



**TALLINNA TEHNIKAÜLIKOOL**  
INSENERITEADUSKOND  
Virumaa kolledž

**Interaktiivse vestlusboti loomine Slack API ja Jira REST  
API kasutades**

**Creating an interactive chatbot using the Slack API and Jira  
REST API**

RAKENDUSINFOTEHNOLOOGIA ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Ivan Panas

Üliõpilaskood: 154523

Juhendaja: N. Ivleva, lektor

# AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputööl alusel ei ole keegi teine varem katse-või inseneridiplomit taotletud.

Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

"....." ..... 202....

Autor: .....

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõutele

"....." ..... 202....

Juhendaja: .....

/ allkiri /

Kaitsmisele lubatud

"....." ..... 202....

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

# **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina Ivan Panas

(sünnikuupäev: 24.02.1991 )

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

Interaktiivse vestlusboti loomine Slack API ja Jira REST API kasutades

mille juhendaja on

Natalja Ivleva

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

---

<sup>1</sup>*Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil.*

\_\_\_\_\_ (allkiri)

\_\_\_\_\_ (kuupäev)

**Virumaa kolledž**  
**LÕPUTÖÖ ÜLESANNE**

**Üliõpilane:** Ivan Panas, 154523

Õppekava, peeriala: RDIR12 Rakendusinfotehnoloogia

Juhendaja(d): lektor, Natalja Ivleva, +372 336-3921

**Lõputöö teema:**

(eesti keeles) Interaktiivse vestlusboti loomine Slack API ja Jira REST API kasutades

(inglise keeles) Creating an interactive chatbot using the Slack API and Jira REST API

**Lõputöö põhieesmärgid:**

1. Luua Slack boti, mis teavitab töötajaid tulemusest Jira kaardi sulgemisel.

**Lõputöö etapid ja ajakava:**

Nr	Ülesande kirjeldus	Tähtaeg
1.	Tutvuda Slack API ja Jira REST API töötamise eripäradega.	15.10.2019
2.	Õppida projekti rakendamiseks kasutatavate botide loomise tehnoloogiaid.	10.11.2019
3.	Esitada saadud andmete põhjal oma projekti versioon.	15.12.2019

**Töö keel:** eesti

**Lõputöö esitamise tähtaeg:** ".....".....202....a

**Üliõpilane:** Ivan Panas

..... ".....".....202....a

/allkiri/

**Juhendaja:** Natalja Ivleva

..... ".....".....202....a

/allkiri/

**Programmijuht:** Žanna Gratšjova

..... ".....".....202....a

/allkiri/

# SISUKORD

1. SISSEJUHATUS .....	8
2. ANALÜÜS.....	10
2.1 Jira REST API ülevaade .....	10
2.2 Slack API ülevaade.....	10
2.3 Tehnoloogia ülevaade Slack botite loomiseks .....	11
2.4 Tehnoloogiate ja tööriistade valik .....	13
3. PROJEKTI RAKENDAMINE .....	14
3.1 Rakenduse loomine platvormil Slack.....	14
3.2 Botkit-põhise botirakenduse loomine .....	15
3.2.1 Yeomani malli installimine Botkiti jaoks.....	15
3.2.2 Teabe saamine Jira kaartide kohta .....	17
3.2.3 Slacki lõime ja Jira kaardi vahelise lingi lisamise meetod .....	17
3.2.4 Slack lõime ja Jira kaardi lahti ühendamise meetod .....	18
3.2.5 Yeoman malli installimine Botkiti jaoks.....	18
3.3 Botirakenduse juurutamine ettevõtte serverisse.....	19
3.4 Botirakenduse ühendamine platvormiga Slack .....	20
3.5 Jira platvormil teatiste seadistamine.....	21
3.6 Näited boti kasutamisest.....	22
3.7 Boti edasiarendamine .....	23
KOKKUVÕTE .....	25
SUMMARY.....	26
KASUTATUD KIRJANDUSE LOETELU .....	27
LISAD .....	29
Lisa 1 Slack rakenduse seadete avaleht.....	30
Lisa 2 Slack boti lisamine .....	31
Lisa 3 Kuvatud Slack rakendusteabe seadistamine .....	32
Lisa 4 Vajalikud load Slack rakenduseks .....	33
Lisa 5 Nupp rakenduse installimiseks Slack kontole.....	34
Lisa 6 Slack rakenduse installimise kinnituse leht .....	35
Lisa 7 Slack rakenduse ja boti load .....	36
Lisa 8 Slack sündmuste seadistamine.....	37
Lisa 9 Meetodid findIssues ja updateIssues.....	38
Lisa 10 Meetod AddThreadToIssue .....	39

Lisa 11 Meetod RemoveThreadFromIssue .....	40
Lisa 12 Jira veebikonksu töötlemismeetod .....	41
Lisa 13 Request URL ja sündmused.....	43
Lisa 14 Jira kohandatud väljad .....	44
Lisa 15 Jira kohandatud välja tüübi valimine .....	45
Lisa 16 Jira kohandatud välja nimi .....	46
Lisa 17 Jira veebikonksu lisamine .....	47
Lisa 18 Jira töövoore redigeerimise leht .....	48
Lisa 19 Jira pärastise funktsiooni lisamine.....	49
Lisa 20 Jira pärastise funktsiooni tüübi valimine .....	50

## Lühendite ja mõistete sõnastik

API	Rakendusliides (Application Programming Interface) on reeglid ja vahendid, mille abil programm suhtleb teiste programmidega ja süsteemiga [1]
ChatOps	Operatsiooniline paradigma, mis võimaldab meeskonnaliikmetel tippida vestlus ruumides käske, mida vestluse bot käib skriptide ja pistikprogrammide kaudu [7]
HTTP	Hüpertexti edastuse protokoll (HyperText Transfer Protocol) on veebi aluseks olev hüpermeediumile suunatud standardne rakenduskihi protokoll, mis määrab sõnumite vormingu, edastuse, veebiserverite ja brauserite käitumise [1]
IIS	Internet Information Server (IIS) on veebiserveri rakendus ja moodul-laineduste kogum, mis on loodud Microsofti poolt Microsoft Windowsi kasutamiseks [2]
REST	Esitusoleku siire (Representational State Transfer) on veebiteenuste programmeerimise paradigma [1]
URL Rewrite	IIS-i laiendustarkvara, mis võimaldab teostada URL-iga manipuleerimise ülesandeid [3]
Slack kanal	Kanal on meeskonna jaoks üks koht, kus saab jagada sõnumeid, tööriistu ja faile [4]
Slack lõim	Lõim on kanali sõnumile lisatud sõnumite grupp
Slack RTM API	Reaalajasõnumite (Real Time Messaging) API on WebSocketil põhinev API, mis võimaldab Slackilt sündmusi reaalajas vastu võtta ja kasutajatena sõnumeid saata [5]
Slack Web API	Veebi API on liides teabe pärimiseks ja Slack-i tööruumist muudatuste tegemiseks [6]

# 1. SISSEJUHATUS

Linnworks on rahvusvaheline e-kaubanduse rakenduste arendusettevõte.

Töötajate vaheliseks suhtlemiseks kasutatakse firmasiseselt rakendust Slack. See rakendus võimaldab töötajatel suhelda ja andmeid omavahel vahetada.

Arendusprotsessi korraldamiseks kasutab ettevõtte välearenduseks mõeldud rakendust - Jira. Selle abil saavad meeskonnad tarkvara kavandada, jälgida ja väljastada.

Linnworks kasutab Slacki ja Jira platvormide integreerimiseks kolmanda osapoole tasuta pistikprogramme ja rakendusi, kuid mõnda ülesannet ei saa nende abiga lahendada. Seetõttu pöördus ettevõtte autori poole ettepanekuga lahendada üks neist probleemidest. See seisneb selles, et ettevõtte töötajad soovivad kiiresti saada teavet Jira platvormi abil tehtud tööde lõpuleviimise kohta.

Autori töö eesmärk on luua Slack boti, mis teavitab töötajaid tulemusest Jira kaardi sulgemisel. Jira kaardi sulgemine tähendab sel juhul, et ülesanne on lõpule viidud või seda ei saa mingil kaardil kirjeldatud põhjusel lõpule viia.

Töö eesmärgi saavutamiseks seadis autor endale järgmised ülesanded:

- 1) tutvuda Slack API ja Jira REST API töötamise eripäradega;
- 2) õppida projekti rakendamiseks kasutatavate botide loomise tehnoloogiaid;
- 3) esitada saadud andmete põhjal oma projekti versioon.

Ettevõtte on autorile seadnud järgmised nõuded:

- bot saab sõnumeid lugeda ainult avalikest kanalitest;
- bot saab töötajaid teavitada ainult kanali lõimes;
- võimalus teatist tellida ja tellimusest loobuda;
- käsud, millele botile vaja vastata, peaksid olema võimalikult lihtsad, mitte rohkem kui 2-3 sõna;
- boti rakendus tuleb juurutada ettevõtte serveris Windows Server 2019 juhtimisega.

Sellega seoses uurib autor töö esimeses osas seda probleemi ja teises kirjeldab selle lahendust. Pärast seda teeb ta järeldused saadud tulemuse kohta ja projekti arenguperspektiividest.



Lõputöö märksõnad on: vestlusbot, Slack, Jira, Botkit, rakenduskõrgharidusõppe lõputöö.

## 2. ANALÜÜS

Selles peatükis uurib autor Slack API, Jira REST API ja olemasolevaid tehnoloogiaid Slack botite loomiseks. Uuringu käigus jõuab autor järeldusele Linnworksile uue Slack boti loomise kohta ning lisaks otsustab, milliste tehnoloogiate ja tööriistadega projekti rakendatakse.

### 2.1 Jira REST API ülevaade

Jira tarkvara peamine eesmärk on ülesannete jälgimine ja haldamine. Töötajatel on võimalus tööülesannetega töötada veebiliidese kaudu, mis võib asuda kas ettevõtte serveris (Jira Server) või pilves (Jira Cloud).

Lisaks veebiliidesele pakub Jira võimalust hallata ülesandeid ka programmi teel, nimelt REST API kaudu. Kuna Linnworks kasutab pilveversiooni, kaalub autor REST API Jira Cloud jaoks - Jira REST API. Autor pöörab sellele tähelepanu, kuna Jira-l on ka teisi REST-i API-sid: Jira tarkvaraserveri REST API ja Jira Service Desk Server REST API [8].

Jira REST API kasutamine toimub HTTP päringute saatmise kaudu. Autentimiseks kasutatakse põhimeetodit, seega peab päring sisaldama spetsiaalset päist kodeeritud kasutajanime ja parooliga. [8]

Kasutatud HTTP päringupäised:

- authorization;
- content-type. [8]

Kasutatud andmete liigid:

- application/json. [8]

Rakendus saab vastuseks sellisele taotlusele JSON-objekti.

### 2.2 Slack API ülevaade

Slack on pilvepõhine sõnumsideplatvorm. Lisaks tavakasutajatele saavad platvormis sõnumeid saata platvormi sisse installitud botid või rakendused. Selliste rakendustega suhtlemiseks pakub Slack avaliku veebi API-d. Selle abil saate saata sõnumeid ning saada mitmesugust rakendusele saadavat teavet.

Kuna Slack API ei ole REST API, töötavad kõik selle meetodid ainult HTTPS-i kaudu ja toetavad ainult GET- või POST-meetodeid. Autentimiseks kasutatakse OAuth 2.0, seetõttu peab päises või HTTP-päringu argumentide hulgas olema Bearer token. [6]

Kasutatud HTTP päringupäised:

- content-type;
- authorization. [6]

Parameetrite edastamiseks on kasutatud järgmiseid tüüpe:

- as query string;
- application/json;
- application/x-www-form-urlencoded. [6]

Kõik Slack API vastused tagastatakse JSON-vormingus ja sisaldavad järgmist teavet:

- edukad ja mitteedukad taotlused;
- vead, kui neid on;
- hoiatused, kui neid on;
- päringuga seotud teave. [6]

## 2.3 Tehnoloogia ülevaade Slack botite loomiseks

Selles osas võrdleb autor olemasolevaid tehnoloogiaid, mis võimaldavad luua ja arendada botid Slack platvormi jaoks. Nende nimekiri on esitatud erinevates programmeerimiskeeltes ja see on saadaval ametlikul Slacki veebisaidil [9]. Tabelis töö autor välja mõned neist, mida ta läbi vaatab.

Tabel 2.1 Tarkvarateekid

Nimi	Programmeerimiskeel	Omadused
SlackAPI	C#	<ul style="list-style-type: none"><li>• Web API</li><li>• Real Time Messaging API</li><li>• Authentication</li></ul>
Slack	Go	<ul style="list-style-type: none"><li>• Bot Users</li><li>• Web API</li><li>• Real Time Messaging API</li></ul>
Slack-client	Java	<ul style="list-style-type: none"><li>• Web API</li></ul>

Tabel 2.1 Tarkvarateekid järg

Nimi	Programmeerimiskeel	Omadused
Botkit	JavaScript	<ul style="list-style-type: none"> <li>• Bot Users</li> <li>• Real Time Messaging API</li> <li>• Web API</li> <li>• Authentication</li> <li>• Slack Apps</li> <li>• Message Buttons</li> </ul>
Slacker	Python	<ul style="list-style-type: none"> <li>• Incoming Webhooks</li> <li>• Web API</li> <li>• Authentication</li> </ul>

SlackAPI on C # avatud lähtekoodiga teek, mis on saadaval MIT-i litsentsi alusel. See võimaldab teil kasutada nii Web API kui ka RTM API. Lisaks toetab see tööd OAuth 2.0 pääsumärkidega. [10]

Slack on Go keeles kirjutatud avatud teek, mille lähtekood asub GitHubi hoidlas. See on saadaval BSD 2-klausli litsentsi alusel. Teek toetab töötamist mitte ainult veebi API ja RTM API, vaid ka Bot Users. [11]

Slack-klient on Ameerika ettevõtte Hubspot Java keeles kirjutatud väike teek. Lähtekood asub GitHubi hoidlas ja on saadaval kasutamiseks Apache-2.0 litsentsi all. Teek toetab ainult Web API põhimeetodeid. [12]

Botkit on Howdy avatud lähtekoodiga JavaScripti teek. Lähtekood asub GitHubi hoidlas ja on saadaval MIT-litsentsi all. Botkit on tööriistakomplekt vestlusbottide loomiseks, sealhulgas platvormi Slack jaoks. Teek võimaldab töötada mitte ainult Web API ja RTM APIga, vaid ka Slack botite ja rakendustega. Samuti on saadaval autentimine OAuth 2.0 juurdepääsulubade kaudu ja võimalus luua sõnumites interaktiivseid nuppe. [13] Selle teegi on ametlikult heaks kiitnud ja seda toetab Microsoft ning see on osa Microsoft Bot Frameworkist [14].

Slacker on Pythonis kirjutatud avatud teek, mille autor on Oktay Sancak. Selle kasutamiseks peate nõustuma Apache-2.0 litsentsiga. Teek toetab nii Web API-d kui ka autentimist OAuth 2.0 ligipääsulubade kaudu. [15]

Arvutinägemine jaguneb kaheks kihiks: kõrgetasemeline nägemine ja madalatasemeline nägemine. Madalatasemeline nägemine vastutab pilditöötuse, kontuuritöötuse, tagaplaani eemaldamise, muutuste detekteerimise, objekti jälgimise, suuruse muutmise eest. Kõrgetasemeline nägemine vastab objektide klassifitseerimisele, objektide asukoha või selle suuruse hindamisele.

## 2.4 Tehnoloogiate ja tööriistade valik

Autor vaatas üle Jira REST API ja Slack API põhipunktid, samuti 5 tehnoloogiat Slacki platvormi vestlusbottide loomiseks. Vestlusboti loomiseks valis ta Botkiti mitmel põhjusel.

Esiteks on teegi funktsioonide loend ulatuslik ja suurim. Lisaks API-ga töötamisele on autoril botite ja rakenduste teabele juurdepääs. Samuti saab ta luua interaktiivsete nuppude ja muude menüüelementidega sõnumeid, mida saab kasutada tulevaste ülesannete jaoks.

Teiseks on autoril selle teegiga töötamiseks piisavalt JavaScripti ja TypeScripti arenduskogemusi. Ja Yeomani malli kasutamine Botkiti rakenduste loomiseks võimaldab boti jaoks malliprojekti luua mõne väikese sammuga [16]. Seega lühendab oluliselt arendusaega.

Ja kolmandaks soovib autor märkida, et Botkiti toetab Microsoft ja see on juba Microsoft Bot Frameworkisse sissehitatud. See annab kindluse selle toote kvaliteedis ning selles, et seda saab edasi arendada ja parandada.

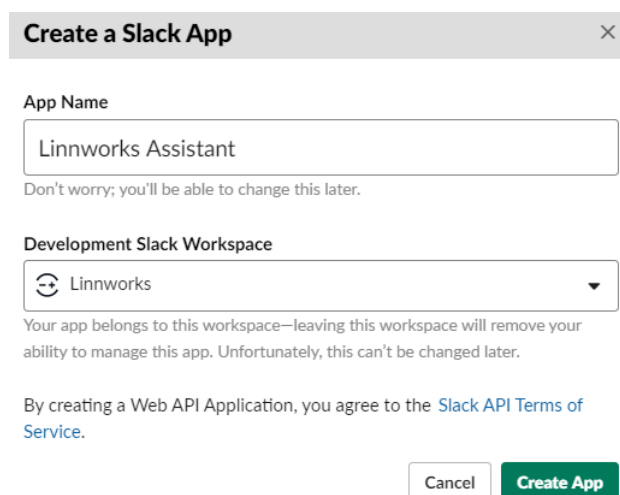
Koodi redigeerimise tööriistade hulgast valis autor kompaktsed ja kerged Visual Studio Code toimetaja, mida saab kasutada tasuta [17]. See võimaldab ühendada erinevaid pistikprogramme ja töötada hoidlatega.

### 3. PROJEKTI RAKENDAMINE

Selles osas kirjeldab autor Linnworksile uue Slack-boti loomist. Ta jagas boti loomise protsessi mitmeks etapiks, mida kirjeldatakse allpool. Samuti juhib ta lugeja tähelepanu asjaolule, et Linnworks kasutatud ja selles töös esitatud salajaseid võtmeid, paroole ja linke saab muuta või peita.

#### 3.1 Rakenduse loomine platvormil Slack

Rakenduse ja botite loomiseks kasutas autor Slacki veebilehe dokumentatsiooni [18]. Kuna bot on Slacki rakenduse osa, lisab see kõigepealt rakenduse ja seejärel boti. Lehel koos installitud rakenduste loendist autor klõpsas *Create New App*. Seejärel täitis vajalikud lahtrid, luges tingimused ja klikkis *Create App* [19]. Selles etapis valis autor rakendusele nimetus ja märkis konto, millele rakendus lisatakse (vt Joonis 3.1).



**Create a Slack App** ×

App Name

Don't worry; you'll be able to change this later.

Development Slack Workspace

 ▾

Joonis 3.1 Rakenduse Slack loomine

Pärast rakenduse loomist avaneb leht rakenduse sätetega (vt Lisa 1), millele lisaks seadete häälestamisele lisab autor ka boti. Selleks on olemas nupp *Bots, Add features and functionality* jaotises *Basic Information* lehel. Sellele klikkides avaneb leht, millel saate botit lisada või muuta. Uue boti loomiseks sisestas autor kuvatava nime ja vaikenime ning klõpsas nupule salvesta (vt Lisa 2).

Pärast boti lisamist rakendusele taasavas autor lehe põhiteabega. Selle kõige alumises osas lisas ta rakendusele lühikese kirjelduse, ikooni ja valis ikooni taustavärvi (vt Lisa 3). Sellel lehel on rakenduse ja boti salajased juurdepääsuklahvid, mida autor hiljem kasutab.

Rakenduse seadistamise juhendi alusel lisas autor talle vajalikud õigused. Selleks lisas neid ta *OAuth & Permissions* lehel *Scopes* jaotises, mis on vajalikud sõnumite lugemiseks ja saatmiseks (vt Lisa 4).

Samal lehel installis autor Slacki rakendusesse Linnworks konto. Selleks klõpsas ta nupul *Add App to Workspace* (vt Lisa 5) ja seejärel nupul *Allow* (vt Lisa 6). Pärast ettevõtte administraatori kinnitust rakendus oli lisatud. *OAuth & Permissions* lehel oli ilmunud boti OAuth tokenid (vt Lisa 7).

Nüüd on rakendus botiga installitud ja peaaegu töövalmis. Viimane samm on teatiste tellimise konfigureerimine, pärast seda saab bot lugeda kasutajate teatiseid, neid analüüsida ja töödelda. Selleks lisab autor sündmuste tellimuste lehele vajalikud sündmused ja *Request URL* (vt Lisa 8). Kus *Request URL* see on rakenduse Botkit lõppsõlm, mis käsitleb sündmuseid. Kuna autor pole rakendust veel lõpule viinud, konfigureeritakse sündmused hiljem.

## 3.2 Botkit-põhise botirakenduse loomine

### 3.2.1 Yeomani malli installimine Botkiti jaoks

Nagu autor juba töö esimeses osas kirjeldas, kasutab ta botirakenduse loomiseks malli Yeoman. Selleks tuleb arvutisse installida Node.js. Autor laadis Node.js alla ametlikult veebilehelt ja installis selle arvutisse [20].

Pärast töökeskkonna ettevalmistamist, avas autor Botkiti teegi ametliku veebilehe, kus ta klõpsas lingil *Get Started* ja järgis rakenduse loomise juhiseid [21].

Esiteks lõi autor rakenduse jaoks kataloogi ja seejärel installis Yeomani ja Botkit *npm* pakette. Selleks avas ta käsurea ja trükkis käsu (vt Joonis 3.2).

```
C:\>mkdir botkit_linnworks  
C:\>cd botkit_linnworks  
C:\botkit_linnworks>npm install yo generator-botkit
```

Joonis 3.2 *Npm* pakettide installimine rakenduste kataloogi

Pärast edukat installimist nägi autor käsurealt kinnitust (vt Joonis 3.3).

```

+ generator-botkit@4.6.1
+ yo@3.1.1
added 612 packages from 326 contributors and audited 4243 packages in 82.813s
found 0 vulnerabilities

C:\botkit_linnworks>

```

Joonis 3.3 Botkit ja Yeoman *npm* pakettide edukas installimine

Pärast seda lõi autor uue rakenduse Botkit. Selleks käivitas ta käsurealt ka spetsiaalse käsu ja täpsustas vajalikud andmed (vt Joonis 3.4).

```

C:\botkit_linnworks>yo botkit
? What is your bot's name? botkit_linnworks
? Which messaging platform? slack
? (Optional) Mongo URI for state persistence:
? Client signing secret (from api.slack.com)
? Configure for single-team or multi-team? single
? Bot Token

```

Joonis 3.4 Botkit rakenduse installimine Yeoman malli abil

Rakenduse käivitamiseks kasutas autor käsku *npm start* (vt Joonis 3.5). Sel juhul kasutab rakendus vaikimisi porti 3000, aga lõppsõlm sõnumite töötlemiseks */api/messages*.

```

C:\botkit_linnworks>npm start

> botkit-botkit_linnworks@1.0.0 start C:\botkit_linnworks
> node bot.js

*****
* WARNING: Your adapter may be running with an incomplete/unsafe configuration. *
* - Ensure all required configuration options are present                       *
* - Disable the "enable_incomplete" option!                                    *
*****

** Your bot is using memory storage and will forget everything when it reboots!
** To preserve dialog state, specify a storage adapter in your Botkit config:
** const controller = new Botkit({storage: myStorageAdapter});
Enabling plugin: Slack Adapter
Webhook endpoint online: http://localhost:3000/api/messages

```

Joonis 3.5 Botirakenduse käivitamine

Vaikepordi muutmiseks lisas autor *.env* konfiguratsioonifaili käsk *PORT = 3001*. Pärast seda rakendust kasutatakse porti 3001.

Vaikimisi loodi 4 faili koos botifunktsioonide näidetega:

- *slack\_features.js*;
- *sample\_hears.js*;
- *sample\_echo.js*;
- *routes\_oauth.js*.





Ühenduse loomiseks Slacki lõime ja Jira kaardi vahel löi autor sõnumite filtreerimiseks kaks malli: *linkIssueKeyRegExp* ja *linkIssueUrlRegExp*. Selle töötleja abil töödeldakse ainult kasutajate sõnumeid, mis sisaldavad *link* märksõna ja kaardi nime (vt Joonis 3.7). Lisaks kaardi nimele saavad kasutajad kasutada ka kaardi linki. Kanali lõime sees saadetavatel sõnumitel on *thread\_ts* atribuut, seega lisas autor selle atribuudi kontrolli. *AddThreadToIssue* meetodi täielik kood on toodud lisas 10.

### 3.2.4 Slack lõime ja Jira kaardi lahti ühendamise meetod

Ühenduse eemaldamiseks lisas autor kahe malliga käitlejat, mis lisaks kaardi nimele (või lingile sellele) sisaldab märksõna *unlink* (vt Joonis 3.8). Selle töötleja abil töödeldakse ainult kasutajate sõnumeid, mis algavad linkimise tühistamise märksõnaga ja sisaldavad Jira kaardi nime (või linki sellele). Meetodi *RemoveThreadFromIssue* kood on esitatud lisas 11.

```
const unlinkIssueKeyRegExp = new RegExp(/^unlink +(?!<key>[a-zA-Z]+\-\d{1,4})$/);
const unlinkIssueUrlRegExp = new RegExp(/^unlink +<https:\/\/[redacted]\/(?!<key>[a-zA-Z]+\-\d{1,4})>$/);

controller.hears([unlinkIssueKeyRegExp, unlinkIssueUrlRegExp], ['message', 'direct_message'], async function (bot, message) {
  try {
    var issueKey = message.matches.groups.key;
    var msg = message.incoming_message.channelData;

    if (!msg.thread_ts) {
      return;
    }

    await RemoveThreadFromIssue(bot.api, issueKey, msg);
  } catch (error) {...}
});
```

Joonis 3.8 Slacki lõime ja Jira kaardi vahelise ühenduse kustutamise meetod

### 3.2.5 Yeoman malli installimine Botkiti jaoks

Autor lisas Jira Webhooki taotluste haldamiseks *bot.js* failisse ka lõppsõlme */api/slacklink* (vt Joonis 3.9). Pärast kaardi sulgemist selle lõppsõlme peale tuleb päring koos teabega suletud kaardi kohta. Kui väli sisaldab linke Slack lõimede kohta, saadab bot neile lõimede teate kaardi sulgemise kohta. Koodi leiaste lisast 15.

```
const baseUrl = "https://[redacted]";
controller.websver.post('/api/slacklink', (req, res) => {
  if (req.body && req.body.issue && req.body.issue.fields) {
    var reg = new RegExp(/https:\/\/[redacted]\/[a-zA-Z]+\-\d{1,4}\/thread_ts=(?!<thread_ts>[\d]{10}\.([\d]{6})\&cid=(?!<channel>[\w]{9})/g);
    var slackThreadField = req.body.issue.fields.customfield_11027;
    var issueKey = req.body.issue.key;
    var issueStatus = req.body.issue.fields.status;
    var issueResolution = req.body.issue.fields.resolution;

    if (issueKey && slackThreadField && slackThreadField.length > 0){...}
  }
  res.send('ok');
});
```

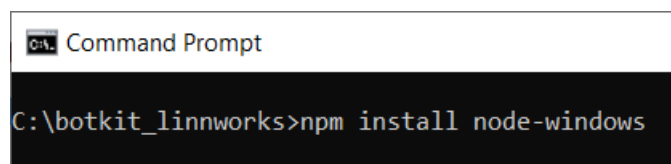
Joonis 3.9 Botirakenduse lõppsõlm Jira veebikonksuks

### 3.3 Botirakenduse juurutamine ettevõtte serverisse

Üks ettevõtte tingimusest oli rakenduse paigutamine ettevõtte siseserverisse. Kuna tegemist on Windows Server 2019 serveriga, käivitab autor rakenduse Windowsi teenusena. See võimaldab rakendusel alustada operatsioonisüsteemi laadimisega [22].

Selleks installib autor serverile Node.js, et rakendust otse käivitada. Autor teostas instal-latsiooni samamoodi nagu selle töö 2.2 osas.

Rakendust käivitava Windowsi teenuse loomiseks kasutas autor *node-windows* npm moodulit [23]. Selleks paigaldas ta selle mooduli konsoolilt spetsiaalse käsu abil (vt Joonis 3.10).



```
Command Prompt
C:\botkit_linnworks>npm install node-windows
```

Joonis 3.10 *Node-windows* installimine

Pärast *node-windows* mooduli installimist lõi autor skripti, mis loob Windowsi teenust (vt Joonis 3.11). Ta paigutas selle skripti rakenduste kataloogis ja märkis suhtelise tee rakenduse käivitusfaili - *bot.js*. *Install* ja *uninstall* funktsioonid ka käivitab ja peatab teenuse.

```
var Service = require('node-windows').Service;
var svc = new Service({
  name: 'JIRA.SlackBot',
  description: 'Runs SlackBot app on node.js server',
  script: 'bot.js'
});

svc.on('install', function(){
  svc.start();
});

svc.on('uninstall', function(){
  svc.stop();
});

svc.install();
```

Joonis 3.11 Skript Windowsi teenuse loomiseks

Pärast Windowsi teenuse installimist ilmub see teenuste loendisse. Teenuse käivitamiseks koos operatsioonisüsteemiga muutis autor käivitustüübi *Automatic* (vt Joonis 3.12). Seega on rakendus juurdepääsetav aadressil <http://localhost:3001>.

Name	Description	Status	Startup Type	Lo
JIRA.SlackBot	Runs SlackBot app on nodejs server	Running	Automatic	Lo
KDC Proxy Server service (KPS)	KDC Proxy Server service runs on edge servers to proxy...		Manual	Ne
KtmRm for Distributed Transaction Coordinator	Coordinates transactions between the Distributed Tran...		Manual (Trigg...	Ne
Link-Layer Topology Discovery Mapper	Creates a Network Map, consisting of PC and device to...		Disabled	Lo
Local Session Manager	Core Windows Service that manages local user session...	Running	Automatic	Lo
Microsoft (R) Diagnostics Hub Standard Collector Service	Diagnostics Hub Standard Collector Service. When run...		Manual	Lo

Joonis 3.12 SlackBot Windows teenus

Rakendusele väljastpoolt pääsemiseks lisas autor IIS-i uue veebilehe ja konfigureeris URL Rewrite mooduli abil ümbersuunamistaotlused sisemisele aadressile *localhost:3001* (vt Joonis 3.13) [24].



### URL Rewrite

Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

Name	Input	Match	Pattern	Action Type	Action URL	Stop Processing	Entry Type
ForwardToNode	URL path after '/'	Matches	*	Rewrite	http://localhost:3001/{R:0}	True	Local

Joonis 3.13 URL Rewrite moodul

Nüüd suunatakse kõik päringud, mis serverile saavad lisatud aadressil, aadressile *localhost:3001*.

## 3.4 Botirakenduse ühendamine platvormiga Slack

Boti rakendus töötab ettevõtte serveris, kuid pole veel Slacki platvormiga ühendatud. Selleks autor konfigureeris sündmuste tellimusi platvormis Slack *Event Subscriptions* jaotises, samamoodi nagu oli kirjeldatud selle töö 2.1 osas.

Boti toimimiseks lisas autor järgmised sündmuste tellimused:

- *message.channels* – kanalites kirjutatud sõnumite töötlemiseks;
- *message.mpim* – rühmadesse kirjutatud sõnumite töötlemiseks.

*Request URL* lahtrisse lisas ta lõigus 2.2 kirjeldatud lõppsõlme sõnumirakenduse (vt Lisa 13). Pärast muudatuste salvestamist töötleb bot kõiki avalikele kanalitele ja gruppidele saadetud sõnumeid. Nii ühendas autor botirakenduse Slack platvormiga.

## 3.5 Jira platvormil teatiste seadistamine

Slack niidi linkide talletamiseks lisas autor Jira konto sätetes lisalahtri *Custom fields* jaotisse (vt Lisa 14). Selleks peavad olema administraatoriõigused, nii et autor logis kontosse sisse administraatorina [25].

Jaotises *Custom fields* klõpsas autor nupul *Add custom field*, mille järel ta valis välja tüübi (vt Lisa 15) ja nime (vt Lisa 16). Kuna väli võib sisaldada mitut linki, *Text Field (multi-line)* tüüp. Nüüd saavad botirakendused salvestada Slacki lõime lingid sellesse lahtrisse.

Pärast uue välja lisamist loob autor veebikonksu, mis saadab päringuid suletud Jira kaardi kohta. Selleks lisas ta Jira konto sätetes *System* jaotises uue veebikonksu *WebHooks* vahekaardile (vt Lisa 17). Lisaks nimele märkis ta netiaadressi, kuhu päringud saadetakse. Sel juhul on selle töö osas 2.2 kirjeldatud botirakenduse */api/slacklink* lõppsõlm.

Nagu on kirjeldatud dokumentatsioonis, saab veebikonksu kasutada *events* kaudu või nagu pärast funktsiooni. Autor kasutab teist võimalust, nii et ta jättis kõik *Events* jaotises olevad märkeruudud tühjaks.

Kui veebikonksu loodi, lisas autor selle pärast funktsioonina üleminekutesse [26]. See võimaldab teil kutsuda välja veebikonksu, kui Jira kaardi olek muutub.

Selleks muutis ta töövoogu *Workflows* jaotises. Töövoogude loendi lehel klõpsas autor *Edit* nupul (*dev*) *Default Workflow* vastas, mida kasutavad arendusmeeskonnad (vt Joonis 3.14).

**Issues** 🔍 Search Jira admin

**Workflows** Add workflow Import from Marketplace ?

**?** • To delete a workflow, you must first unassign it from all workflow schemes and draft workflow schemes.

▼ **Active**

Name	Assigned Schemes	Actions
(dev) Default Workflow <b>i</b>	<ul style="list-style-type: none"><li>(cst) Default Workflow Scheme</li><li>(dev) Default Workflow Scheme</li></ul>	View Edit Copy

Joonis 3.14 Jira töövoogu leht

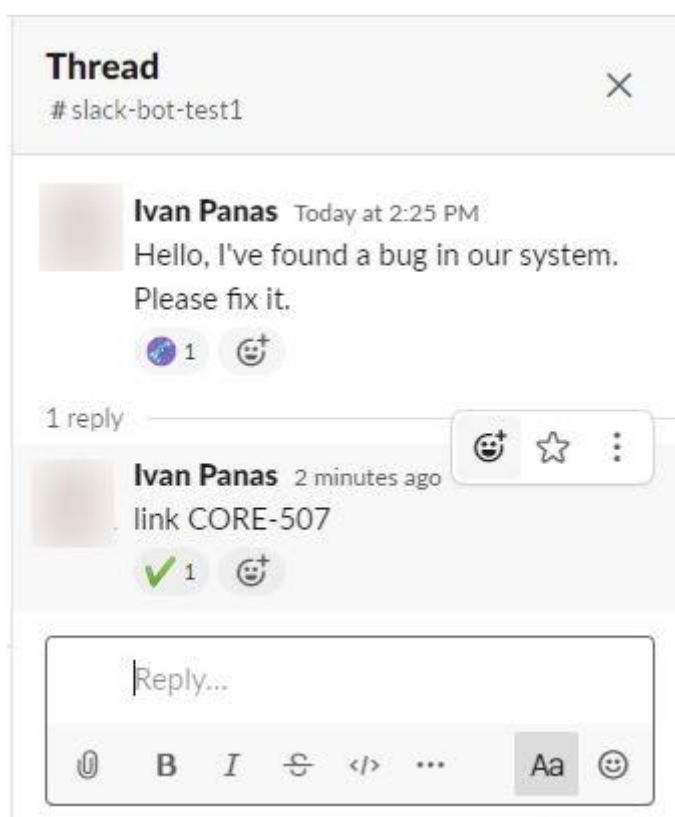
Järgmisena klõpsas ta redigeerimislehel noolele, mis viib *DONE* olekusse, ja ilmunud aknas valis *Post Functions* (vt Lisa 18). Avanenud lehel klõpsas autor nupul *Add post*

*function* (vt Lisa 19) ja valis toiminguga *Trigger a WebHook* (vt Lisa 20). Pärast seda valis pärast funktsiooni parameetrite lehel *JIRA Link* veebikonksu. Samal viisil lisas autor pärast funktsiooni *ARCHIVED* oleku jaoks.

Nii seadis autor veebikonksude saatmise Jira platvormile. Botirakendus saab nüüd teavet suletud Jira kaartide kohta.

### 3.6 Näited boti kasutamisest

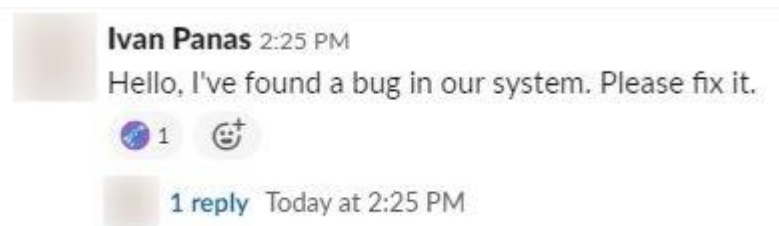
Teavituste tellimiseks saadab kasutaja kanali lõime sees teate, mis algab *link* märksõnaga ja kaardi nime või lingiga (vt Joonis 3.15).



Joonis 3.15 Lisatud *heavy\_check\_mark* pilt

Vastusena sellisele teatele lisab bot pärast edukat töötlemist ikooni, millel on roheline linnuke. See tähendab, et käsk võetakse vastu ja töödeldakse. Kuna Linnworks Slacki konto on Jira integreerimiseks juba rakendusi installinud, vastab teine bot sõnumitele, mis sisaldavad nime Jira kaardid, ja lisab selle sõnumile kirjelduse. Seetõttu seadistas autor boti tekstiga vastussõnumi asemel ikooni lisamiseks, et mitte kanali lõime mitme sõnumiga üle koormata.

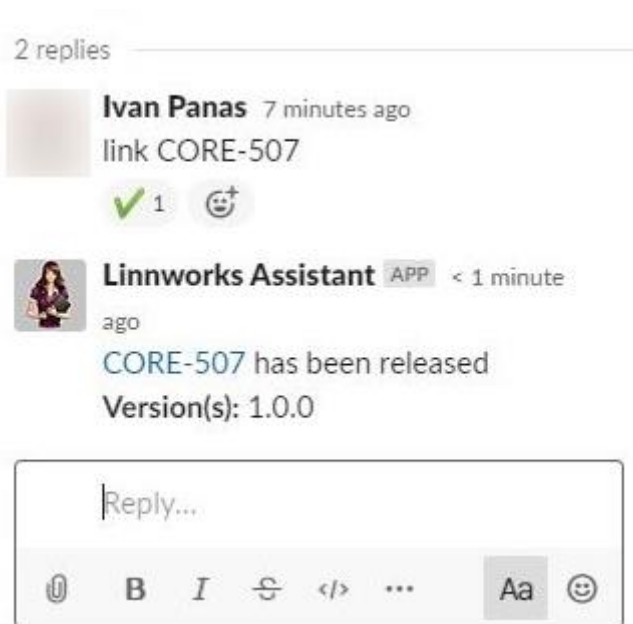
Lisaks kanali lõime sõnumile lisatud ikoonile lisab bot kanali põhisõnumile ka Jira ikooni (vt Joonis 3.16). See tähendab, et kanali lõim on ühendatud Jira kaardiga.



Joonis 3.16 Lisatud *jira* pilt

Teate tellimuse eemaldamiseks saadab kasutaja teate võtmesõnaga *unlink* ja kaardi nimega. Vastuseks sellele lisab bot ka rohelise linnukesega ikooni.

Kui Jira kaart suletakse, saadetakse teade kõikidele *SLACK THREADS* lahtris salvestatud lõimedele. See teade sisaldab põhiteavet suletud kaardi kohta, näiteks selle nime ja väljalaskeversiooni või sulgemise põhjuse (vt Joonis 3.17).



Joonis 3.17 Jira kaardi sulgemise kohta sõnum

### 3.7 Boti edasiarendamine

Autori tehtud töö tulemuseks on ChatOps bot, mis vastab Linnworksile seatud nõuetele. Bot saab suhelda ettevõtte töötajatega ja teatada neile suletud Jira kaardist. Samal ajal on võimalik laiendada botite funktsioonide komplekti, määrates selle muude ülesannete täitmiseks.

Pärast botite käivitamist sai autor mitu päeva kestnud kasutamise ajal kasutajatelt mitmeid ettepanekuid ja märkusi. Seetõttu on järgmine samm, mida autor näeb, vigade parandamine ja boti olemasolevas rakenduses muudatuste tegemine.

Autorit huvitab väga ChatOps boti loomise idee. Samal ajal on Linnworks huvitatud selle integreerimisest oma tööprotsessidesse. Tulevikus rakendatavate ülesannete hulgas tõi autor välja järgmised:

- statistika saamine (näiteks andmed vaba serverimälu kohta);
- kasutajakäskude käivitamine (näiteks töö alustamine kolmanda teenuse osapoolel);
- teatiseid kriitiliste serverite olekute kohta;
- serverite automaatne taaskäivitamine kriitilistes situatsioonides.



## KOKKUVÕTE

Autori töö eesmärk oli luua Slack bot, mis suudab kasutajaid teavitada Jira kaartide sulgemisest. Selle eesmärgi saavutamiseks seadis ta endale mitu ülesannet, mille jooksul see ka saavutati.

Ülesannete hulgas olid:

- 1) Slack API ja JIRA REST API õppimine;
- 2) ülevaade tehnoloogiatest, mis võimaldavad luua Slack botid;
- 3) Slack boti rakendamine uuritud andmete põhjal.

Autori loomingu esimene osa on teoreetiline. Selles viis autor kõigepealt läbi Jira ja Slacki platvormide pakutavate API-de uuringu. Seejärel tegi ta tehnoloogiaülevaate Slack botite loomiseks. Esimese osa lõpus määrati kindlaks tööde teises osas kasutatud tehnoloogiate loetelu. Eelkõige otsustas autor kasutada Slacki boti arendamiseks Botkiti teeki.

Töö teine osa on praktiline ja sisaldab tegelikult esimese osa järelduste põhjal boti väljatöötamise protsessi. Esiteks lõi ta rakenduse Slacki platvormil. Pärast seda kirjeldas botirakenduse loomise protsessi Botkiti teegi põhjal. See Node.js rakendus võeti kasutusele Linnworks Windows serveris. Selleks kasutas autor node-windows npm paketti.

Kui rakendus serveris käivitati, ühendas autor selle Slack rakendusega, seadistades platvormi Slack sündmuste tellimustele. Viimane samm botite väljatöötamisel oli veebikonksu saatmise seadistamine Jira platvormil.

Töö lõpus tõi autor näiteid boti kasutamisest. Samuti hindas ta tulemust ja tegi järeldused Slack boti edasise arengu väljavaadete kohta.

## SUMMARY

The introduction of bots in various chat messengers is gaining popularity nowadays. In this thesis, the author describes one such integration for Slack. The purpose of the author's work was to create a Slack bot that is able to notify users about the closure of Jira cards. He set himself several tasks in the course of which the goal of this work was achieved.

Among the tasks were:

- learning the Slack API and the JIRA REST API;
- an overview of technologies to create Slack bots;
- implementation of Slack bot based on the conclusions of the studied material.

The first part of the author's work is theoretical. In it, the author first conducted research of the APIs provided by the Jira and Slack platforms. After that, he compared the technologies for creating Slack bots. At the end of the first part, a list of technologies that were used to implement the work was determined. In particular, the author decided to use the Botkit library to develop a Slack bot.

The second part of the work is practical and contains the process of developing a bot based on the conclusions from the first part. First, the author created an application in the Slack platform. After that, the process of creating a bot application based on the Botkit library was described. This Node.js application was deployed to a Windows server using the npm node-windows package.

When the application was launched on the server, the author connected it to the Slack application, setting up event subscriptions inside the Slack platform. The last step in developing the bot was setting up sending webhooks in the Jira platform.

At the end of the work, the author gave examples of using the bot. He also assessed the result and made conclusions about the prospects for the further development of the Slack bot.

## KASUTATUD KIRJANDUSE LOETELU

1. Andmekaitse ja infoturbe leksikon. [Online] <https://akit.cyber.ee/> (04.01.2020)
2. IIS Web Server. Vuollet P. (8.05.2018) [Online] <https://stackify.com/iis-web-server/> (04.01.2020)
3. URL Rewrite. [Online] <https://www.iis.net/downloads/microsoft/url-rewrite> (04.01.2020)
4. What is a channel? [Online] <https://slack.com/intl/en-ee/help/articles/360017938993-What-is-a-channel> (02.01.2020)
5. Real Time Messaging API. [Online] <https://api.slack.com/rtm> (04.01.2020)
6. How to use the Slack Web API. [Online] <https://api.slack.com/web> (04.01.2020)
7. What is ChatOps? [Online] <https://docs.stackstorm.com/chatops/chatops.html> (02.01.2020)
8. REST APIs [Online] <https://developer.atlassian.com/server/jira/platform/rest-apis/> (27.12.2019)
9. Widely-used open source libraries. [Online] <https://api.slack.com/community> (27.12.2019)
10. SlackAPI. [Online] <https://github.com/Inumedia/SlackAPI> (07.01.2020)
11. Slack API in Go. [Online] <https://github.com/nlopes/slack> (07.01.2020)
12. Slack-client. [Online] <https://github.com/HubSpot/slack-client> (07.01.2020)
13. Botkit is an open source developer tool for building chat bots, apps and custom integrations for major messaging platforms. [Online] <https://github.com/howdyai/botkit> (07.01.2020)
14. Microsoft to acquire XOXCO [Online] <https://blogs.microsoft.com/blog/2018/11/14/microsoft-to-acquire-xoxco-bringing-together-leading-bot-development-communities-to-help-advance-conversational-ai/> (28/12/2019)
15. Slacker. [Online] <https://github.com/os/slacker> (07.01.2020)
16. Getting started with Yeoman. [Online] <https://yeoman.io/learning/index.html> (27.12.2019)
17. Getting Started. [Online] <https://code.visualstudio.com/docs> (04.01.2020)

18. Enabling interactions with bots. [Online] <https://api.slack.com/bot-users> (29.12.2019)
19. Your Apps. [Online] <https://api.slack.com/apps> (29.12.2019)
20. Node.js. [Online] <https://nodejs.org/en/> (29.12.2019)
21. Let's build a bot! [Online] <https://botkit.ai/getstarted.html> (29.12.2019)
22. Node Windows Service for Running Node.js Application. Tej C. (28.02.2019)  
[Online] <http://bestirtech.com/blog/2019/02/node-windows-service-node-js-app/>  
(30.12.2019)
23. Windows support for Node.JS scripts. [Online]  
<https://github.com/coreybutler/node-windows> (30.12.2019)
24. Creating Rewrite Rules for the URL Rewrite Module. Yakushev R. (28.03.2014)  
[Online] <https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/creating-rewrite-rules-for-the-url-rewrite-module> (30.12.2019)
25. Adding a custom field. [Online] <https://confluence.atlassian.com/adminjiraserver/adding-a-custom-field-938847222.html> (29.12.2019)
26. Adding a webhook as a post function to a workflow. [Online] <https://developer.atlassian.com/server/jira/platform/webhooks/> (29.12.2019)

**LISAD**

# Lisa 1 Slack rakenduse seadete avalet



Linnworks Assis... ▾

**Settings**

- Basic Information**
- Collaborators
- Install App
- Manage Distribution

**Features**

- App Home BETA
- Incoming Webhooks
- Interactive Components
- Slash Commands
- OAuth & Permissions
- Event Subscriptions
- Bot Users
- User ID Translation

**Tools**

- Update to Granular Scopes

Slack ♥

- Help
- Contact
- Policies

## Basic Information

### Building Apps for Slack

Create an app that's just for your workspace (or build one that can be used by any workspace) by following the steps below.

- Add features and functionality ✓ ▶
- Install your app to your workspace ✓ ▶
- Manage distribution ▶

### App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID	Date of App Creation
<input type="text"/>	<input type="text"/>

## Bot User

You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.

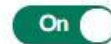
### Display name

Names must be shorter than 80 characters, and can't use punctuation (other than apostrophes and periods).

### Default username

If this username isn't available on any workspace that tries to install it, we will slightly change it to make it work. Usernames must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores.

### Always Show My Bot as Online



When this is off, Slack automatically displays whether your bot is online based on usage of the RTM API.

Changes will affect **future installations** of your app. If you've already installed your app to a workspace and want to rename the bot user, navigate to your workspace and choose **Administration > Manage apps**.

## Lisa 3 Kuvatud Slack rakendusteabe seadistamine

### Display Information

This information will be shown in the Slack App Directory and in the Slack App. For more information, view our [App Detail Guidelines](#).

App name

Linnworks Assistant

Short description

Your Personal Assistant

App icon & Preview


















Background color

#36446e



## Lisa 4 Vajalikud load Slack rakenduseks

OAuth scopes let you specify exactly how your app needs to access a Slack user's account.

OAuth Scope	Description	
<b>bot</b>	Add the ability for people to direct message or mention @jiralink	
<b>channels:history</b>	View messages and other content in the user's public channels	
<b>channels:read</b>	View basic information about public channels in the workspace	
<b>chat:write:bot</b>	Send messages as Linnworks Assistant	
<b>commands</b>	Add actions and/or slash commands that people can use	
<b>identify</b>	View information about the user's identity	
<b>im:history</b>	View messages and other content in the user's direct messages	
<b>im:read</b>	View basic information about the user's direct messages	
<b>incoming-webhook</b>	Post messages to specific channels in Slack	
<b>links:read</b>	View some URLs in messages	
<b>links:write</b>	Show previews of URLs in messages	
<b>mpim:history</b>	View messages and other content in the user's group direct messages	
<b>mpim:read</b>	View basic information about the user's group direct messages	
<b>reactions:read</b>	View emoji reactions in the user's channels and conversations and their associated content	
<b>users:read</b>	View people in the workspace	

[Add an OAuth Scope](#)

Scopes define the API methods an app is allowed to call, which information and capabilities are available on the workspace it's installed on. Many scopes are restricted to specific resources like channels or files.

## OAuth & Permissions




### OAuth Tokens & Redirect URLs

These [OAuth Tokens](#) will be automatically generated when you finish connecting the app to your workspace. You'll use these tokens to authenticate your app.

[Install App to Workspace](#)

## Lisa 6 Slack rakenduse installimise kinnituse leht

### What will Linnworks Assistant be able to view?

-  Content and info about you ▶
-  Content and info about channels & conversations ▶
-  Content and info about your workspace ▶

### What will Linnworks Assistant be able to do?

-  Perform actions in channels & conversations ▶
-  Perform actions in your workspace ▶

### Where should Linnworks Assistant post?

# Linnworks Assistant requires a channel to post to as an app

Search for a channel... ▼

Cancel

Allow

## OAuth & Permissions

### OAuth Tokens & Redirect URLs

#### Tokens for Your Workspace

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

##### OAuth Access Token

[Redacted token] Copy

##### Bot User OAuth Access Token

[Redacted token] Copy

[Reinstall App](#)

## Event Subscriptions

### Enable Events



Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)

#### Request URL

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. [Learn more.](#)

### Subscribe to bot events

### Subscribe to workspace events

To subscribe to an event, your app must have access to the related [OAuth permission scope](#).

Event Name	Description	Required Scope
------------	-------------	----------------

No events added yet.

### App unfurl domains

## Lisa 9 Meetodid findIssues ja updateIssues

```
var api = GetJiraApi();
await api.findIssue(issueKey)
  .then(function (issue) {
    var slackThreadsField = issue.fields.customfield_11027;
    var threadResponse = GetLinkForThread(msg.channel, msg.thread_ts, controller.adapter.slack.token);
    threadResponse.then(function (thread) {
      if (slackThreadsField == null || !slackThreadsField.includes(thread.permalink)) {
        if (slackThreadsField == null) {
          slackThreadsField = thread.permalink;
        }
        else {
          slackThreadsField += `${thread.permalink}\n`;
        }

        var fieldToSave = { fields: { customfield_11027: slackThreadsField } };
        api.updateIssue(issueKey, fieldToSave)
          .then(function (result) {
            AddReaction(botApi, msg.ts, msg.channel, 'heavy_check_mark');
            AddReaction(botApi, msg.thread_ts, msg.channel, 'jira');
          })
          .catch(function (error) { .. });
      }
      else {
        AddReaction(botApi, msg.ts, msg.channel, 'heavy_check_mark');
      }
    })
  }).catch(function (error) { .. });
})
.catch(function (err) { .. });
```

## Lisa 10 Meetod AddThreadToIssue

```
async function AddThreadToIssue(botApi, issueKey, msg) {
  var api = GetJiraApi();
  await api.findIssue(issueKey)
    .then(function (issue) {
      var slackThreadsField = issue.fields.customfield_11027;
      var threadResponse = GetLinkForThread(msg.channel, msg.thread_ts,
controller.adapter.slack.token);
      threadResponse.then(function (thread) {
        if (slackThreadsField == null ||
!slackThreadsField.includes(thread.permalink)) {
          if (slackThreadsField == null) {
            slackThreadsField = thread.permalink;
          }
          else {
            slackThreadsField += `${thread.permalink}\n`;
          }
        }

        var fieldToSave = { fields: { customfield_11027:
slackThreadsField } };
        api.updateIssue(issueKey, fieldToSave)
          .then(function (result) {
            AddReaction(botApi, msg.ts, msg.channel,
'heavy_check_mark');
            AddReaction(botApi, msg.thread_ts, msg.channel, 'jira');
          })
          .catch(function (error) {
            console.log(`Failed to update issue field:
${issueKey}`);
          });
      });
    }
    else {
      AddReaction(botApi, msg.ts, msg.channel, 'heavy_check_mark');
    }
  }).catch(function (error) {
    console.log(`Failed to get thread permalink: ${issueKey}`);
  });
})
.catch(function (err) {
  console.log(`The issue is not found: ${issueKey}`);
});
}
```

## Lisa 11 Meetod RemoveThreadFromIssue

```
async function RemoveThreadFromIssue(botApi, issueKey, msg) {
  var api = GetJiraApi();
  api.findIssue(issueKey)
    .then(function (issue) {
      var slackThreadsField = issue.fields.customfield_11027;
      if (slackThreadsField !== undefined) {
        if (slackThreadsField == null) {
          slackThreadsField = '';
        }

        var threadResponse = GetLinkForThread(msg.channel, msg.thread_ts,
controller.adapter.slack.token);
        threadResponse.then(function (thread) {
          var exist = slackThreadsField.includes(thread.permalink);
          var existWithNewLine =
slackThreadsField.includes(`${thread.permalink}\n`);


          if (exist) {
            if (existWithNewLine) {
              slackThreadsField =
slackThreadsField.replace(`${thread.permalink}\n`, '');
            }
            else {
              slackThreadsField =
slackThreadsField.replace(thread.permalink, '');
            }

            var fieldToSave = { fields: { customfield_11027:
slackThreadsField } };
            api.updateIssue(issueKey, fieldToSave)
              .then(function (result) {
                AddReaction(botApi, msg.ts, msg.channel,
'heavy_check_mark');
              })
              .catch(function (error) {
                console.log(`Failed to update issue field:
${issueKey}`);
              });
            }
            else {
              AddReaction(botApi, msg.ts, msg.channel,
'heavy_check_mark');
            }
          }).catch(function (err) {
            console.log(`Failed to get thread permalink: ${issueKey}`);
          });
        }
      })
    })
    .catch(function (err) {
      console.log(`The issue is not found: ${issueKey}`);
    });
}
}
```







 Linnworks Assis... ▾

**Settings**

- Basic Information
- Collaborators
- Install App
- Manage Distribution

**Features**

- App Home BETA
- Incoming Webhooks
- Interactive Components
- Slash Commands
- OAuth & Permissions
- Event Subscriptions**
- Bot Users
- User ID Translation

**Tools**

- Update to Granular Scopes

Slack ❤️

- Help
- Contact
- Policies
- Our Blog

## Event Subscriptions

**Enable Events**  On

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)

**Request URL Verified** ✓

https://[REDACTED]/api/messages
Change

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. [Learn more.](#)



---

**Subscribe to bot events** ▶

---

**Subscribe to workspace events** ▾

To subscribe to an event, your app must have access to the related OAuth permission scope.

Event Name	Description	Required Scope	
message.channels	A message was posted to a channel	channels:history	
message.mpim	A message was posted in a multiparty direct message channel	mpim:history	

Add Workspace Event

## Lisa 14 Jira kohandatud väljad

Jira Software

Jira settings

Issues

ISSUE TYPES

Issue types

Issue type schemes

Sub-tasks

WORKFLOWS

Workflows

Workflow schemes

SCREENS

Screens

Screen schemes

Issue type screen schemes

FIELDS

Custom fields

Field configurations

### Issues

Search Jira admin

### Custom fields

Add custom field ?

Name	Type	Available Context(s)	Screens
<b>Approvers</b> Contains users needed for approval. This custom field was created by Jira Service Desk.	User Picker (multiple users)	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Change completion date</b> Specify the completion time for the change request	Date Time Picker	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Change reason</b> Choose the reason for the change request	Select List (single choice)	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Change risk</b>	Select List (single choice)	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Change start date</b> Specify the time and date for the start of the change	Date Time Picker	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Change type</b>	Select List (single choice)	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...
<b>Epic Color</b> <span>LOCKED</span> Epic Color field for Jira Software use only.	Color of Epic	Issue type(s): <ul style="list-style-type: none"><li>Global (all issues)</li></ul>	...

## Lisa 15 Jira kohandatud välja tüübi valimine

### Select a Field Type



All

**Standard**

Advanced

#### Select List (single choice)

A single select list with a configurable list of options.

#### Text Field (multi-line)

A multiline text area custom field to allow input of longer text strings.

#### Text Field (single line)

A basic single line text box custom field to allow simple text input.

#### URL Field

Allow the user to input a single URL.

#### User Picker (single user)

Choose a user from the user base via a popup picker window.

**Next**

Cancel

## Lisa 16 Jira kohandatud välja nimi

### Configure 'Text Field (multi-line)' Field

Name\*

Description

[Previous](#) [Create](#) [Cancel](#)

# Lisa 17 Jira veebikonksu lisamine

The screenshot shows the Jira Software administration interface. On the left is a blue sidebar with navigation options: Restore system, MAIL (Global Mail Settings, Outgoing Mail, Incoming Mail, Send email), ADMIN HELPER (Permission helper, Notification helper), SHARED ITEMS (Shared filters, Shared dashboards), and ADVANCED (Attachments, Events, WebHooks, Services, LDAP/Okta management). The 'WebHooks' option is selected.

The main content area is titled 'System WebHooks'. At the top right, there is a search bar 'Search Jira admin' and a '+ Create a Webhook' button. The configuration for a 'JIRA Link' is shown with the following fields:

- Name \***: JIRA Link
- Status \***: Enabled (selected), Disabled
- URL \***: https://[redacted]/api/slacklink

Below the URL field, there is a list of variables that can be used in the URL: `{attachment.id}`, `{board.id}`, `{comment.id}`, `{issue.id}`, `{issue.key}`, `{mergedVersion.id}`, `{modifiedUser.accountId}`, `{modifiedUser.key}`, `{modifiedUser.name}`, `{project.id}`, `{project.key}`, `{property.key}`, `{type.id}`, `{version.id}`, and `{worklog.id}`. A 'Read more' link is provided.

A **Description** text area is also present.

**Events**

**Issue related events**

You can specify a JQL query to send only events triggered by matching issues. The JQL filter only applies to events under the Issue and Comment columns.

All issues

**Syntax help**

Issue	Issue link	Comment	Attachment	Worklog	Entity property
<input type="checkbox"/> created	<input type="checkbox"/> created	<input type="checkbox"/> created	<input type="checkbox"/> created	<input type="checkbox"/> created	<input type="checkbox"/> created or updated
<input type="checkbox"/> updated	<input type="checkbox"/> updated	<input type="checkbox"/> updated	<input type="checkbox"/> updated	<input type="checkbox"/> updated	<input type="checkbox"/> updated
<input type="checkbox"/> deleted	<input type="checkbox"/> deleted	<input type="checkbox"/> deleted	<input type="checkbox"/> deleted	<input type="checkbox"/> deleted	<input type="checkbox"/> deleted

**Project related events**

# Lisa 18 Jira töövoore redigeerimise leht

## Issues

Search Jira admin

 You are editing a draft workflow. [Publish Draft](#) [Discard Draft](#) [View original](#)

Workflows

(dev) Default Workflow  **DRAFT** SHARED BY 16 PROJECTS 

**Diagram** Text Export ▾

[+ Add status](#) [+ Add transition](#)  Show transition labels Last edited by Evgeny Agureev, 

+  
-



### Archive

**Screen** (dev) Archive Issue Screen

[Edit](#) [Delete transition](#)

#### Options

- Properties (0)
- Triggers (0)
- Conditions (0)
- Validators (1)
- Post Functions (9)



## Lisa 19 Jira pärastise funktsiooni lisamine

Transition is available to **ALL** steps in the workflow.

**Screen:** (dev) Archive Issue Screen

Triggers 0 Conditions 0 Validators 1 Post Functions 9

### The following will be processed after the transition occurs

[Add post function](#)

1. The value of field **Assignee** will be cleared.
2. The value of field **Sprint** will be cleared.
3. Set issue status to the linked status of the destination workflow step.
4. Add a comment to an issue if one is entered during a transition.
5. Update change history for an issue and store the issue in the database.
6. Re-index an issue to keep indexes in sync with the database.
7. Fire a **Issue Archived** event that can be processed by the listeners.
8. Send transition to webhook: JIRA Integration
9. Send transition to webhook: JIRA Link

### Administration

🔍 Search Jira admin

#### Add Post Function To Transition

Name	Description
<input type="radio"/> Assign to Current User	Assigns the issue to the current user if the current user has the 'Assignable User' permission.
<input type="radio"/> Assign to Lead Developer	Assigns the issue to the project/component lead developer
<input type="radio"/> Assign to Reporter	Assigns the issue to the reporter
<input type="radio"/> Clear Field Value	Clear value of a given field.
<input type="radio"/> Copy Value From Other Field	Copies the value of one field to another, either within the same issue or from parent to sub-task.
<input type="radio"/> Create Crucible Review Workflow Function	Creates a Crucible review for all unreviewed code for this issue
<input type="radio"/> Set issue security level based on user's project role	Set the issue's Security Level to the specified level if the current user is in a specified Project Role.
<input checked="" type="radio"/> Trigger a Webhook	If this post-function is executed, Jira will post the issue content in JSON format to the URL specified.
<input type="radio"/> Update Issue Custom Field	Updates an issue custom field to a given value.
<input type="radio"/> Update Issue Field	Updates a simple issue field to a given value.

Add Cancel