raniju formu) ukoliko tip fajla nije pdf ili ako zadatak sa istim imenom već postoji. Ukoliko zadatak ne postoji kreira se fajl **nazivZad.json** gdje je **naziv** naziv iz forme i **naziv.pdf** fajl sa postavkom. U json fajlu se treba nalaziti {naziv:NAZIV,postavka:LINK\_ZA\_DOWNLOAD\_POSTAVKE}.

### Zadatak 3. (1.5B)

Ako se uputi GET zahtjev na <http://localhost:8080/zadatak> sa url parametrom **naziv** kao rezultat vraća se pdf fajl sa postavkom. Napravite testne zadatke sa nazivima Zadatak1, Zadatak2 i Zadatak3, a postavka tih zadataka neka bude ovaj dokument. Za zadatke dodane kroz formu vratite odgovarajuću postavku.

### Zadatak 4. (1.25B)

Ako se uputi POST zahtjev iz forme za unos godine sa poljima [name=nazivGod, type=text], [name=nazivRepVje, type=text] i [name=nazivRepSpi, type=text] na url <http://localhost:8080/addGodina> u fajl **godine.csv** na serveru se treba dodati novi red sa navedenim podacima. Ukoliko godina sa nazivGod već postoji korisniku vratite greska.html, a ukoliko ne postoji vratite addGodina.html.

#### Zadatak 5. (0.75B)

Ako se uputi GET zahtjev na <http://localhost:8080/godine> vraća se JSON niz svih dodanih godina u formatu [{nazivGod: ### ,nazivRepVje: ###, nazivRepSpi: ### }, ...]

#### Zadatak 6. (1B)

Napravite modul GodineAjax:

```
var GodineAjax = (function(){
    var konstruktor = function(divSadrzaj){
        return {
            osvjezi:function(){}
        }
    }
    return konstruktor;
})();
```

Pozivom konstruktora pravi se zahtjev koristeći XMLHttpRequest na <http://localhost:8080/godine>. U divSadrzaj upisuju se godine koje izgledaju kao na spirali 1, a podaci unutar godine su naziv godine, naziv repozitorija vjezbi i naziv repozitorija spirale.

Metoda osvjezi ponovo šalje zahtjev i osvježava sadržaj diva divSadrzaj.

#### Zadatak 7. (2B)

Ukoliko se uputi GET zahtjev na <http://localhost:8080/zadaci> vraća se spisak svih zadataka u formatu json, xml ili csv u zavisnosti koji je traženi format u headeru Accept. Ukoliko se više formata nalaze u navedenom headeru vratite odgovor u jednom gdje je prioritet u sljedećem redosljedu json,xml pa csv. Npr. ako se traže formati csv,xml vi vratite xml, ako se traze json,xml,csv vratite json i sl.

Primjer formata:

Naziv formata	Primjer:
JSON	[{naziv: "Zadatak 1" postavka: "/zadatak1.pdf"}, ... ]
XML	<?xml version="1.0" encoding="UTF-8"?> <zadaci> <zadatak> <naziv> Zadatak 1 </naziv> <postavka> /zadatak1.pdf </postavka> </zadatak> <zadatak> <naziv> Zadatak 2 </naziv> <postavka> /zadatak2.pdf</postavka> </zadatak> </zadaci>
CSV	Zadatak1,/zadatak1.pdf Zadatak2,/zadatak2.pdf

### Zadatak 8\*. (1B)

Napravite modul ZadaciAjax

```
var ZadaciAjax = (function(){
    var konstruktor = function(callbackFn){
        return {
            dajXML:function(){},
            dajCSV:function(){},
            dajJSON:function(){},
        }
    }
    return konstruktor;
})();
```

Metode dajXML, dajCSV i dajJSON prave GET zahtjev koristeći XMLHttpRequest na url iz prethodnog zadatka. Kada se dobije odgovor sa servera pozovite callbackFn i kao parametar proslijedite tijelo odgovora sa servera. Ukoliko se pozove više metoda,, a odgovor nije stigao pozovite callbackFn sa parametrom {greska:"Već ste uputili zahtjev"}. Ako odgovor za neku od metoda ne stigne u roku od 2 sekunde prekinite taj zahtjev.

Zadaci označeni sa \* su nešto teži. Pokušajte prvo uraditi ostale zadatke. Rok za rad na spirali je 29.12.2018 do 23:59. Review spirale 3 nosi 1B i biće otvoren nakon isteka roka spirale. Spirala 3 sa review-om nosi 11.5B. Implementaciju spirale 3 trebate postaviti na branch spirala3.