PHP 10권 – 페키지 배포

**11. 패키지 배포**

====================

작성한 페키지를 배포를 하기 위해서는 페키지스트에 등록을 해야 합니다.

패키지스트는 자체적인 소스 저장소를 가지지 않고, 외부의 형상관리(Git) 서비스와 연계하여 동작을 하게 됩니다. 따라서 작성한 패키지를 Git으로 형상관리를 한후에, Git헙브와 같은 공개 저장소에 등록을 합니다.

등록된 공개 저장소를 다시 페키지스트에 등록을 함으로써 페키지의 배포가 됩니다.

**11.1 Git**

작성한 페키지를 Git으로 형상관리를 합니다.

|  |
| --- |
| cd packages\jiny\hello |

먼저 작성한 페키지의 작업 디렉토리로 이동을 합니다.

**11.1.1 초기화**

깃 초기화 명령을 통해서 깃을 레포지토리를 생성을 합니다.

|  |
| --- |
| D:\htdocs\composer\packages\jiny\hello>**git init**  Initialized empty Git repository in D:/htdocs/composer/packages/jiny/hello/.git/ |

**11.1.2 등록과 커밋**

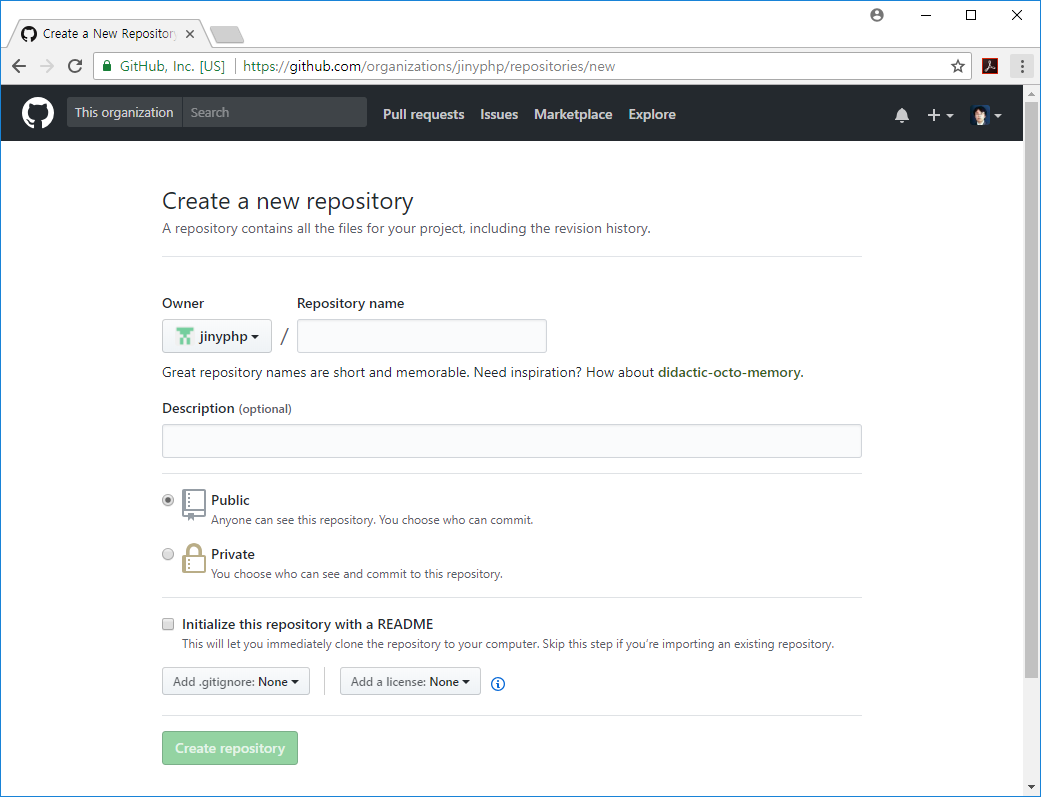
|  |
| --- |
| D:\htdocs\composer\packages\jiny\hello>**git add .**  The file will have its original line endings in your working directory.  D:\htdocs\composer\packages\jiny\hello>**git commit -m "First commit"**  [master (root-commit) 5e433c3] First commit  2 files changed, 34 insertions(+)  create mode 100644 composer.json  create mode 100644 src/HelloMessage.php |

**11.2 깃허브 등록**

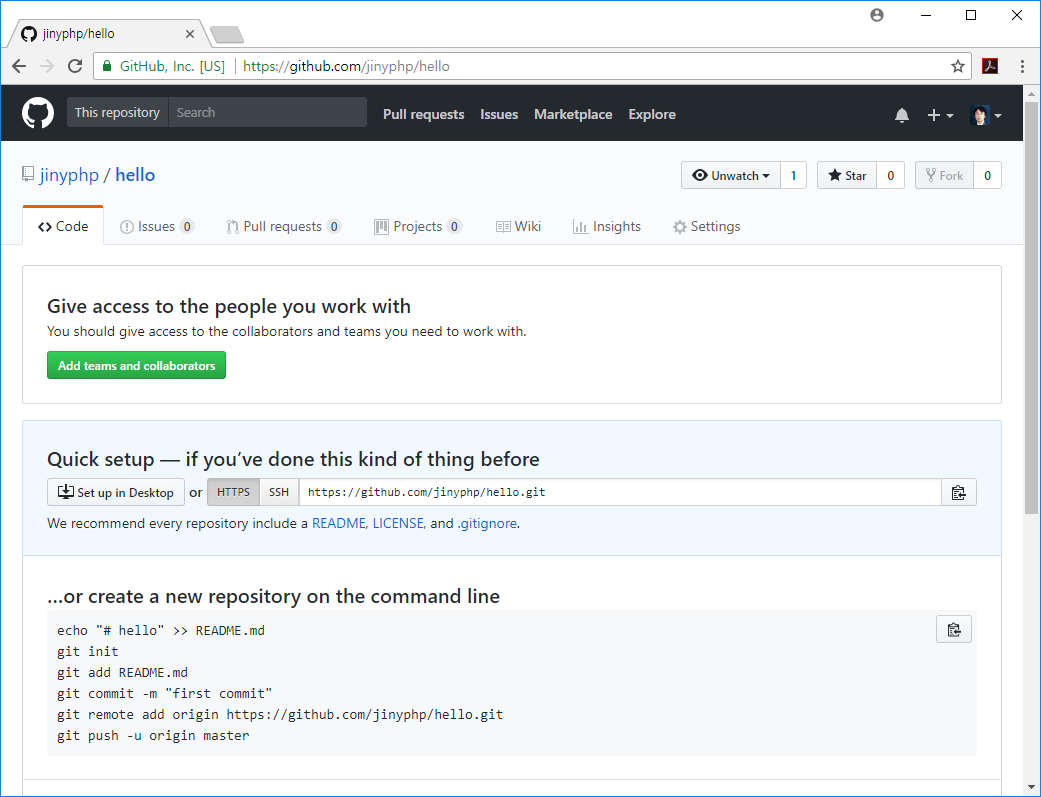
깃을 통하여 관리되는 페키지를 깃허브와 같은 공개된 저장소에 등록을 합니다. 깃허브는 무료로 사용을 할 수 있습니다. 깃허브를 사용하기 위해서는 먼저 회원등록이 되어 있어야 합니다.

**11.2.1 저장소 생성**

회원 가입후에 자신의 페키지 이름으로 저장공간을 하나 만듭니다.



저장소를 생성하면 다음과 같이 초기화 설정방법을 알려 줍니다.

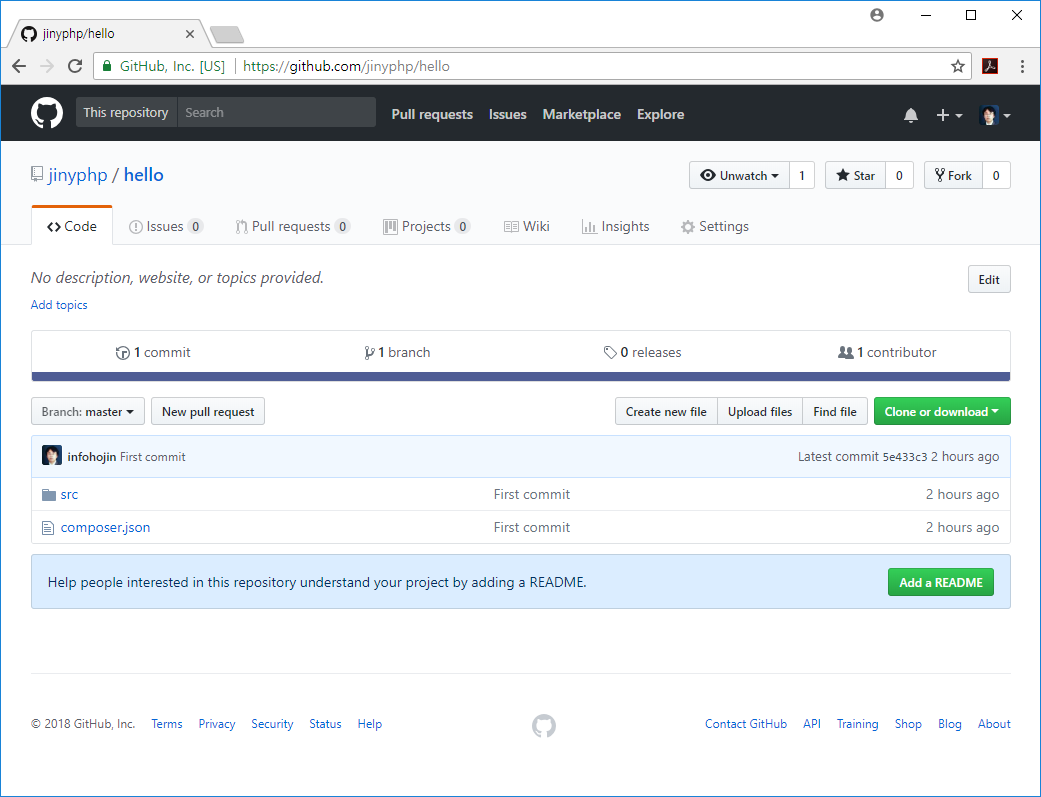


**11.2.2 패키지 저장**

페키지 작업폴더를 깃허브 원격 저장소와 연결을 합니다. 처음 깃 저장소를 생성을 하게 되면 친절하게 원격저장소로 연결할 수 있는 방법을 알려 줍니다.

|  |
| --- |
| D:\htdocs\composer\packages\jiny\hello>**git remote add origin https://github.com/jinyphp/hello.git**  D:\htdocs\composer\packages\jiny\hello>**git push -u origin master**  Counting objects: 5, done.  Delta compression using up to 8 threads.  Compressing objects: 100% (4/4), done.  Writing objects: 100% (5/5), 699 bytes | 349.00 KiB/s, done.  Total 5 (delta 0), reused 0 (delta 0)  To https://github.com/jinyphp/hello.git  \* [new branch] master -> master  Branch 'master' set up to track remote branch 'master' from 'origin'. |

깃허브 저장소를 다시 확인합니다. 정상적으로 파일들이 업로드 된 것을 확인할 수 있습니다.

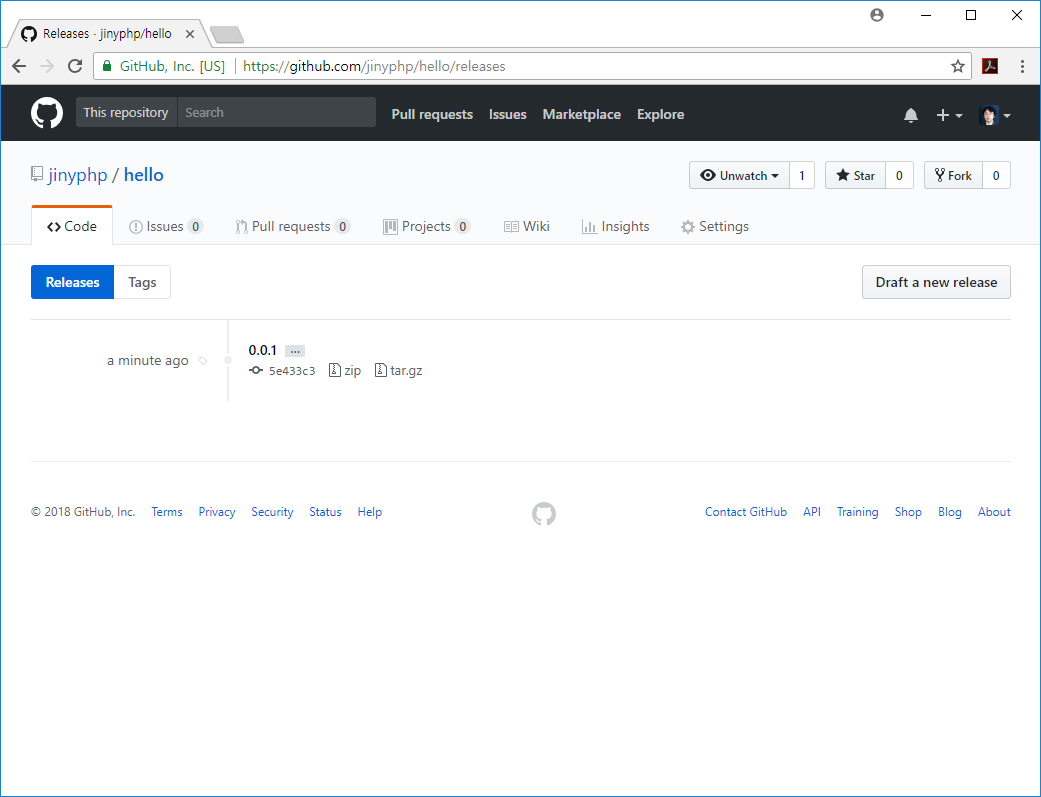


**11.2.3 테그 및 등록**

패키지의 버전을 관리할 수 있도록 테그를 설정합니다. 깃 테그 명령을 입력후에 데크부분도 같이 업로드 합니다.

|  |
| --- |
| D:\htdocs\composer\packages\jiny\hello>**git tag -a 0.0.1 -m "First Version"**  D:\htdocs\composer\packages\jiny\hello>**git push --tags**  Counting objects: 1, done.  Writing objects: 100% (1/1), 163 bytes | 163.00 KiB/s, done.  Total 1 (delta 0), reused 0 (delta 0)  To https://github.com/jinyphp/hello.git  \* [new tag] 0.0.1 -> 0.0.1 |

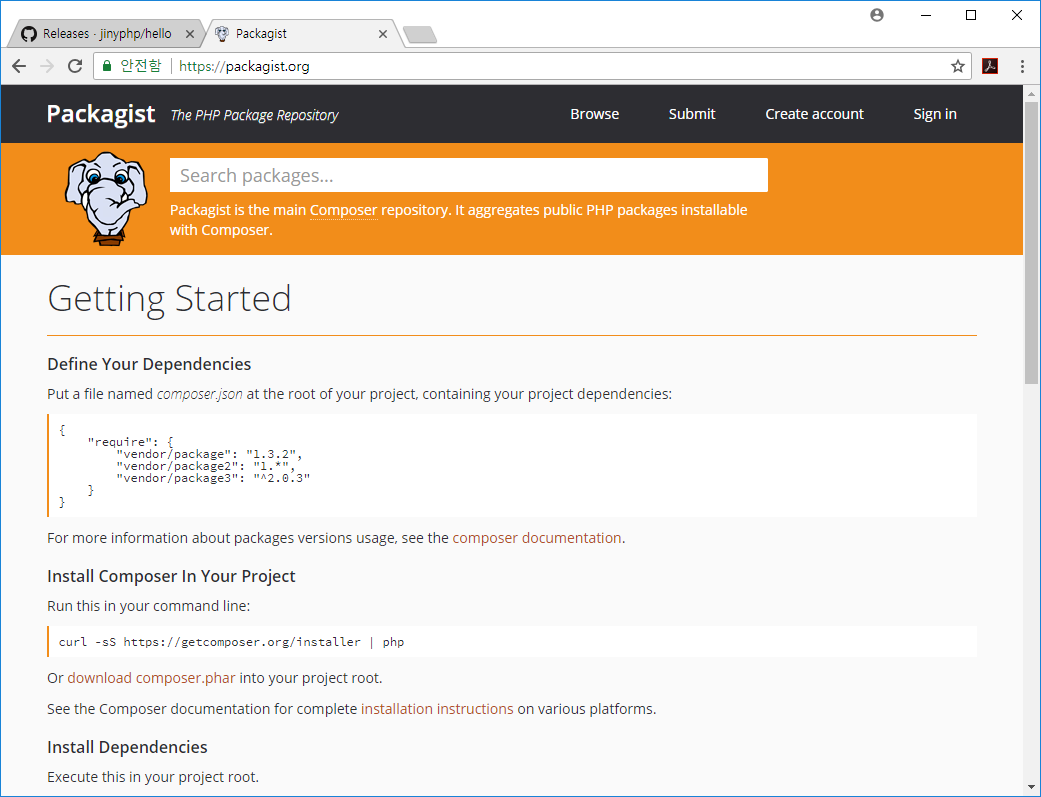
깃허브 저장소 이름아래에 있는 텝메뉴에서 Release를 선택합니다. 번호가 한단계 늘어나서 1로 변경된 것을 확인해 볼 수 있습니다.



이제 만들어 놓은 페키지를 깃허브에 등록을 하였습니다. 이제 페키지스트로 이동하여 페키지를 등록하고 배포를 시작합니다.

**11.3 페키지스트 등록**

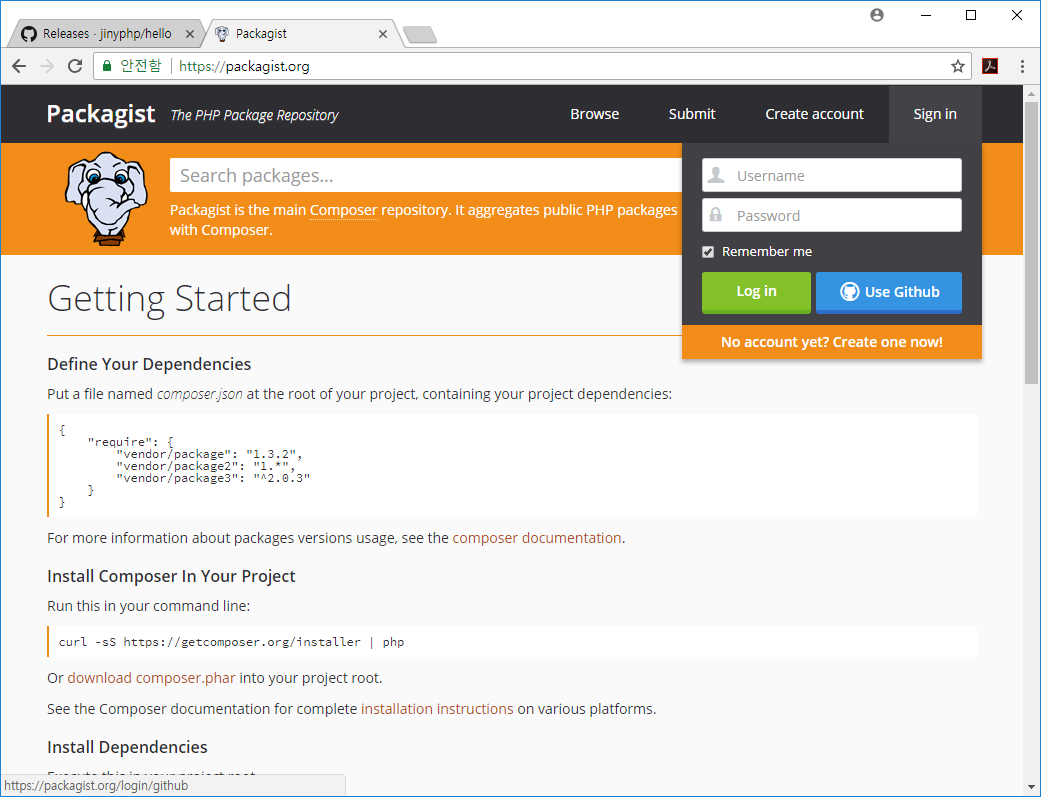
깃허브에 등록한 페키지를 컴포저의 페키지저장소인 페키지스트에 등록을 하도록 하겠습니다.



**11.3.1 회원가입**

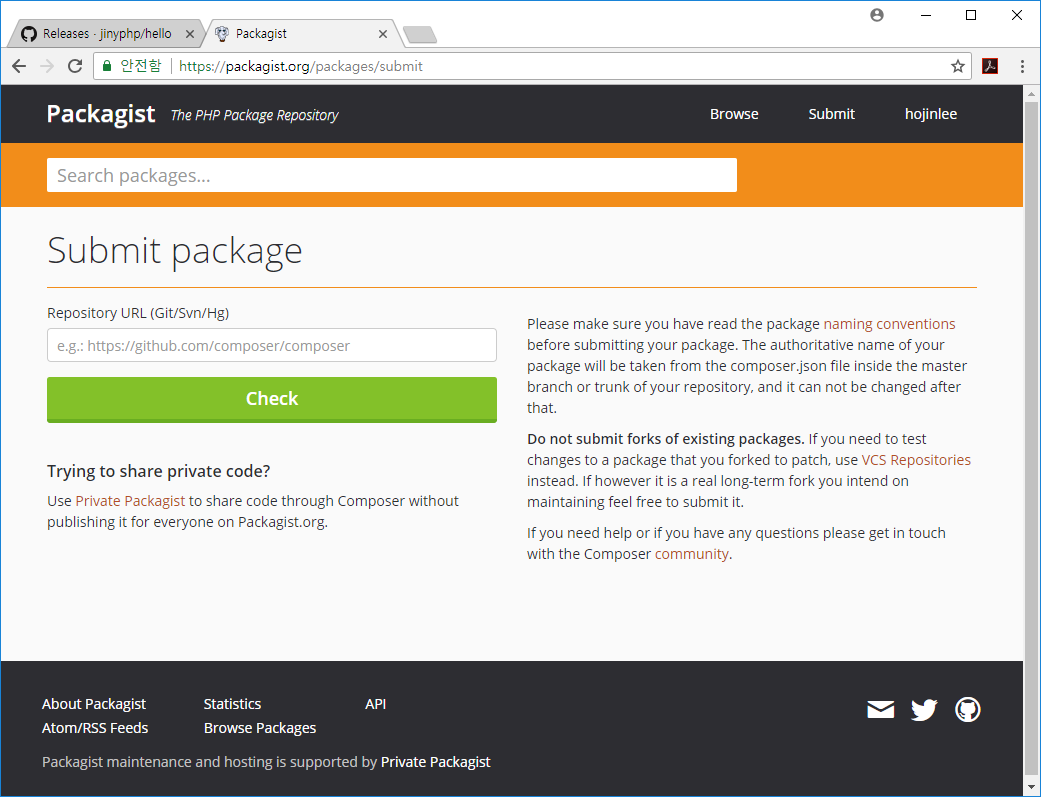
페키지를 사용만 한다고 하면 페키지스트에 회원가입을 할 필요는 없습니다. 하지만, 페키지를 만들어 배포를 해야 한다면 이는 계정이 필요할 것입니다.

먼저 회원등록을 통하여 계정을 생성을 하도록 합니다. 또는 깃허브의 계정을 통하여 로그인을 할 수도 있습니다.



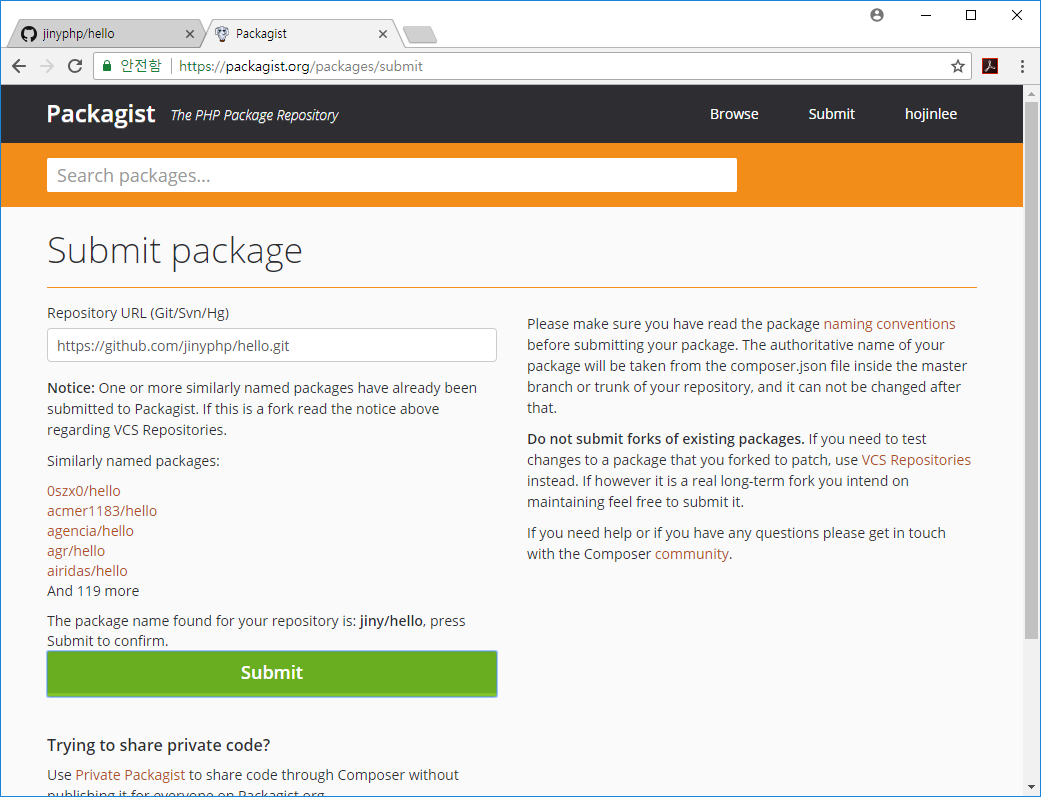
**11.3.2 등록 제출**

깃허브에 공개한 페키지를 등록을 합니다. 상단 메뉴에 있는 “submit”을 선택합니다.

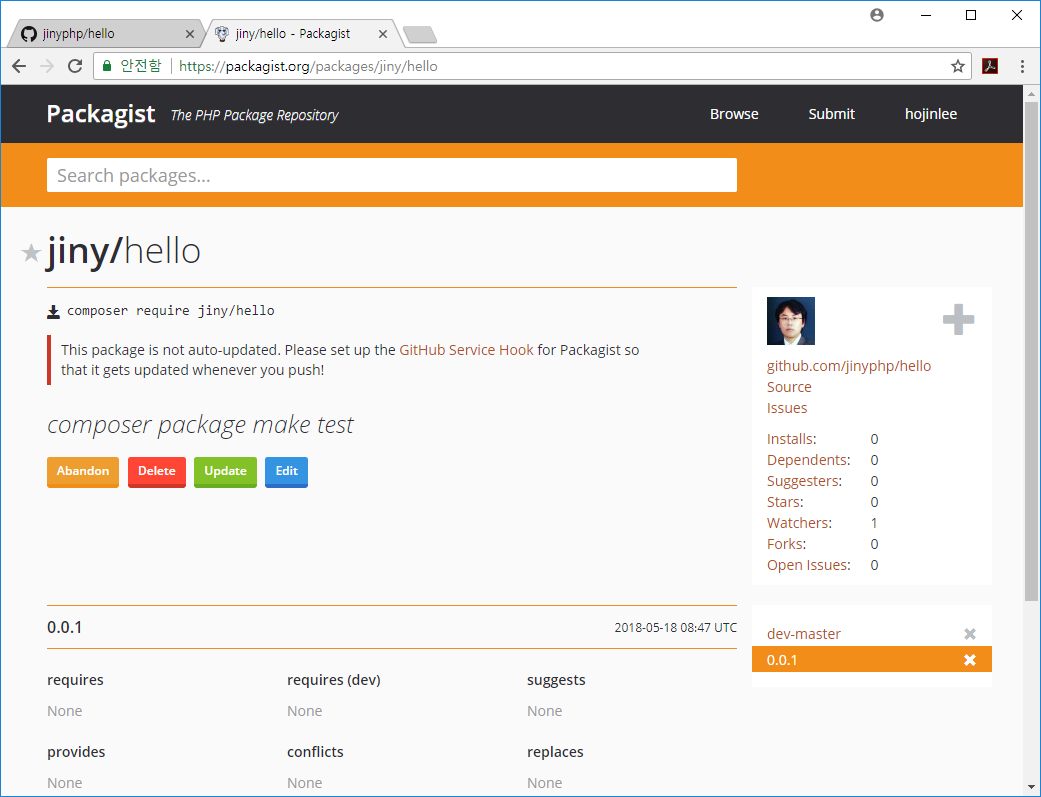


깃 저장소의 주소를 입력하는 곳이 있습니다. 주소는 깃허브의 저장소에서 확인을 할 수 있습니다. “check”를 선택하여 등록을 진행합니다.

문제가 없다고 판단이 되면 “submit” 버튼이 활성화 됩니다.



서브밋을 다시 확인하시면 등록이 완료가 됩니다.



페키지를 배포하기 위해서는 벤더명이 중복되면 안됩니다. 벤더명은 페키지를 배포하는 개발자를 구분하기 위함입니다.

만일 자신의 벤더명이 중복이 되어 있다면, 다른 이름으로 새롭게 만들어야 합니다.

**11.4 webHook**

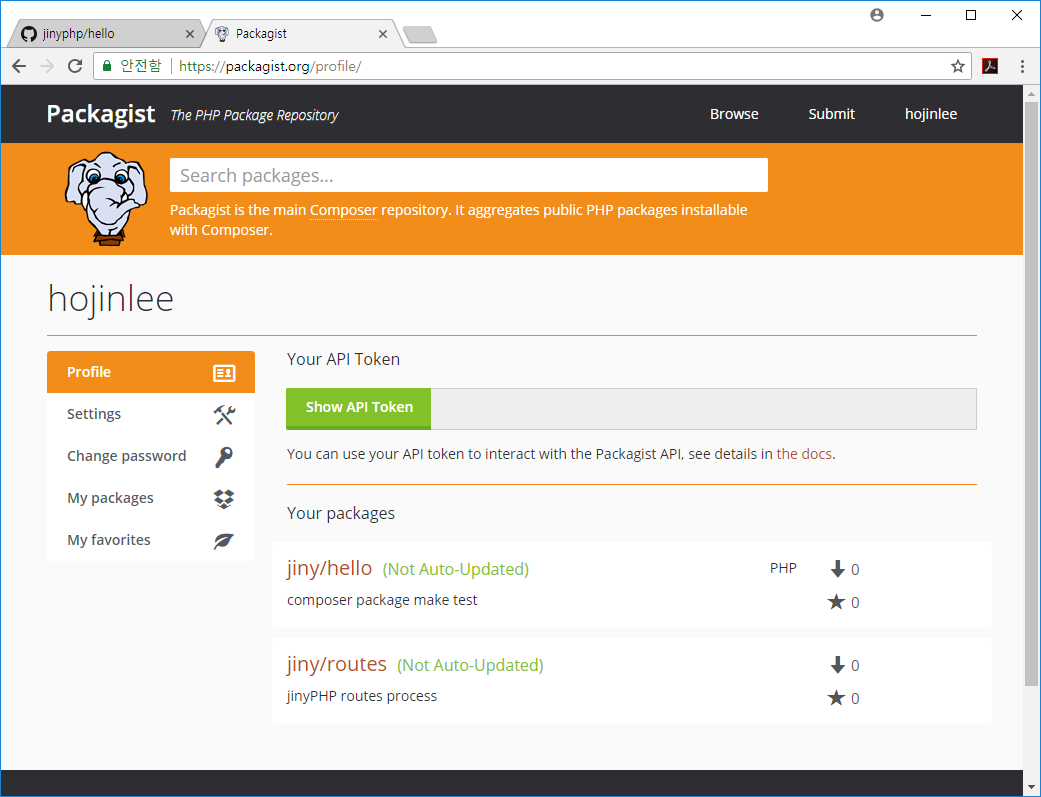
만일 패키지를 수정후에 깃허브에 등록을 할때마다 패키지스트의 내용을 재갱신해주어야 하는 번거로움이 있습니다.

웹훅 기능을 이용하면, 깃허브의 내용이 갱신될때마다 자동으로 페키지스트의 내용을 갱신처리 할 수 있습니다.

11.4.1 API 토큰얻기

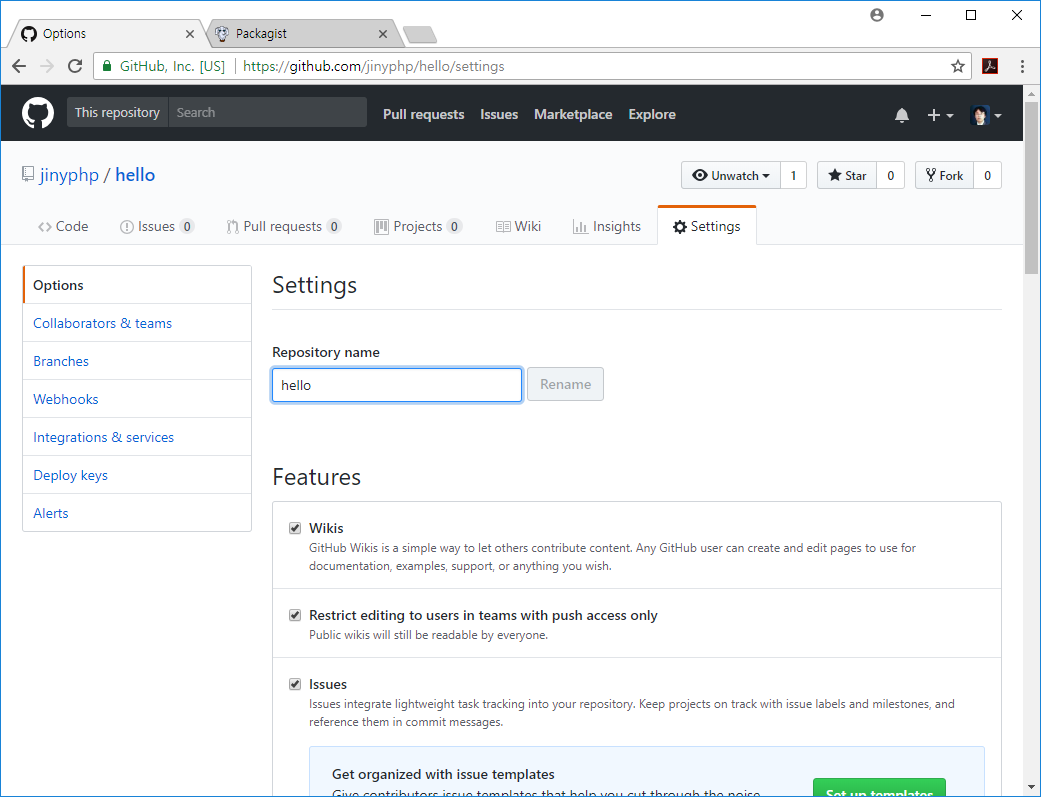
패키지스트 우측상단의 자신의 계정이름을 클릭하시면 본인이 등록한 페키지들의 목록을 확인할 수 있습니다.

“Show API Token” 버튼을 클릭하면 우측에 토큰기를 생성하여 출력해 줍니다.

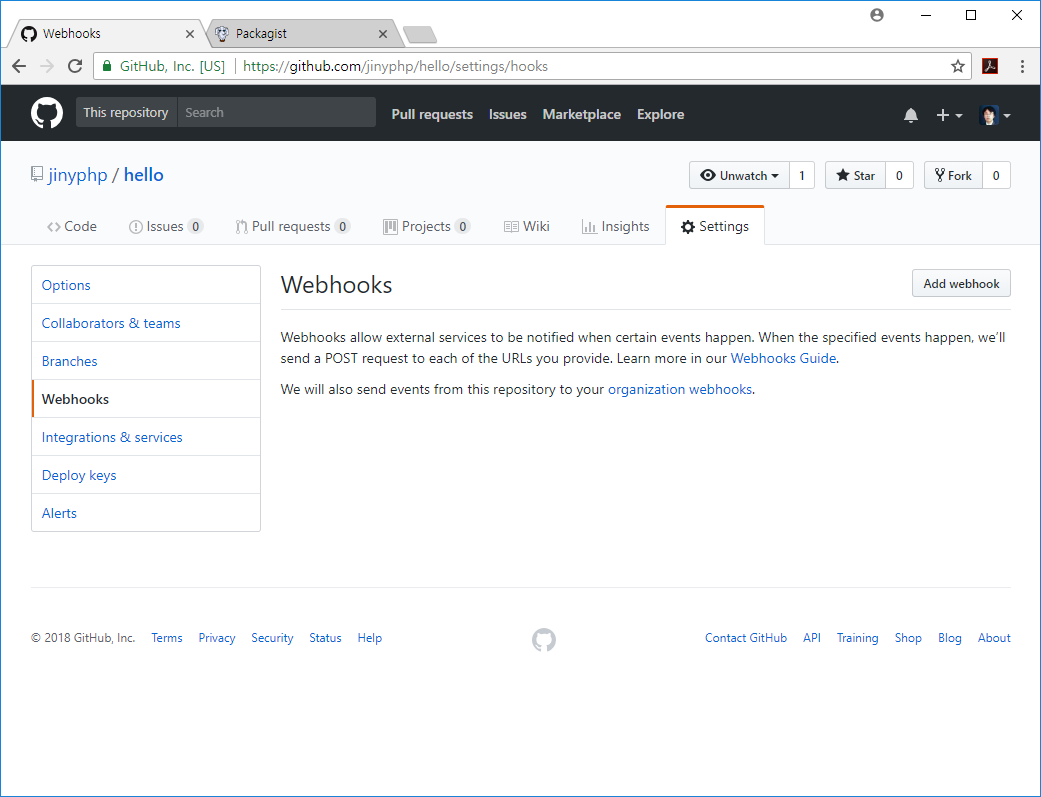


이 토큰키를 복사하여 깃허브 저장소에 등록해 주도록 합니다.

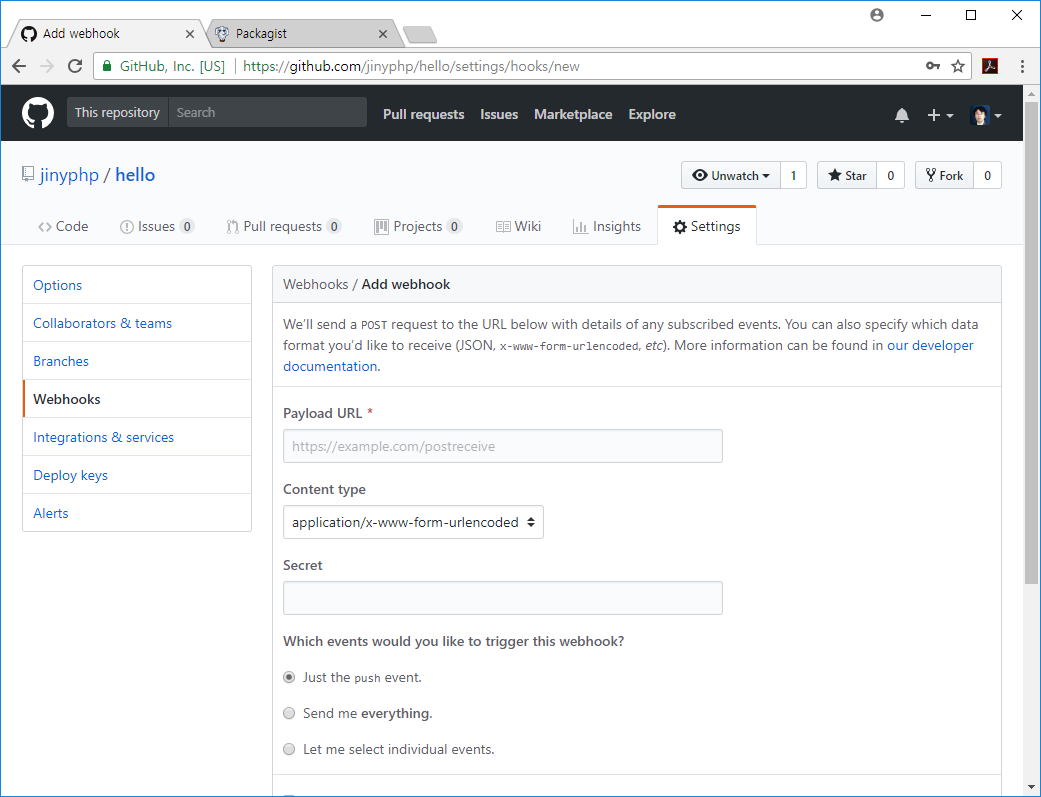
깃허브 저장소로 이동하여 저장소의 “Setting”을 부분을 선택합니다.



좌측 메뉴에 보면 WebHooks 메뉴를 확인할 수 있습니다.

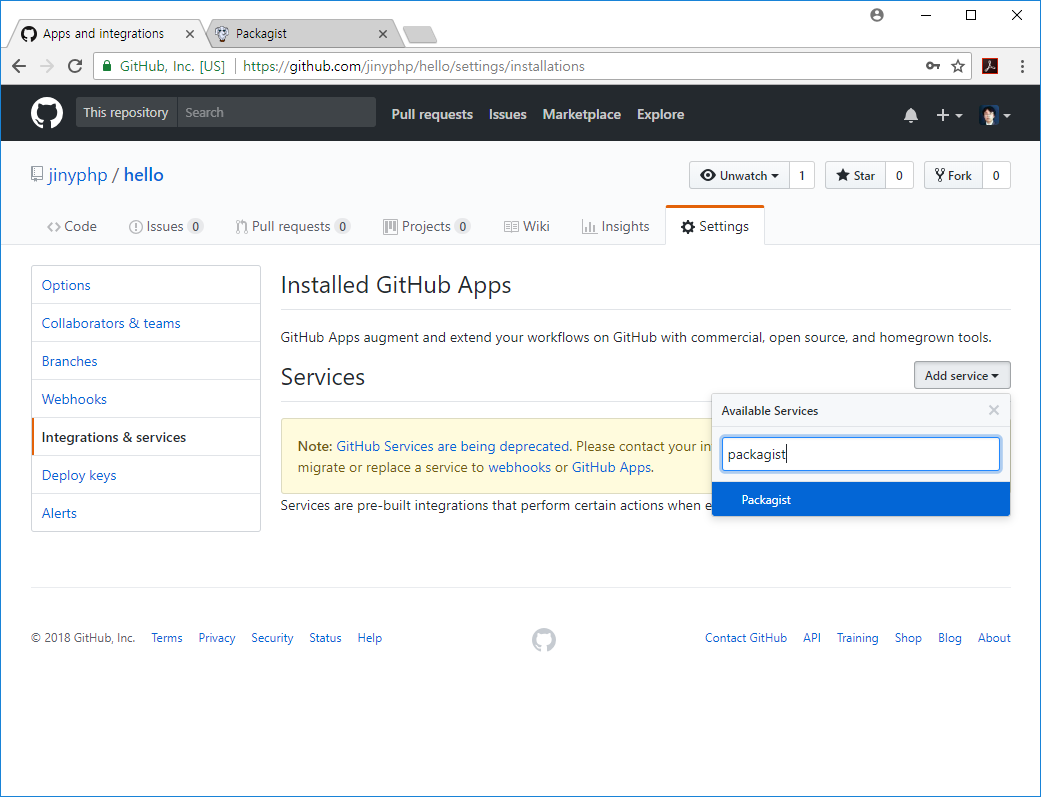


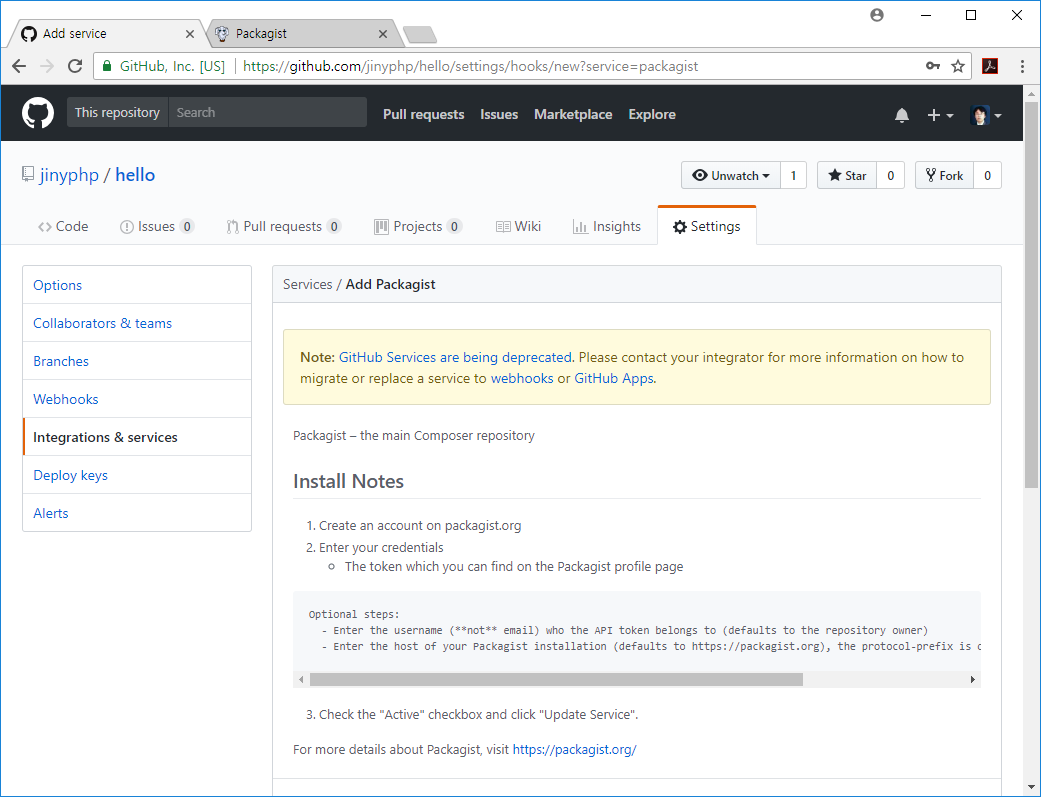
“add webhook”을 선택합니다.

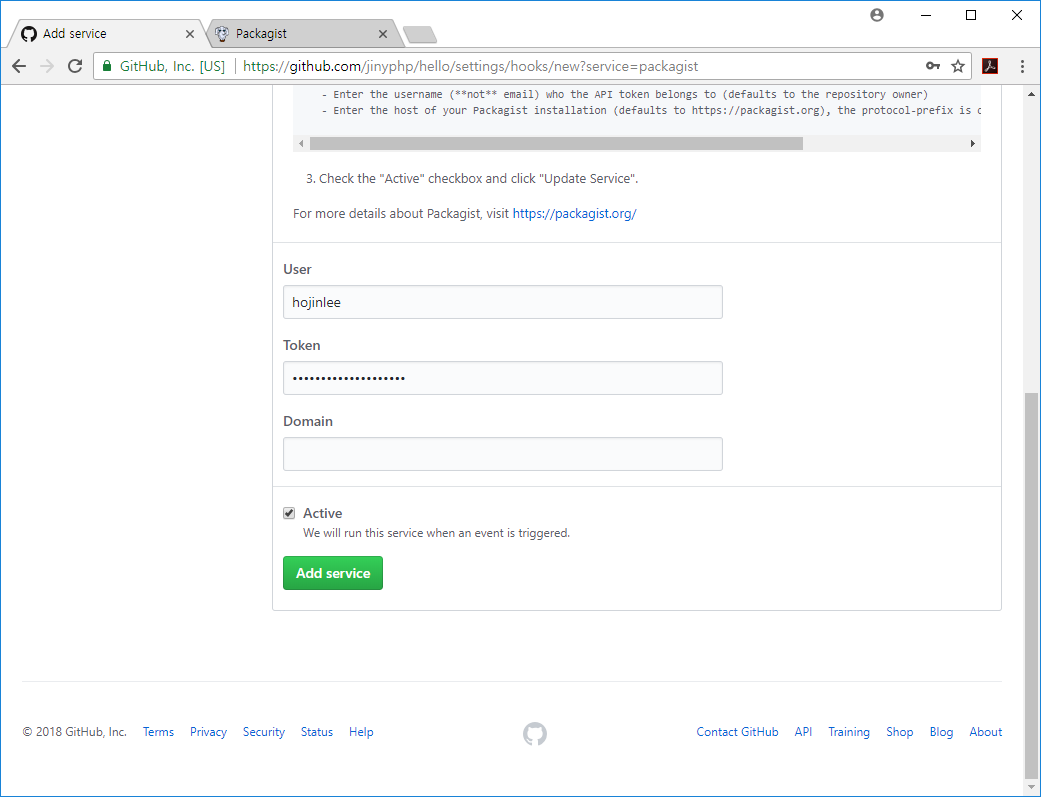


“Integrations & service”를 선택합니다. “add service”를 선택합니다.

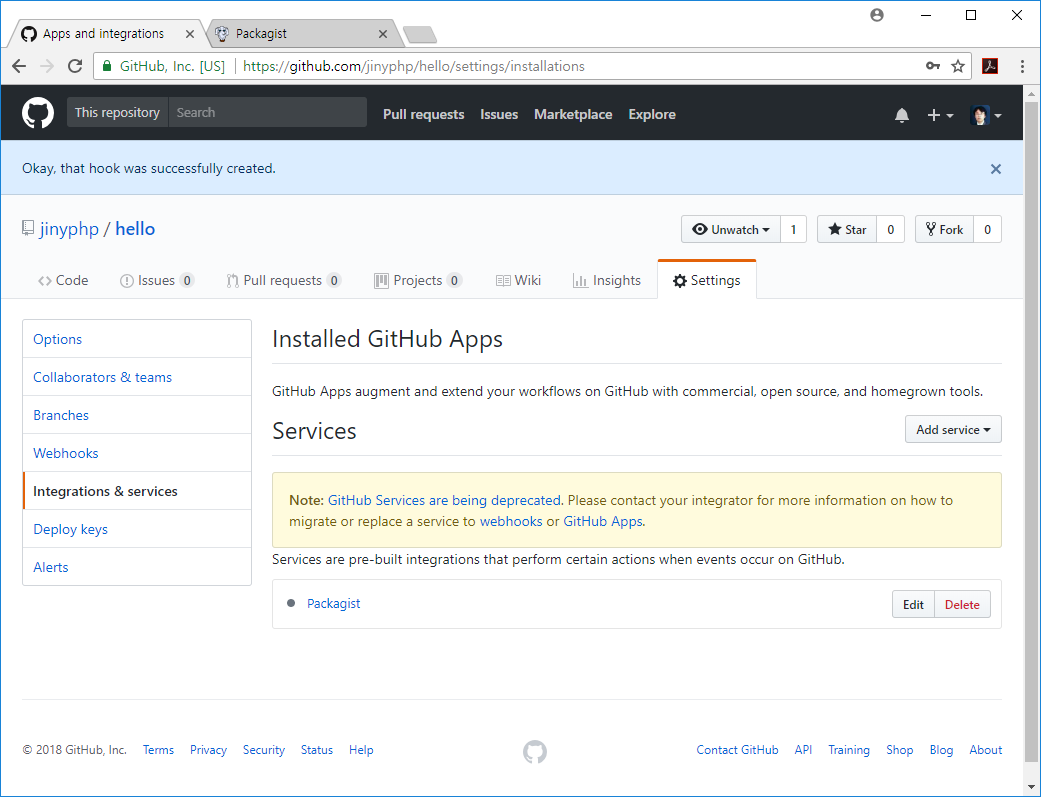
서비스 목록에서 packagist를 선택합니다.



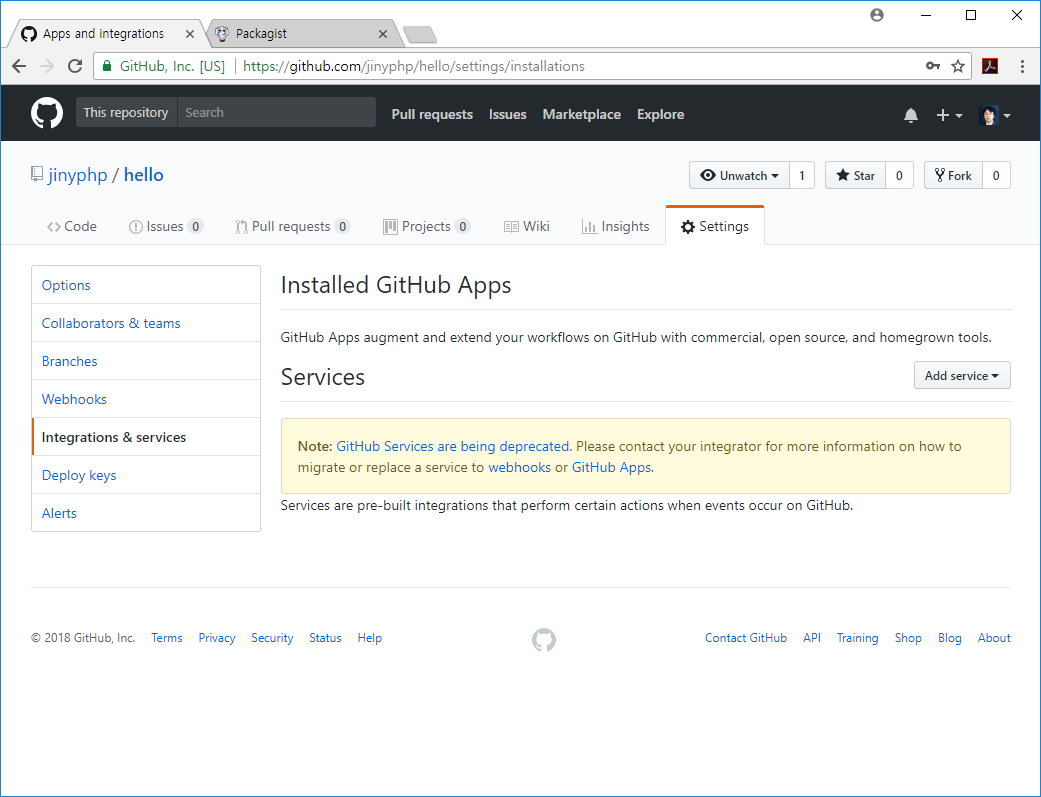




페키지스트 사이트에서 복사한 토큰키를 저장후에 등록을 합니다.



등록된 것을 확인할 수 있습니다.



**11.5 패키지 설치**

이제 다시 등록한 페키지를 컴포저로 설치하여 실행해 보도록 하겠습니다.

**11.5.1 되돌리기**

앞에서 등록한 오토로드를 원상태로 되돌립니다. 추가한 내용을 삭제하고, dumpautoload를 실행합니다.

그러면 아래와 같이 PHP는 관련 클래스를 찾을 수 없어 오류를 발생하게 됩니다.

[Fri May 18 20:19:07 2018] PHP Fatal error: Uncaught Error: Class 'Jiny\Hello\HelloMessage' not found in D:\htdocs\composer\index.php:6

St[Fri May 18 20:19:07 2018] ::1:55047 [500]: / - Uncaught Error: Class 'Jiny\Hello\HelloMessage' not found in D:\htdocs\composer\index.php:6

Stack trace:

#0 {main}

thrown in D:\htdocs\composer\index.php on line 6

**11.5.2 패키지 설치**

|  |
| --- |
| D:\htdocs\composer>**composer require jiny/hello**  Using version ^0.0.1 for jiny/hello  ./composer.json has been updated  Loading composer repositories with package information  Updating dependencies (including require-dev)  Package operations: 1 install, 0 updates, 0 removals  - Installing jiny/hello (0.0.1): Downloading (100%)  Writing lock file  Generating autoload files |

패키지가 성공적으로 설치가 되었습니다. 컴포저의 패키지를 담고 있는 /vendor 디렉토리를 확인해 봅니다.

D:\htdocs\composer**\vendor**>dir/w

D 드라이브의 볼륨: dev

볼륨 일련 번호: 1669-566C

D:\htdocs\composer\vendor 디렉터리

[.] [..] autoload.php [bin] [composer] [doctrine] [illuminate] **[jiny]**

[nesbot] [psr] [symfony]

1개 파일 178 바이트

10개 디렉터리 537,091,948,544 바이트 남음