

به نام خدا

تحقیق میان ترم برنامه نویسی وب

علی احمدی

جعفر صادقی

در این مقاله بر انیم که در لینوکس یک اپلیکیشن نود را به همراه سلری پیاده سازی کنیم.
در ابتدا می خواهیم بدانیم که سلری چیست؟

سلری یک صف غیرهمزمان () برای انجام کارهای ما است. به عبارتی دیگر ما یک سری کار به سلری می دهیم و این برنامه آن ها را به صف تبدیل کرده و یکی پس از دیگری انجام می دهد. از مزایای دیگر سلری میتوان به برنامه ریزی کردن یک کار برای زمان خاص اشاره کرد که در موارد زیادی لازم و ضروری است.

حال یک برنامه نود می نویسیم که بتوانیم رفتار سلری را مشاهده کنیم و کارهای غیرهمزمان را در آن وارد کنیم.
مرحله اول ابتدا برنامه را ایجاد کرده و کتابخانه های مورد نیاز را به آن اضافه می کنیم.

```
$ npm init
```

```
$ npm install express
```

```
$ npm install celery-node
```

توجه شود از اکسپرس فقط برای راه اندازی یک وب سرور ساده و دیدن بهتر نتایج استفاده خواهد شد.

حالا در فایل package.json در بدنه Scripts یک خط به صورت زیر اضافه می کنیم تا بتوانیم برنامه را اجرا کنیم.

```
"develop": "node index.js"
```

می توان خطوط دیگر scripts را حذف نمود.

حال فایل index.js را تعریف می کنیم و یک وب سرور ساده می نویسیم.

```
const express = require('express')  
const app = express()
```

```
const port =
```

```
app.get('/add-task', (req, res) => {  
  res.send({})  
})
```

```
app.listen(port, () => {  
  console.log("Node Started at port " + port)  
})
```

حالا میتوان با دستور

```
npm run develop
```

برنامه را اجرا کرد و از نصب درست آن اطمینان حاصل کرد.

سپس به اتصال سلری با برنامه نوشته شده می پردازیم.
ابتدا باید در سیستم خود سرور ربیت را راه اندازی کنیم که میتوان با استفاده از داکر این کار را به سرعت انجام داد

```
$ docker run -it --rm --name rabbitmq -p : -p  
: rabbitmq: . -management
```

حال میتوان با استفاده از پکیج node-celery که نصب کردیم با ربیت ارتباط بگیریم.

```
const celery = require('celery-node');
```

```
const celery_client = celery.createClient(  
  "amqp://",  
  "amqp://"  
);
```

سپس میتوان مسیری که برای ایجاد تسک درست کرده ایم را تکمیل کنیم:

```
app.get('/add-task', (req, res) => {  
  const num1 = Number(req.query.num1);  
  const num2 = Number(req.query.num2);
```

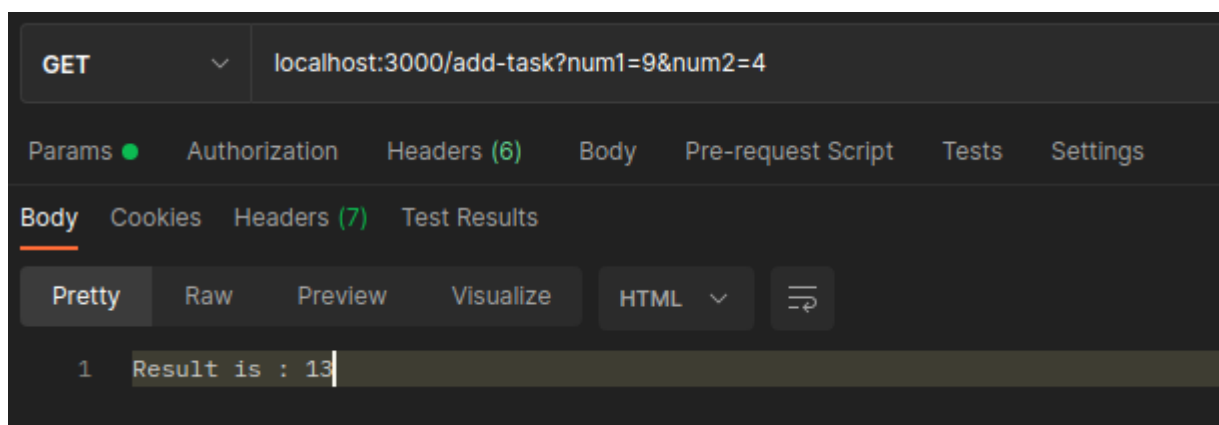
```
  const task = celery_client.createTask("tasks.add");  
  const result = task.applyAsync([num1, num2]);  
  result.get().then(data => {  
    res.send("Result is : " + data);  
  });  
})
```

در این مسیر ابتدا از پارامترهای num1 و num2 که در درخواست GET می آید دو مقدار موردنظر را میخوانیم و با استفاده از صف سلری این دو مقدار را با هم جمع می کنیم.

حال باید یک اتصال برای کارگر برقرار کرد تا تسک را از ربیت بخواند و کار مورد نظر را انجام دهد:

```
const worker = celery.createWorker(  
  "amqp://",  
  "amqp://"  
);  
worker.start();
```

حالا میتوان با استفاده از مسیر `add-task` یک کار جدید به سلری اضافه نمود و کارگر از ریت این کار را میخواند و انجام می دهد.



توجه شود که در این پروژه یک کار ساده مانند انجام یک جمع مورد نظر ما بود و میتوان بسته به پروژه شما کارهایی که با سلری انجام می شود پیچیده تر شود. کد نهایی در [اینجا](#) موجود هست.