

# Promises - stanja i metode

Promisi mogu biti u nekoliko stanja:

- pending, što je početno stanje i znaci da promise nije niti ispunjen niti odbijen,
- fulfilled, što znaci da je promise uspješno izvršen,
- rejected, što znaci da je nešto prošlo loše i da promise nije ispunjen.

Kranji status je uvijek fulfilled ili rejected - nas kod treba da stremi da ovome, pending nas obično je interesuje.

Promisi se mogu nadovezivati jedna na drugi i na taj način se izvršavaju u "lancu". Međutim, bitna stvar kod promisa jeste da se mogu započeti istovremeno. Ukoliko imamo takav scenario, promisi se tehnički završavaju brže, jer se ne čeka sa jednog na drugi.

Za istovremeni izvršavanje više promisa, možemo koristiti različite strategije u zavisnosti od potrebe. Ove strategije se odnose na sledeće metode:

- Promise.all([arrayOfPromises]) - ispunjava se kada su svi promisi iz arrayOfPromises fulfilled, odbija se kada je bilo koji promise rejected

- Promise.any([arrayOfPromises]) - ispunjava se kada je bar jedan promise iz arrayOfPromises fulfilled, odbija se kada su svi promisi rejected,

- Promise.allSettled([arrayOfPromises]) - ispunjava se kada su svi promisi započeti, ne čekajući na rezultat,

- Promise.race([arrayOfPromises]) - ispunjava se kada je prvi završeni promise fulfilled, odbija se kada je prvi izvršeni promise rejected (faktički dobijamo status na osnovu prvog završenog promise-a),

finally - je metoda koja izvršava promise bez obzira da li je originalni promise bio fulfilled ili rejected

Kompletan spisak metoda i karakteristika vezanih za promise možete pročitati na:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)

#### Zadatak 1:

Napisati funkciju koja asinhrono ucitava tri razlicita fajla i ispisuje u konzoli sadrzaj prvog fajla koji se ucita

#### Zadatak 2:

Napisati funkciju `startJob` koja prima parametar `numberOfTasks` i vraca `promise` koji se "ispunjava" nakon 2 sekunde od starta sa porukom: "[broj\_takova] tasks finished". Funkcija vraca `reject` sa greskom "Invalid input" ako `numberOfTasks` nije broj ili nije prosledjen.