

Բարի գալուստ NerdNote հավելված

Nerdnote իրենից ներկայացնում է **CLI** հավելված նախատեսված արագ նշումների համար միագնամից տերմինալում: Այն տրամադրում է բազմապատկերական ֆունկցիոնալ, որը ներառում է *նշումների գրանցում, օրվա ընթացքում արված նշումների հավաքագրումը մեկ ֆայլում, դիտում, վերափոխում այլ ֆորմատների (pdf, html), ջնջում:* Այն աշխատում է մարկդաուն (markdown) ֆորմատի հետ:

ԿԱՐԵՎՈՐ Է

Ծրագիրը համագործակցում է *Python 3.10* կամ ավելի բարձր վերսիաների հետ: Հավելվածը արդիական է միայն **macos, linux, openbsd և այլ unix-like** Օպերացիոն Համակարգով աշխատող մեքենաների համար:

NerdNote | ՏԵՂԱԴՐՄԱՆ ԳՈՐԾԸՆԹԱՑԸ:

Ռեպոզիտորիայի պատճենում:

Հավելվածի հավաքագրման համար անհրաժեշտ բոլոր ֆայլերը ընդգրկող ռեպոզիտորիան կարելի է պատճենել հետևյալ հրամանով.

```
git clone https://github.com/stpatriarch/nerdonte.git
```

Տվյալների բազա:

Որպես տվյալների պահպանման բազա օգտագործված է **SQLite3** այն Python-ի հիմնական գրադանների մի մասն է, հետևաբար չունի հավելյալ տեղադրման կամ կարգավորման կարիք: *Ընտրվել է այս բազան որպեսզի հեշտ և արագ դարձնի հավելվածի տեղադրումը:*

```
mdutils >= 1.8.1,  
rich >= 14.2.0,  
click >= 8.3.1,  
markdown_pdf >= 1.10,  
markdown >= 3.10,  
pytest >= 9.0.2
```

NerdNote | **ԳՈՐԾԱՐՔՈՒՄ ԵՎ ՇԱՅԱԳՈՐԾՈՒՄ:**

Գործիքը կարելի է տեղադրել լոկալ և գլոբալ ձևով: Տարբերությունը միայն այն է որ լոկալ տեղադրման պարագայում այն հասանելի կլինի միայն վիրտուալ միջակայքում կամ դրա ակտիվ լինելու պարագայում:

Տեղադրումը ավելի մանրամասն [ասյտել](#)

Շահագործում

- `nn add նշում` - կստեղծի md ֆայլ կգրանցի նշումը, կամ կավելացնի գոյություն ունեցողին:
- `nn ls` - Կցուցադրի տվյալների բազայում գրանցված բոլոր գրանցումները և դրանց id-ները:
- `nn cat id` - Կցուցադրի md ֆայլը ըստ տրված id-ի, (id կարող է լինել միայն թվային արժեք) տերմինալում:
- `nn rm id` - Կջնջի ֆայլը ըստ տրված id-ի, (id կարող է լինել միայն թվային արժեք) և գրանցումը:
- `nn export id pdf` - կվերափոխի ֆայլը ըստ տրված id-ի, (id կարող է լինել միայն թվային արժեք) pdf ֆոմատի:

nerdnote.modules.dbase | MODULE

Մոդուլը կատարում է բոլոր հնարավոր մանիպուլացիաները կապված տվյալների բազայի հետ: Ընդգրկում է իր մեջ բազայի հետ կապ, աղյուսակի ստեղծում, տվյալների գրանցում, տվյալների ընտրումը բազայից և տվյալների ջնջումը այնտեղից:

class *nerdnote.modules.dbase*. **NoteDataBase**

Bases: DecoMixin

Դասը իր վրա է վերձևում մոդուլին վերագործած բոլոր վերոհիշյալ գործողությունները:

NoteDataBase. **connection**(*content*, *values=None*)

Կապահովվի SF ավտանգ միացումը, տվյալների գրանցումը եթե դրանք առկա են, իսկ եթե ոչ բոլոր հարցումների իրականացումը բազայի հետ: Ունի որոշիչ դեր այս մոդուլում, մնացած ֆունկցիաների լոգիկան կապված է այս ֆունկցիայից:

- **Parameters:**

- **content** (str) - հարցումների պարունակությունը:
- **values** (tuple | None) - գրանցման ենթակա տվյալներ:

- **Returns:**

- **Return type** - sqlite3.cursor
- **Return** - Հարցման պատասխանը պարունակող բազայի կուրսոր օբյեկտ.

NoteDataBase. **_create**()

Ստեղծում է անհրաժեշտ աղյուսակը և անհրաժեշտ տվյալների կազմությունը:

- **Returns:**

- **Return type** - sqlite3.cursor
- **Return** - Աղյուսակի գրանցման հարցում:

NoteDataBase. **insert**()

Կատարում է վերոհիշյալ _create ֆունկցիան այսպիսով ստեղծում աղյուսակը իր արժեքներով: Ստուգում որ նման տվյալներ առկա չեն, դուբլիկատներից խուսափելու համար: Գրանցման գործընթացը իրենից ներկայացնում է Ֆայլի անվան գրանցում և դրան ստեղծման ամսաթվի գրանցում, յուրաքանչյուր գրանցում ունենում է իր ունիակալ id-ին:

- **Returns:**

- **Return type** - sqlite3.cursor
- **Return** - Աղյուսակի գրանցում և տվյալների մուտքագրում:

`NoteDataBase. select (id=None)`

Ֆուկցիան նախատեսված է կատարելու `select` կամ տվյալների ստացման հարցում: Այն աշխատում է երկու ռեժիմում տվյալների **հատական** որը կատարվում է ընտ դրանց ունիկալ `id`-ի և **ամբողջական** բոլոր առկա գրանցումների ցուցակի արտաբերման:

- **Parameters:**

- **id** (`int | None`) - Գրացման իդենտիֆիկացիոն համար:

- **Returns:**

- **Return type** `list`
- **Return** - Գրացման կամ գրանցումների ցուցակ:

`NoteDataBase. drop (id)`

Կատարի գրանցման ջնջման գործընթացը ըստ դրա ունիկալ `id`:

- **Parameters:**

- **id** (`int`) - Գրացման իդենտիֆիկացիոն համար:

- **Returns:**

- **Return type** `sqlite3.cursor | None`
- **Return** - Չնջումը կատարող Հարցման օբյեկտ:

`NoteDataBase. is_record_exists (id=None)`

Կատարում է հարցում պարզելու համար այդո՞ք գրանցումը առկա է: Ֆուկցիան նույնպես հարցում է կատարում երկու սցենարով, ըստ `id` կամ անվանական փնտրտուք:

- **Parameters:**

- **id** (`int | None`) - Գրացման իդենտիֆիկացիոն համար:

- **Returns:**

- **Return type** `bool`
- **Return** - կվերադառնա բուլային արժեք (`True | False`):

nerdnote.modules.mdown | MODULE

Մոդուլը կապված է տնօրինելու բոլոր գործընթացները կապված markdown ֆայլերի հետ:

class *nerdnote.modules.mdown*. **MarkdownInit**

Bases: MdUtils, MarkdownFile, DecoMixin

Կատարում է markdown ֆայլի ստեղծում, նշուների գրանցում և վերափոխումը այլ (*pdf, html*) ֆորմատի:

MarkdownInit. **create_note**(*note*)

Ֆունկցիան կստեղծի ըստ ընթացիկ ամսաթվով ֆայլ, այտեղ նշուները գրանցելու համար, նշուներից յուրաքանչյուրը կգրանցվի գրանցման ժամով ենթավերնագրի տակ: Եթե ֆայլը արդեն առկա է այն ուղղակի կավելացնի նշումը ֆայլի վերջում դրա գրանցման ժամով:

- **Parameters:**

- **note**(str) - նշուման պարունակությունը մուտքագրված օգատատիրոջ կողմից:

MarkdownInit. **exporter**(*path, raw_file, pattern*)

Ֆունկցիան կկատարի վերափոխումը ըստ տրված պատտերնի և կպահմանի այն տրված ուղղությամբ, լռելայն ուղղությունը այն դիրեկտորիան է որտեղ գտնվում են SF ֆայլը և մնացած մարկդաուն ֆայլերը:

- **Parameters:**

- **path**(str) - Ելքային ֆայլի պահպանման ուղղությունը:
- **raw_file**(str) - Մարկդաուն ֆայլի պարունակությունը տողային ֆորմատով:
- **pattern**(str) - Ելքային ֆոխմատի ցուցումներ:

nerdnote.modules.controller | MODULE

Մոդուլը կատարում է խմբագրիչի դեր, այն կապում և կարգավորում է մնացած մոդուլների լոգիկական բերելով դրանք վերջնական տեսքի:

class *path.module.script_file_name*. **NoteCommandController**

Bases: NoteDataBase, MarkdownInit

Դասը կատարում է CLI առկա հրամանների ռեալիզացիան:

NoteCommandController. **add** (**note**)

Ֆունկցիան ընդունում է նշումները տողի տեսքով կատարում դրանց գրանցումը ֆայլի և համապատասխան գրանցումը տվյալների բազայում: Եթե վերջիններս բացակայում են դրանք ստեղծվեն ավտոմատ կերպով: Եթե Հավելվածը գործարքվում է առաջին անգամ, ապա կստեղծվի նաև դիրեկտորիա որում պետք է պահպանվեն դրանք (լռելայն "Documents/NNotes"):

- **Parameters:**

- **note** (str) - նշուման պարունակությունը մուտքագրված օգտատիրոջ կողմից:

NoteCommandController. **ls** ()

Ֆունկցիան կցուցադրի բոլոր առկա գրանցումները, դասավորված ըստ id-ի: Դիանք կցուցադրվեն աղյուակում, իդենտիֆիկացիոն համար, ֆայլի անվանում, ստեղծման ամսաթիվ: Գրացումների բացակայության պարագյում կգգուշացնի այդ մասին:

NoteCommandController. **rm** (**id**)

Ֆունկցիան կատարում ջնջման գորընթացը ըստ ստացված id-ի: Այն կպահանջի օգտատիրոջ կողմից կրկնակի հաստատում: Հաստատումից հետո միայն Ֆայլը և գրացումը այդ մասին կջնվեն անվերադարձ:

- **Parameters:**

- **id** (int) - Գրացման իդենտիֆիկացիոն համար:

NoteCommandController. **cat** (**id**)

Ֆունկցիան կարտաբերի մարկդաուն ֆայլը ըստ տրված id-ի, միանգիամից ռենդերելով այն տերմինալում: Տվյալների բացակայության դեպքում կգգուշացնի այդ մասին: Օգտատերը կարող է այն կարդալ: Ֆունկցիան չունի ֆայլը փոխելու հնարավորություն:

- **Parameters:**

- **id**(int) - Գրացման իդենտիֆիկացիոն համար:

NoteCommandController. **export(id, pattern)**

Ֆունկցիան կնդունի իդենտիֆիկացիոն համարը և վերափոխումն պատտերնը, Այն կվերափոխի տվյալները ըստ այդ պատտերնի (pdf կամ html): Իսկ վերջում կհայտնի վերափոխված տվյալների պահպանման վայրի մասին:

- **Parameters:**

- **id**(int) - Գրացման իդենտիֆիկացիոն համար:
- **pattern**(str) - Պատտերնի ընդունելի ֆորմատներ:

NoteCommandController. **is_exists(id)**

Ֆունկցիան կստուգի արդյո՞ք տված իդենտիֆիկացիոն համարի անվամբ գոյություն ունի համանուն ֆայլ, եթե կա ապա կվերադարձնի դրա ուղղին, եթե ոչ կհայտնի այդ մասին:

- **Parameters:**

- **id**(int) - Գրացման իդենտիֆիկացիոն համար:

- **Returns:**

- **Return type** str | None
- **Return** - Այն կվերադարձնի ֆայլի ճանապահը եթե այն առկա է հակառակ դեպքում None:

NoteCommandController. **note_mkdir()**

Ֆունկցիան ստուգում է տրված դիրեկտորիային առկայությունը: Երաշխավորում է որ այն գոյություն ունի: Եթե ոչ ստեղծում է այն:

nerdnote.modules.tools.tool | MODULE

Մոդուլը ապահովվում է դեկորացիոն, ժամային, ամսաթվային տվյալների առկայությունը բոլոր այլ մոդուլների համար:

class *nerdnote.modules.tools.tool*. **DecoMixin**

Bases: Object

Դասը ինիցիալիզացնում է ֆորմատավորված ամսաթիվը, ժամը: Տրամադրում է արտաբերման աղյուսակի ֆորմատավորված օբյեկտը:

DecoMixin. **table_gen()**

Ֆուկցիան կամզում է արտաբերման աղյուսակը, օժտելով նրան համապատասխան դաշտերով, տալիս դրանց գույնը, այլ պարապետրեր, վերադարձնում է այն որպես մեկ-ամբողջական օբյեկտ:

- **Returns:**

- **Return type** rich.Table
- **Return** - կվերադարձնի պատրաստի ֆորմատավորված աղյուսակը որպես Table օբյեկտ: