Juno Dokumentácia

Hierarchia entít:

* Najvyšší rodič je Project
* Project obsahuje 4 nazvem to submoduly:
  + Test Analyza
  + Test Plan
  + Test Exekúcia
  + Issues

**Test Analyza**

V test analýze v jednoduchosti vytváraš konkrétne, čo budú užívatelia testovať. Aby v tom bola nejaká hierarchia, tak je to rozdelené na:

* Najvyššie je Test Set => ten obsahuje podmnožinu Test Caseov (alebo to niekto nazýva aj Test Scenáre) => a najnižšia podmnožina, ktorá sa REALNE testuje sú Test Step.
* Čiže napríklad:
  + Test Set môže byť = Otestovanie wordu
  + Test Case = Otestovanie menu
  + Test Step = Po kliku na view, nech sa otvori submenu

--------------------------------------

To je pripravené v Test Analyze, tak z toho si dokážeme skladať Testovacie Plány.

**Test Plan**

V submodule test plan vytvárame test plany z jednotlivých Test Caseov (test scenárov).

Aby sme nemuseli pri opätovnom rovnakom testovaní vytvárať znova test plán a priradovať mu test cases, vytvárajú sa Test Sprinty, ktoré sú vo finále **testované!**

Čiže 1 test plán môže mať X test sprintov.

Napríklad:

* Test Plan = Otestovanie Webu
* Test Sprint = Otestovanie Webu vo februári, Otestovanie webu v máji

**Test Exekúcia**

V test exekúcií sa testujú jednotlivé Test Sprinty. Vo finále tester vždy testuje, respektíve nastavuje statusy Test Stepom!

**Issues**

Issues je väzba na Test Step. Čiže máme väčšinou, ale nie vždy patrí 1 ISSUE = 1 TEST STEPU.

Môže sa stať, že je chyba dajme tomu v databáze, čiže nevieš zadať Issue na konkrétny test STEP, ale zadáš issue na dajme tomu test sprint.

**Práva**

Takmer všetky akcie(kliky, add, edit,delete) sú naviazané na nejaký **PRIVILEGE.**

Na webe rozlišujeme:

1. Projektové Role – Sa nastavujú pre modul projekt
2. Webové role – sa nastavujú pre celý systém okrem modulu projekty.

Funkcionalita vychádza z Nette ACL. https://doc.nette.org/cs/2.4/access-control

Ako funguje naše ACL:

* ACL pracuje s objektom Nette\Security\Permission.
* Aby sme túto triedu nemuseli vytávrať pri každom requste, vytvára sa raz a cachuje sa.
* Invalidácia = zmena, respektíve úprava tohto objektu nastáva vždy pri:
  + Úprave práv v nastaveniach.
  + Úprave práv pri projekte
  + Úprave užívatela
* Refresh tohto objektu obsahuje trieda UserModule\Services\User:: refreshAuthorizator()