

RESTful Kaynakları

- GET /api/node
- POST /api/node
- GET /api/node/<name>
- PUT /api/node/<name>
- POST /api/node/<name>/refresh
- DELETE /api/node/<name>
- GET /api/task
- POST /api/task
- GET /api/task/<id>
- DELETE /api/task/<id>
- GET /api/report/<id>/<format>
- GET /api/pcap/<id>

GET /api/node

Etkin olan tüm nodeları döndürür. Her node için bilgiler ilişkili adı, API URL'sini ve makineleri içerir:

```
$ curl http://localhost:9003/api/node
{
  "success": true,
  "nodes": {
    "localhost": {
      "machines": [
        {
          "name": "cuckoo1",
          "platform": "windows",
          "tags": []
        }
      ],
      "name": "localhost",
      "url": "http://localhost:8090/"
    }
  }
}
```

POST /api/node

Adı ve URL'yi belirterek yeni bir cuckoo node kaydedin:

```
$ curl http://localhost:9003/api/node -F name=localhost \
-F url=http://localhost:8090/
{
  "success": true
}
```

GET /api/node/<name>

Belirli bir cuckoo node hakkında temel bilgileri alın:

```
$ curl http://localhost:9003/api/node/localhost
{
  "success": true,
  "nodes": [
    {
      "name": "localhost",
      "url": "http://localhost:8090/"
      "machines": [
        {
          "name": "cuckoo1",
          "platform": "windows",
          "tags": []
        }
      ]
    }
  ]
}
```

PUT /api/node/<name>

Bir cuckoo node hakkındaki temel bilgilerini güncelleyin:

```
$ curl -XPUT http://localhost:9003/api/node/localhost -F name=newhost \
-F url=http://1.2.3.4:8090/
{
  "success": true
}
```

POST

/api/node/<name>/refresh

Cuckoo node tarafından ilişkilendirilen, özellikle de makineleri içeren metadata'yı yeniler:

```
$ curl -XPOST http://localhost:9003/api/node/localhost/refresh
{
  "success": true,
  "machines": [
    {
      "name": "cuckoo1",
      "platform": "windows",
      "tags": []
    },
    {
      "name": "cuckoo2",
      "platform": "windows",
      "tags": []
    }
  ]
}
```

DELETE /api/node/<name>

Bir cuckoo node devre dışı bırakın, bunun sonucunda yeni görevleri işleyemez, ancak geçmişini cuckoo distributed veritabanında tutar:

```
$ curl -XDELETE http://localhost:9003/api/node/localhost
{
  "success": true
}
```

GET /api/task

Veritabanındaki tüm görevlerin bir listesini alın. Sonuçları sınırlamak için bir offset, limit, finished ve owner alanı bulunmaktadır:

```
$ curl http://localhost:9003/api/task?limit=1
{
  "success": true,
  "tasks": {
    "1": {
      "clock": null,
      "custom": null,
      "owner": "",
      "enforce_timeout": null,
      "machine": null,
      "memory": null,
      "options": null,
      "package": null,
      "path": "/tmp/dist-samples/tmphal8mS",
      "platform": "windows",
      "priority": 1,
      "tags": null,
      "task_id": 1,
      "timeout": null
    }
  }
}
```


POST /api/task

Analiz edilmek üzere yeni bir dosya veya URL gönderin:

```
$ curl http://localhost:9003/api/task -F file=@sample.exe
{
  "success": true,
  "task_id": 2
}
```

GET /api/task/<id>

Belirli bir görev hakkında temel bilgiler edinin:

```
$ curl http://localhost:9003/api/task/2
{
  "success": true,
  "tasks": {
    "2": {
      "id": 2,
      "clock": null,
      "custom": null,
      "owner": "",
      "enforce_timeout": null,
      "machine": null,
      "memory": null,
      "options": null,
      "package": null,
      "path": "/tmp/tmpPwUeXm",
      "platform": "windows",
      "priority": 1,
      "tags": null,
      "timeout": null,
      "task_id": 1,
      "node_id": 2,
      "finished": false
    }
  }
}
```

DELETE /api/task/<id>

Bir görevin tüm ilişkili verilerini, yani binary dosyayı, PCAP'i ve raporları silin:

```
$ curl -XDELETE http://localhost:9003/api/task/2
{
  "success": true
}
```

GET

/api/report/<id>/<format>

Belirtilen formatta verilen görev için bir rapor alın:

```
# Defaults to the JSON report.  
$ curl http://localhost:9003/api/report/2  
...
```

GET /api/pcap/<id>

Verilen görev için PCAP'i alır:

```
$ curl http://localhost:9003/api/pcap/2  
...
```