****

**Kauno technologijos universitetas**

Informatikos fakultetas

Architektūros specifikacija

|  |
| --- |
| **Tadas Laurinaitis**  studentas |
| **Doc. Šarūnas Packevičius**  Dėstytojas |

**Kaunas, 2024**

Contents

[1. Komponento aprašymas 3](#_Toc167662631)

[1.1. Teikiamos funkcijos 3](#_Toc167662632)

[1.2. Paskirtis 3](#_Toc167662633)

[1.3. Kokias problemas sprendžia 3](#_Toc167662634)

[1.4. Kokiu tikslu galima naudoti 3](#_Toc167662635)

[1.4. Kokie apribojimai taikomi komponento naudojimui 3](#_Toc167662636)

[1.5. Kita svarbi informacija komponento vartojimui 3](#_Toc167662637)

[2. Komponento architektūra 4](#_Toc167662638)

[2.1. Klasių diagrama 4](#_Toc167662639)

[2.2. Sekų diagrama 4](#_Toc167662640)

[2.3. Būsenų diagrama 5](#_Toc167662641)

[2.4. Išdėstymo diagrama 5](#_Toc167662642)

[2.5. Kita papildoma informacija ir modeliai, kuriais remiantis galima būtų suprasti kaip komponentas funkcionuoja 5](#_Toc167662643)

[3. Komponento aprašymas 6](#_Toc167662644)

[3.1. Klasių diagramos 6](#_Toc167662645)

[3.2. Detalūs metodų, klasių, paketų aprašai 6](#_Toc167662646)

[3.3. Dokumentacija 7](#_Toc167662647)

[3.4. Naudojimo pavyzdys 9](#_Toc167662648)

[3.5. Komponento vartojimo vadovas 9](#_Toc167662649)

[3.6. Komponento diegimo instrukcijos ir papildomi reikalavimai naudojimui 10](#_Toc167662650)

[3.7. Reikalavimai diegimo aplinkai 10](#_Toc167662651)

[3.8. Pakeitimų istorija 11](#_Toc167662652)

[3.9. Licencija 11](#_Toc167662653)

[4. Rinkoje esamų komponentų apžvalga 12](#_Toc167662654)

# 1. Komponento aprašymas

## 1.1. Teikiamos funkcijos

Komponentas suteikia galimybę naudotis bendros paskirties HTTP kliento sąsaja (angl. Generic HTTP Client), kuri palaiko GET, POST, PUT, DELETE užklausų tipus, turi automatinę JSON atsakymų deserializaciją bei išsamų klaidų tvarkymą, su aiškiais atsakymų statusais ir klaidų pranešimais.

## 1.2. Paskirtis

Šio komponento paskirtis yra suteikti naudotojui galimybę kurti naujas API klientines aplikacijas greičiau ir lengviau, negalvojant apie daug laiko užimančius procesus, tokius kaip - objektų deserializacija, klaidų tvarkymas ir klaidų pranešimų apdorojimas.

## 1.3. Kokias problemas sprendžia

Kuriant API klientines aplikacijas dažnu atveju būna labai daug pasikartojančio kodo, ypač susijusio su klaidų valdymu ir JSON deserializacija, kuris galiausiai apkrauna visą kuriamą sistemos architektūrą ir apsunkina navigaciją tos pačios sistemos kode. Šis komponentas leidžia naudotojui išvengti šios problemos, padengdamas ją sukeliančius veiksnius bendros paskirties kodu, kurį lengva naudoti - paduodant url kartu su norimais objektais į komponentą.

## 1.4. Kokiu tikslu galima naudoti

Šį komponentą galima naudoti aplikacijose, dirbančiuose su HTTP protokolu grįstais API, klientinės logikos rašymo palengvinimui, kadangi padengiama tiek objektų deserializacijos, tiek klaidų valdymo ir pranešimų grąžinimo dalys.

## 1.4. Kokie apribojimai taikomi komponento naudojimui

Komponentas neveikia su senesnėmis nei .NET 8 versijomis. Komponentas skirtas darbui tik su HTTP protokolą naudojančiais API.

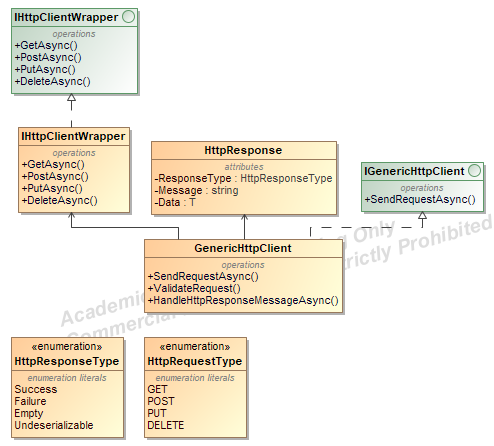
## 1.5. Kita svarbi informacija komponento vartojimui

Komponentas yra pritaikytas naudoti .NET projektuose, turinčiuose bent .NET 8 versiją.

# 2. Komponento architektūra

## 2.1. Klasių diagrama

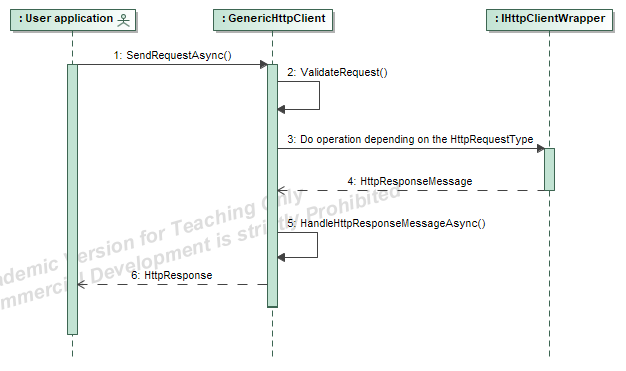
Komponento klasių diagrama pateikiama žemiau esančiame paveikslėlyje (**1 pav.).**



**1 pav.** Komponento klasių diagrama.

## 2.2. Sekų diagrama

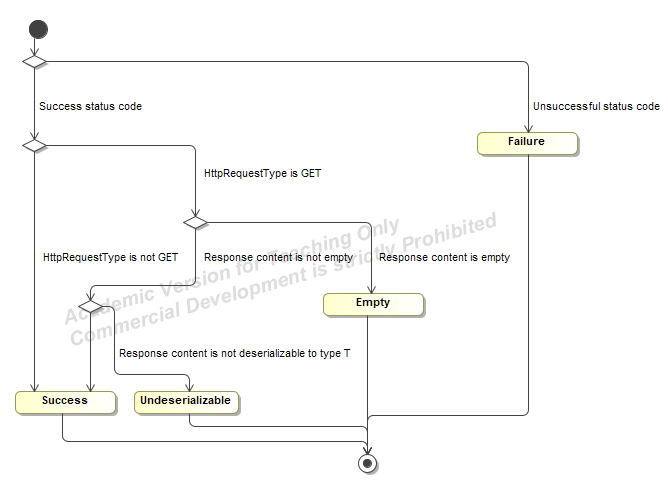
Komponento sekų diagrama pateikiama žemiau esančiame paveikslėlyje (**2 pav.).**



**2 pav.** Komponento sekų diagrama.

## 2.3. Būsenų diagrama

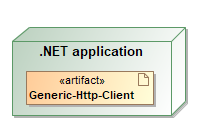
Komponento klasės HttpResponse būsenų diagrama pateikiama žemiau esančiame paveikslėlyje (**3 pav.**).



**3 pav.** Komponento klasės HttpResponse būsenų diagrama.

## 2.4. Išdėstymo diagrama

Komponento išdėstymo diagrama pateikiama žemiau esančiame paveikslėlyje (**4 pav.**)



**4 pav.** Komponento išdėstymo diagrama.

## 2.5. Kita papildoma informacija ir modeliai, kuriais remiantis galima būtų suprasti kaip komponentas funkcionuoja

Prieš inicializuojant komponento GenericHttpClient klasę, reikia inicializuoti IHttpClientWrapper interfeisą išpildančios klasės objektą ir jį paduoti į GenericHttpClient klasės konstruktorių. IHttpClientWrapper interfeisas gali būti įgyvendinamas pagal naudotojo pageidavimus, tačiau tinkant paprastam komponento funkcionalumui, naudotojas gali naudoti jau sukurtą HttpClientWrapper klasę, kuri išpildo IHttpClientWrapper interfeisą. Visas GenericHttpClient funkcionalumas pasiekiamas per vienintelį viešą metodą SendRequestAsync.

# 3. Komponento aprašymas

## 3.1. Klasių diagramos

Klasių diagrama matoma poskyryje „2.1. Klasių diagrama“ **1 pav.**

## 3.2. Detalūs metodų, klasių, paketų aprašai

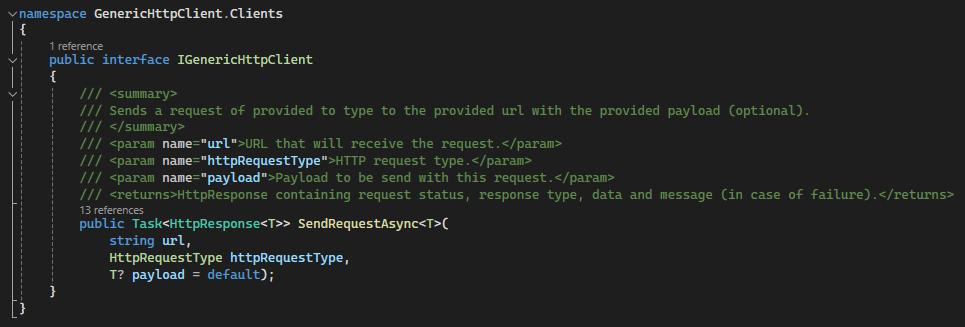
Komponento klasių aprašai pateikiami žemiau esančioje lentelėje (**1 lentelė**).

**1 lentelė.** Komponento klasių aprašai

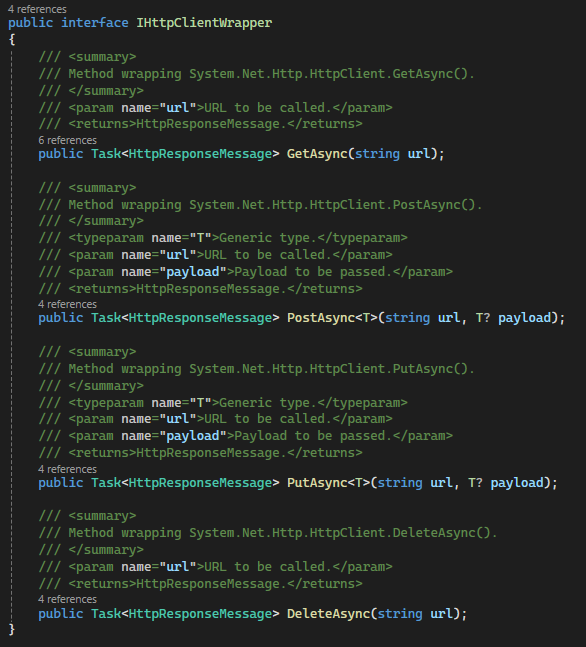
|  |  |  |  |
| --- | --- | --- | --- |
| Klasės pavadinimas | Klasės aprašymas | Metodo pavadinimas | Metodo aprašymas |
| HttpClientWrapper | Klasė, apgaubianti System.Net.Http.HttpClient klasę. | GetAsync | Atlieka GET HTTP operaciją. |
| PostAsync | Atlieka POST HTTP operaciją. |
| PutAsync | Atlieka PUT HTTP operaciją. |
| DeleteAsync | Atlieka DELETE HTTP operaciją. |
| GenericHttpClient | Bendros paskirties HTTP kliento klasė, atsakinga už HTTP operacijų validavimą ir kvietimą, bei operacijos rezultatų apdorojimą – JSON objektų deserializaciją, klaidos statusų ir pranešimų apdorojimą ir grąžinimą. | SendRequestAsync | Atlieka GET/POST/PUT/DELETE HTTP operacijas priklausomai nuo paduotų parametrų. Priima URL, kurį norima kviesti, HttpRequestType enumeratoriaus reikšmę, ir objektą, kuris gali būti perduodamas PUT/POST operacijose, kaip parametrus. Metodas yra bendrinio tipo (generic), todėl yra skirtas atlikti pagrindines HTTP operacijas su bet kokiu objekto tipu. |
| HttpResponse | Klasė, grąžinama naudotojui atgal SendRequestAsync metu. |  |  |
| HttpRequestType | HTTP operacijos, kurią norima atlikti, tipas. Galimos reikšmės – GET, PUT, POST ir DELETE. |  |  |
| HttpResponseType | Komponento apdoroto atsako tipas, kuris priklauso nuo SendRequestAsync metu įvykusių veiksmų. Galimos reikšmės – Success, Failure, Empty, Undeserializable. |  |  |

## 3.3. Dokumentacija

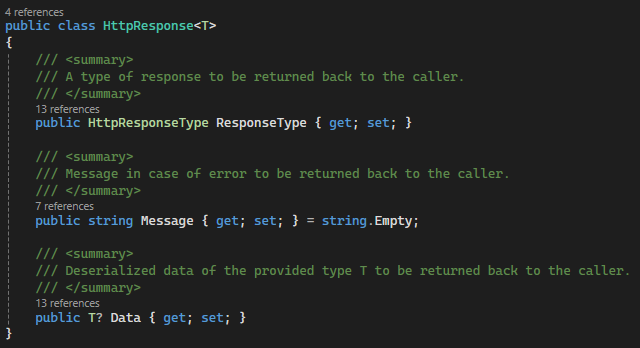
Pagrindinių komponento klasių dokumentacija pateikiama žemiau esančiuose paveikslėliuose (**5 – 9 pav.**)



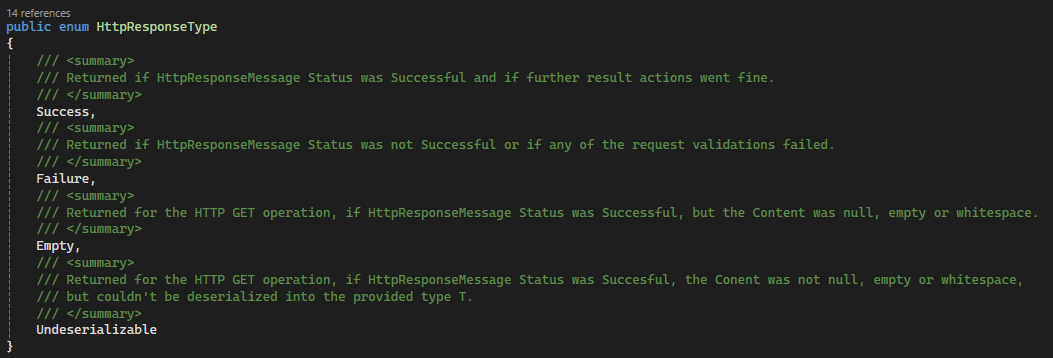
**5 pav.** Klasės „GenericHttpClient“ dokumentacija.



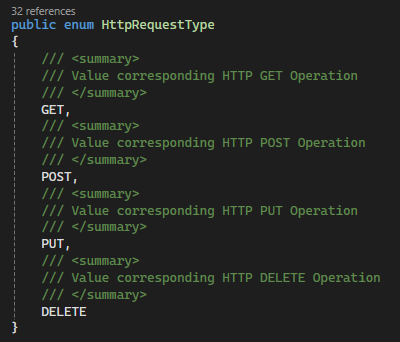
**6 pav.** Klasės „HttpClientWrapper“ dokumentacija.



**7 pav.** Klasės „HttpResponse“ dokumentacija.



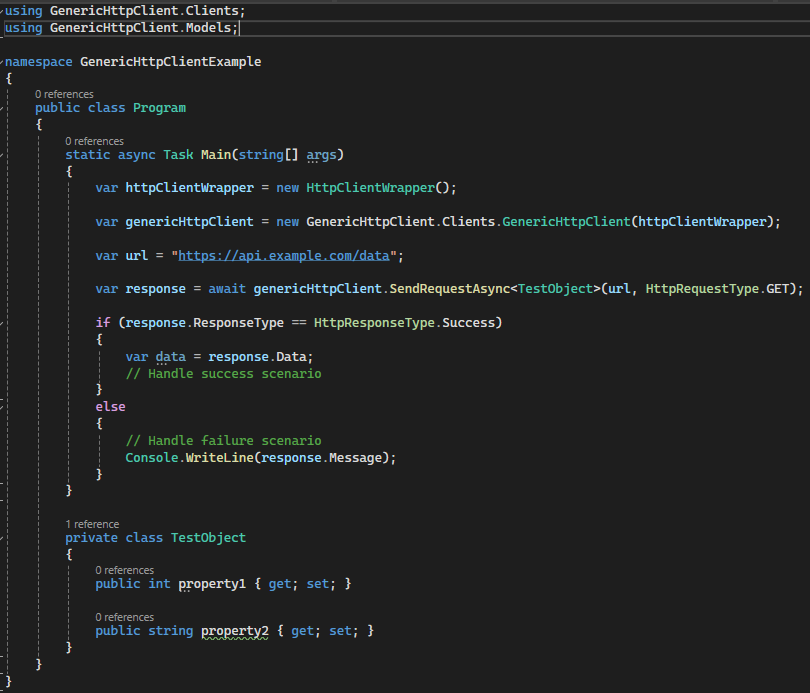
**8 pav.** Enumeratoriaus „HttpResponseType“ dokumentacija.



**9 pav.** Enumeratoriaus „HttpRequestType“ dokumentacija.

## 3.4. Naudojimo pavyzdys

Komponento naudojimo pavyzdys pateikiamas žemiau esančiame paveikslėlyje (**10 pav.**)



**10 pav.** Komponento naudojimo pavyzdys.

## 3.5. Komponento vartojimo vadovas

1. Sukuriamas klasės HttpClientWrapper tipo objektas (**11 pav.**)



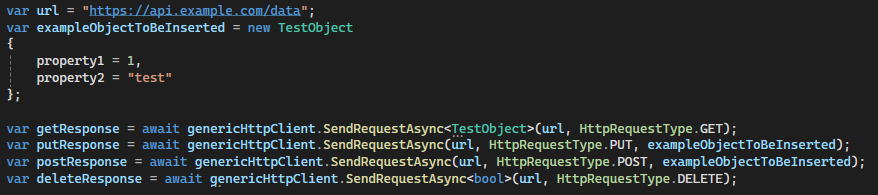
**11 pav.** HttpClientWrapper objekto sukūrimas.

2. Sukuriamas klasės GenericHttpClient tipo objektas (**12 pav.**)



**12 pav.** GenericHttpClient objekto sukūrimas.

3. Siunčiamos užklausos su pasirinktomis HTTP operacijomis (**13 pav.**)



**13 pav.** GET, PUT, POST ir DELETE HTTP užklausų siuntimas naudojantis GenericHttpClient komponentu.

## 3.6. Komponento diegimo instrukcijos ir papildomi reikalavimai naudojimui

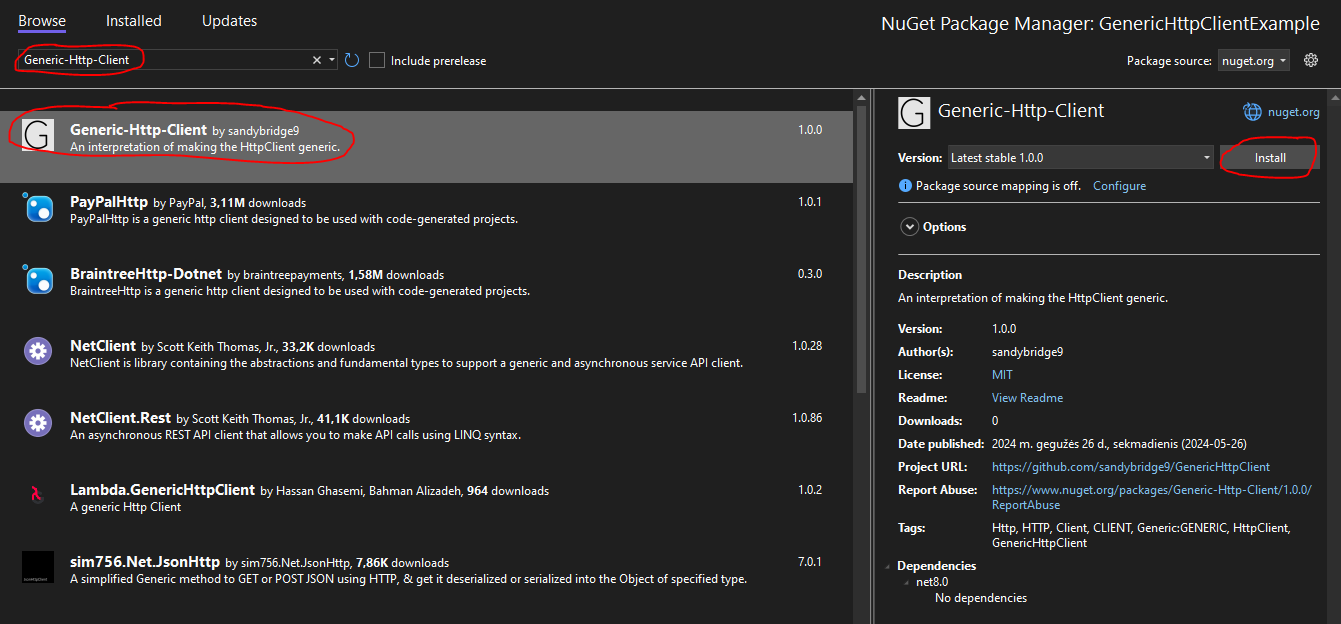
Prieš pradedant naudoti komponentą, jį reikia įsidiegti. Išskiriami trys įdiegimo būdai: naudojantis .NET CLI (**14 pav.**), NuGet Package Manager (**15 pav.**) arba naudojantis Visual Studio NuGet Package Manager grafine sąsaja (**16 pav.**).



**14 pav.** .NET CLI komponento diegimo būdas



**15 pav.** NuGet Package Manager diegimo būdas



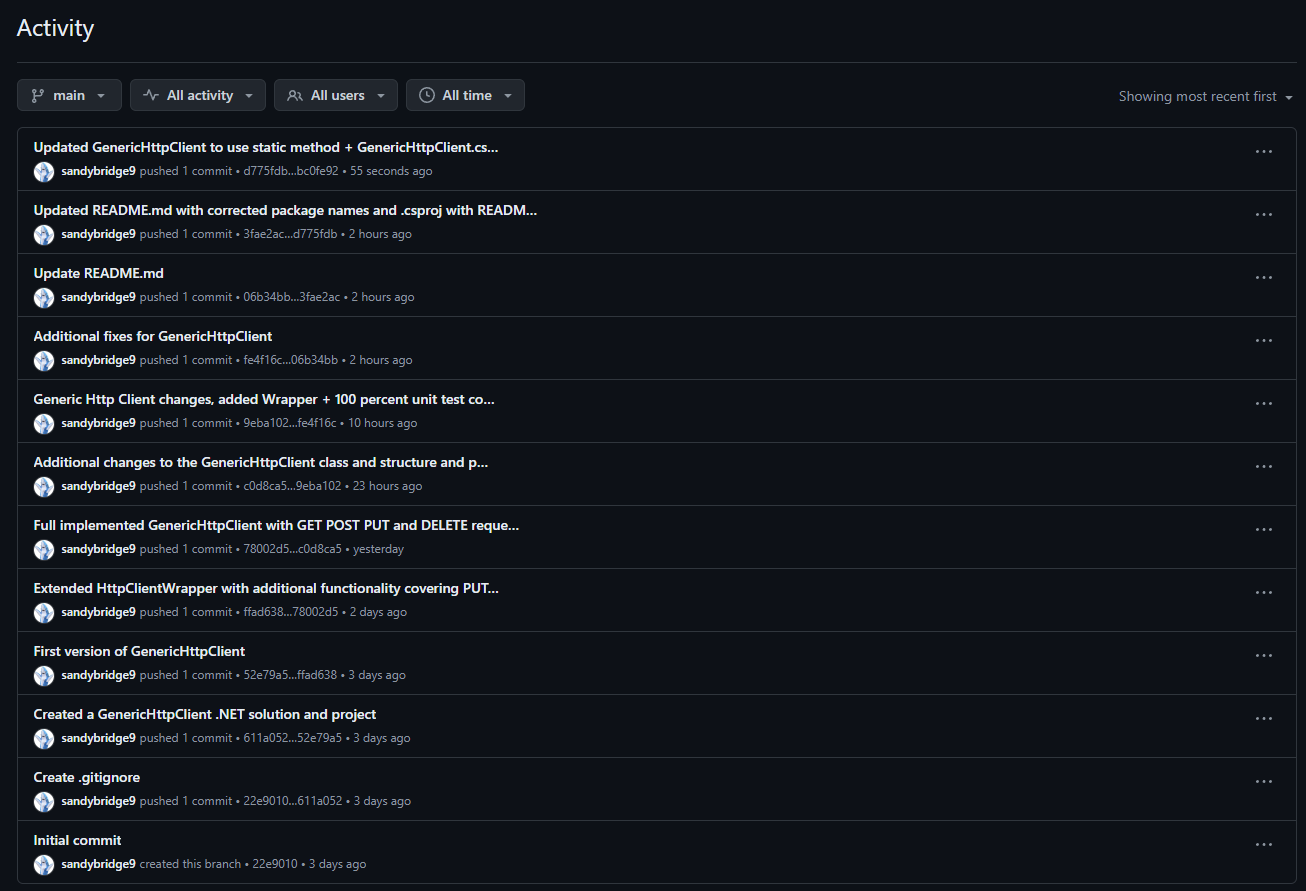
**16 pav.** Visual Studio NuGet Package Manager grafinės sąsajos diegimo būdas.

## 3.7. Reikalavimai diegimo aplinkai

Šis komponentas pritaikytas veikti .NET projektuose, turinčiuose bent .NET 8 versiją.

## 3.8. Pakeitimų istorija

Komponento Github pakeitimų istorija pateikiama žemiau pateiktame paveikslėlyje (**17 pav.**)



**17 pav.** Komponento Github pakeitimų istorija

## 3.9. Licencija

GenericHttpClient komponentui taikoma MIT licencija.

# 4. Rinkoje esamų komponentų apžvalga

.NET HttpClient klasę apgaubiančių ir jos funkcionalumą patobulinančių komponentų rinkoje yra daug, tačiau kiekvienas komponentas skirtingai praplečia HttpClient funkcionalumą, todėl skirtingose situacijose logiškiausia naudoti skirtingus komponentus. Peržvelgus vienus populiariausių HttpClient praplečiančių komponentų, randame pavyzdžių - pavyzdžiui Flurl.Http (<https://www.nuget.org/packages/Flurl.Http>) turi labai gerą URL konstravimo mechanizmą ir lengvą praplėčiamumą, tačiau neturi įskiepytos JSON objektų deserializacijos ir paprasto klaidų valdymo mechanizmo. Tuo metu Polly.Extensions.Http (<https://www.nuget.org/packages/Polly.Extensions.Http>) turi labai gerą klaidų valdymo mechanizma, tačiau neturi jokio URL konstravimo mechanizmo ir jokio bendrinio JSON objektų deserializavimo mechanizmo.

Didžioji dauguma .NET komponentų taip pat yra talpinami per nuget.org, kadangi tai yra pagrindinis komponentų talpinimo ir valdymo įrankis .NET platformai. Norint sukurti naują NuGet komponentą, reikia sekti šių instrukcijų:

1. Sukurti klasės bilbiotekos tipo projektą.
2. Sukonfiguruoti paketo parametrus projekto savybių lange:
   1. Pridėti paketo pavadinimą
   2. Pridėti viešai matomą paketo pavadinimą
   3. Pridėti paketo versijavimo taisykles
   4. Pridėti nuorodas
   5. Pridėti paketo identifikatorių
   6. Pridėti README.md failą
   7. Pridėti licenciją
3. Sugeneruoti projekto .nupkg failą
4. Susikurti nuget.org paskyrą ir įkelti šį failą į nuget.org
5. Palaukti patvirtinimo kad komponentas sėkmingai patalpintas nuget.org talpykloje.

Šio projekto metu kurtas komponentas yra viešai pasiekiamas per nuorodą: <https://www.nuget.org/packages/Generic-Http-Client/> . Komponento kodas taip pat yra viešai prieinamas adresu: <https://github.com/sandybridge9/GenericHttpClient>. Dabartinė komponento NuGet versija – 1.0.0