

5 dk da Telegram Botu nasıl yapılır?



Fatih Sarhan

Mar 12, 2018 · 3 min read

Kendi botumuzu oluşturmak için 2 adım gereklidir.

Birincisi **BotFather** ile kendi botumuzun hesabını oluşturmalıyız. (**BotFather** bir telegram botudur. Telegram botu oluşturmamızı kolaylaştırır. Bot oluşturmak için başka bir bot kullanıyoruz, çok iyi la:)).

İkincisi de Telegram'ın bot'a gelen mesajları yönlendireceği uygulamaya, uygulamadan gelen cevabı bot ile konuşan kişiye yönlendireceği sisteme ihtiyacımız var. Allah razı olsun bu sistemi yapan paketler yazılmış (**Nodejs** için paket, **Python** için paket). Bu paketler genel olarak Telegram'ın bot apisi ile haberleşir. Şunu unutmayın **BotFather** ile botu oluşturduğumuzda bot aktif değil. İkinci adım gerçekleştirildikten sonra aktif olacaktır. Aktiflikten kasıt gönderilen mesajları yorumlayıp, cevap göndermek.

Genel girişi yaptık, şimdi nasıl yapılacağını konuşalım.

. . .

1.Adım: (BotFather ile botun hesabını oluşturmak)

Kendi botumuzun hesabını oluşturmak için **BotFather** botu ile konuşmaya başlamalıyız. **BotFather** linkine tıklayınız, sonrasında **START** butonuna bastıktan sonra konuşmaya başlamış olacaksınız. Sizi **BotFather**'in yardım(help) mesajı karşılayacak. Bu mesajı okuyarak kendi botunuzu oluşturabilirsiniz.

/newbot komutunu mesaj olarak gönderdikten sonra sizden botun ismini isteyecektir. Suanlık biz **Benim Telegram Botum** ismini vericeğiz. Siz istediğiniz ismi verebilirsiniz. Sonrasında sizden telegram botunuz için kullanıcı ismi isteyecektir. Sadece 2 şartları var. Ya **Bot** ya da **_bot** ile bitmesidir. Örneğin **BenimTelegramBot**, **benimtelegram_bot** gibi kullanıcı adları kullanılmalıdır.

Kullanıcı adı seçtikten sonra **BotFather** size botunuzun telegram url'sini ve token değerini içeren mesaj atacaktır. İsterseniz botunuza özelliklerini değiştirebilir veya özellikler ekleyebilirsiniz. **/help** komutunu mesaj olarak gönderdiğinizde **BotFather** neler yapabileceğinizi yardım eden mesajı gönderecektir.

Şuan bot oluşturduk, ama bot şuan aktif değil. Yani botun çalışması için ikinci adımı gerçekleştirmeliyiz.

2.Adım: (Bot Uygulamamız)

Aslında burda istediğiniz bir dil ve o dildeki istediğiniz herhangi bir paketini kullanabilirsiniz. libraries.io sitesinden rahatça bulabilirsiniz. Ve Dökümanları genellikle çok iyi oluyor. Aslında bu kısımda anlatacaklarımız dokümanlardan bulabilirsiniz. Emin olun ki paketin dokümanı daha iyi:))

Python, Js, Golang, C# ...

Biz Şuanlık Nodejs de ki **node-telegram-bot-api** paketini kullanacağız. İlk olarak bot projemize başlamadan önce nodejs, npm ve yarn gibi araçları yüklemeyi unutmayın. Bu paketleri sitelerindeki dokümanlara göre kendi sisteminize indirebilirsiniz.

Her şey tamam ise bot projemiz için bir dizin oluşturuyoruz. Sonrasında dizinin içine girdikten sonra nodejs için **yarn** paket yöneticisi ile node-telegram-bot-api paketini indiriyoruz. Dizin içindeki dosyaları listeliyoruz. **package.json** projenin özelliklerini ve gerekli paketlerin listesini tutar. **yarn.lock** dosyası gerekli paketlerin versiyonunu, hangi siteden indirileceğini, paketin hash değerini, paketin için gerekli küçük paketleri tutar. **node_modules** dizinide paketlerin kaynak kodlarını tutar. Daha sonra index.js dosyası oluşturuyoruz.

```
$ mkdir telegram-bot-projem
$ cd telegram-bot-projem
$ yarn init
yarn init v1.5.1
question name (telegram-bot-projem):
question version (1.0.0):
question description: Telegram Bot Projem
question entry point (index.js):
question repository url:
question author: benim
question license (MIT): GPL-3.0
question private: public
success Saved package.json
Done in 30.31s.
```

```
$ yarn add node-telegram-bot-api
yarn add v1.5.1
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 82 new dependencies.
info Direct dependencies
└─ node-telegram-bot-api@0.30.0
info All dependencies
...
Done in 3.25s.
$ ls
node_modules  package.json  yarn.lock
$ touch index.js
```

Paketin README.md içindeki örneği index.js dosyasına kopyalayalım. Ve token değerini değiştirelim. Sonrasın uygulamamızı çalıştırılalım.

```
$ vim index.js
const TelegramBot = require('node-telegram-bot-api');

// Telegram Değerinize göre değiştirin
const token = 'YOUR_TELEGRAM_BOT_TOKEN';

// Telegram botu oluşturuyoruz,
// yeni mesajları almak için polling methodunu kullanırız.
const bot = new TelegramBot(token, {polling: true});

// eşleşir"/echo [herhangi]"
bot.onText(/\s/echo (.+)/, (msg, match) => {
  // 'msg' değeri telegram mesaj nesnesidir.
  // 'match' değeri herhangi kısımdaki değerdir.
  const chatId = msg.chat.id;
  const resp = match[1]; // (.+) kisminda uyuşan yazi alinir.

  // mesajı gönderen kişiye [herhangi] kısımdaki
  // mesaj gönderilir.
  bot.sendMessage(chatId, resp);
});

// Bütün mesajları dinler.
bot.on('message', (msg) => {
  const chatId = msg.chat.id;

  // mesaj atan kişiye mesaj göndeririz.
  bot.sendMessage(chatId, 'Mesajini aldım kardeş.');
```

```
$ node index.js
...
```

Test etmek için Botumuz ile konuşmaya çalışalım. Herhangi bir mesaj yazdığımızda ‘Mesajını aldım kardeş’ diye bir mesaj gönderecektir. Unutmayın bu her mesaj gönderdiğinizde çalışacaktır.“/echo naber bot?“ dediğimizde de bize ‘Mesajını aldım kardeş.’ ‘naber bot’ diye mesajlar gönderecektir.

Bundan sonra hangi paketi kullandıysanız onun dokümanlarından botunuzu geliştirebilirsiniz.

Bir sonraki yazı da telegram bot api paketi yazmaya çalışacağız.

[Telegram Bot](#) [Nodejs](#) [Turkish](#)

[About](#) [Help](#) [Legal](#)