

MVC, Git it?

The name is Yoeri

@yoerivd

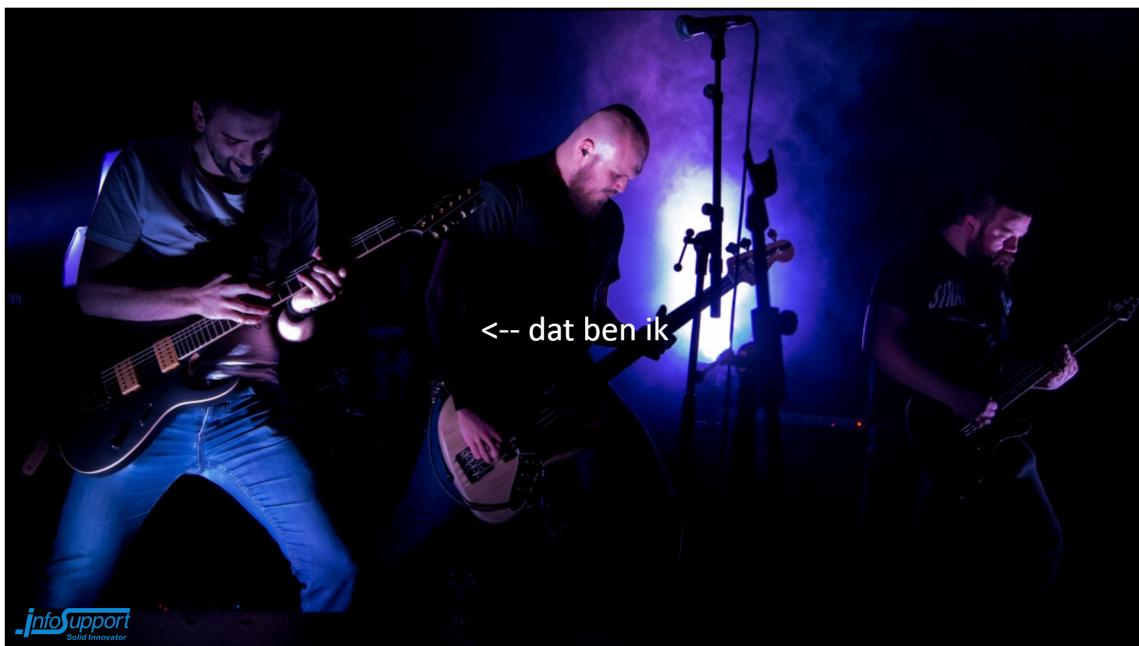
everything web - git - CI/CD - javascript - C#



InfoSupport
Solid Innovator



InfoSupport
Solid Innovator



The screenshot shows the InfoSupport artist profile for the band 'Titans'. At the top, there's a 'PLAY' button, a 'FOLLOWING' button, and a three-dot menu. Below that, tabs for 'OVERVIEW' (which is selected), 'ABOUT', and 'CONCERTS' are visible. A large image of the band members in dark clothing is displayed. In the 'Latest Release' section, a thumbnail for 'Change of Heart' is shown with the release date 'OCT 21, 2017'. The 'Popular' section lists three tracks: 'Lost Future' (1,000 plays), 'Change of Heart' (1,941 plays), and 'Vices and Virtues' (< 1,000 plays). On the right, a 'Artist's Pick' section features a video thumbnail for 'The Meltdown, Titans, Here Stands a ...' on March 24th at 8:00 pm in Wijnegem.



Inhoud

- Hoe begin ik aan een project?
- Git? Pull request? Branches? REBASE? MERGE??
- Hidden gems of Razor
- Migrations

Getting started

New project

- file -> new project
- MVC
- Geen unit tests (zelf later toevoegen)

UPDATE ALL NUGET PACKAGES!

let op: template gebruikt standaard bootstrap 3! Na de
upgrade -> bootstrap 4

<https://getbootstrap.com/docs/4.0/migration/>

T4MVC

T4MVC

weg met "magic strings"!

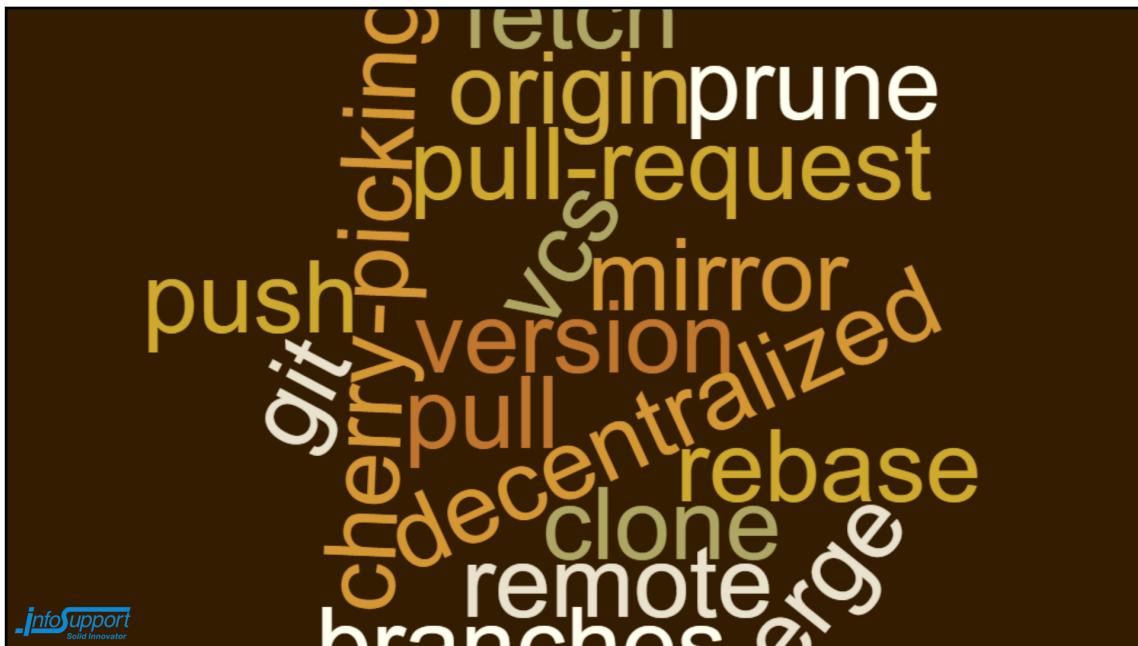
```
<ul class="nav navbar-nav">
    <li>@Html.ActionLink("Home", "Index", "Home")</li>
    <li>@Html.ActionLink("About", "About", "Home")</li>
    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
```

==>

```
<ul class="nav navbar-nav">
    <li>@Html.ActionLink("Home", MVC.Home.Index())</li>
    <li>@Html.ActionLink("About", MVC.Home.About())</li>
    <li>@Html.ActionLink("Contact", MVC.Home.Contact())</li>
</ul>
```

voordeel?

Git



waarom VCS ?





Life saving tip # 1

-> werk nooit rechtstreeks op de master branch
conflicten
lelijke history
master zou ten allen tijden moeten werken

*meeste VCS systemen kennen
'protected branches'*

Dure woordjes

Git	git it
repository	het gehele project dat in VCS zit
clone	een lokale kopie maken van het repo (om zelf te bewerken)
remote	git kan sync'en met verschillende 'remote' repositories tegelijk. Meestal github, gitlab, bitbucket,..
commit	wanneer je een wijziging aanbrengt kan je die opslaan in git, dit noemt men een commit
push	lokale commit(s) naar een remote sturen
branch	een aftakking van de 'main' branch. Bevat alle code, maar commits gebeuren enkel op die branch

Git	git it
merge	wijzigingen van een andere branch ook op de huidige branch toepassen
rebase	alternatief voor merge, negeren tot later :-)
upstream	de naam van de branch op de remote die getracked mag worden
pull request	wanneer je niet zelf een branch mag wijzigen, kan je een aanvraag doen naar iemand die het wel mag

Life saving tip # 2

-> gebruik git cmd, geen ui

Getting sources

command	uitleg
git pull	get all commits from master and merge with the active local branch
git stash	"shelve" all pending changes (useful before git pull)
git stash pop	"unshelve" previously stashed changes
git remote prune origin	lokale remotes branches die niet meer op de server bestaan ook lokaal verwijderen
git fetch	kan je een oa. branch vanuit workitem gecreeerd lokaal beschikbaar maken

Branching

command	uitleg
git checkout -b someFeatureOrBugFixName	create a new branch from the current branch and switch to the new branch
git checkout master	switch to a branch with the name master
git branch	list all branches present on this machine"
git branch -d someFeatureOrBugFixName	delete a branch with name "someFeatureOrBugFixName" (only if it's merged!)
git branch --merged	list local merged branches (normally it's safe to delete)

Committing changes

command	uitleg
git status	show unstaged and staged changes
git add *.cs	stage all unstaged files ending in ".cs"
git commit -am "some message"	add all staged changes, commit with a message

Merging

command	uitleg
git pull origin master	huidige branch up to date brengen met master
git mergetool	=> fix conflicts
git commit [...]	=> check in the merge
git push	=> sync with the master repository

Razor full power
onze eerste feature

the plan

1. maak een feature branch
2. git push --set-upstream origin <branch>
3. Dummydata tonen
4. Dummydata beter tonen :)
5. pull request maken

DEMO

DisplayTemplates

1. maak een folder DisplayTemplates in eender welke view (of in Shared voor elke view)
2. maak een nieuwe view: naamModel.cshtml
3. declareer het model type in de view

```
@model mvc_demo.web.Models.MyModel
```

4. Gebruiker volgende helper waar gewenst:

```
@Html.DisplayFor(model => model.PropertyVanHetTypeMyModel)
```

Pull requests

pull request

- niet mogelijk vanuit cmd
- handig voor reviews
- build status

AppVeyor

pull request

1. complete zorgt voor een merge op de server
2. life saving tip #3: delete source branch!

Sentiment toevoegen

the plan

1. maak een feature branch
2. Dummydata -> db data
3. migrations!
4. vaste data toevoegen
5. editor aanmaken voor onze teksten
6. overview aanpassen

life saving tip #4

migrations in een apart project stoppen

```
enable-migrations -ContextProjectName MyProject.MVC  
-StartUpProjectName MyProject.MVC  
-ContextTypeName MyProject.MVC.MyContextFolder.MyContextName  
-ProjectName MyProject
```

EditorTemplates

1. maak een folder EditorTemplates in eender welke view (of in Shared voor elke view)
2. maak een nieuwe view: naamModel.cshtml
3. declareer het model type in de view

```
@model mvc_demo.web.Models.MyModel
```

4. Gebruiker volgende helper waar gewenst:

```
@Html.EditorFor(model => model.PropertyVanHetTypeMyModel)
```

Migrations

1. voor het gemak -> ander project
2. Add-Migration -> nieuwe migratie toevoegen
3. snapshot van model
4. teams -> update snapshot met

```
add-migration KiesMaarEenNaam -IgnoreChanges
```



Vakantiewerk?

- weer rekken vullen?
- weer ploegen?
- of .. deze keer iets wat je graag doet?

neem een flyer mee!

