

Patch files

Patch files toepassen via Composer en zelf patch files maken

- [Bestaande patch file toepassen](#)
- [Zelf een patch maken](#)

Bestaande patch file toepassen

Patch file lokaal testen zonder via composer te moeten

patch file eerst opslaan in root van project (niet de web folder), commando uitvoeren vanuit root folder:

Drupal core patch:

```
git apply -v --directory=web 3061782-9.patch
```

Drupal module patch:

```
git apply -v --directory=web/modules/contrib/media_pdf_thumbnail 3061782-9.patch
```

```
git apply --check -v --directory=web/modules/contrib/views_entity_form_field 3061782-9.patch
```

Zelf een patch maken

Zelf een patch maken voor een contrib drupal module

-> via git clone project binnenhalen, bv:

```
git clone https://git.drupalcode.org/project/formtips.git
```

```
git checkout -b [issue-number]-[short-description]
```

voorbeeld:

```
git checkout -b 3261983-use-template
```

-> nodige aanpassingen doen in code die in de patch moeten komen

Verskil tussen huidige en vertrekkende branch bekijken:

```
git diff origin/8.x-2.x
```

Als alles juist lijkt: patch maken via git diff commando:

```
git diff HEAD > [project_name]-[short-description]-[issue-number]-[comment-number].patch
```

als je patch niet alle wijzigingen blijkt te bevatten, probeer dan --cached:

```
git diff --cached > [project_name]-[short-description]-[issue-number]-[comment-number].patch
```

Als je patch een nieuwe bestand toevoegt:

```
git add doen voor alle gewijzigde bestanden + nieuwe bestanden
```

```
git add config/schema/media_bulk_config.schema.yml
```

```
git add src/UploadRedirectManager.php
```

```
...
```

```
git diff --cached > test.patch
```

enkele voorbeelden:

```
git diff origin/8.x-2.x > view_modes_display-use-template-3261983-3.patch
```

```
git diff > colorbox-class-crypt-not-found-3163173-6.patch
```

```
git diff > seeds_toolbar-undefined-index-weight-3219355.patch
```

```
git diff > media_pdf_thumbnail-undefined-index-3244511-5.patch
```

-> patch is nu aangemaakt in de root van de drupal module, en kan je dus uploaden in een issue

op drupal.org.

eigen drupal module: Create Release

```
git tag 2.0.0
git push origin 2.0.0
-> naar module pagina gaan op drupal.org en klikken op Add new release
```

Een patch die niet volledig werkt gedeeltelijk toepassen

```
git apply --reject <patchfile>
```

Zelf een MR maken als er nog geen is:

- Klik op de issue pagina op 'Create Issue Fork'
- Klik op Show commands
- In terminal eerst project clonen: git clone ...
- Dan project openen in vscode en de commands van show commands uitvoeren

```
git remote add leaflet_countries-3595522 git@git.drupal.org:issue/leaflet_countries-3595522.git
git fetch leaflet_countries-3595522
git checkout -b '3595522-php-8.4-errors' --track leaflet_countries-3595522/'3595522-php-8.4-errors'
```

Dan nodige wijzigingen doen

- ```
git add . && git commit -am "blabla"
git push
```

- Dan in issue, op de naam klikken achter 'Issue fork' (en zien dat je daar ingelogd bent)
- Daar klikken op '**Create merge request**'

## Een nieuwe MR maken als er al een bestaande is:

Klik op de issue pagina op 'Get push access'

```
git remote add 3109763 git@git.drupal.org:issue/views_entity_form_field-3109763.git
git fetch 3109763
git checkout -b 3109763-v2 3109763/8.x-1.x
```

```
git add .
git commit -m "Fix XYZ in views_entity_form_field"
git push 3109763 3109763-v2
```