Sémantické verzování

Lukáš Meidl 3.IT.

Obsah

Sémantické verzování - základy

Číslo verzí zapisujeme ve formátu MAJOR.MINOR.PATCH. Navyšování jednotlivých čísel verzí probíhá následovně:
MAJOR – když nastala změna, která není zpětně kompatibilní s ostatními (API)
MINOR – když se přidá funkcionalita se zachováním zpětné kompatibility
PATCH – když se opravila chyba a zůstala kompatibilita
Pomocí předběžných verzí a přidáváním metadat je možné upřesnit informace. Např.: 1.0.0-alfa, 1.0.1-beta+2

Specifikace Sémantického verzování (SemVer)

Software používající Sémantické verzování, MUSÍ mít nadefinované API, buďto přímo ve zdrojovém kódu a nebo v externí dokumentaci. V obou případech to musí být hlavně přesné a komplexní.

Číslo verzí MUSÍ být ve formátu X, Y, Z. Jedná se o celá nezáporná čísla, přičemž X se NESMÍ rovnat hodnotě nula. Může se rovnat nule jen v případě, kdy se jedná o počáteční vývoj. X je číslo MAJOR verze, Y je číslem MINOR verze a Z je číslem PATCH verze, přičemž každé číslo má svoji hodnotu a navyšují se zvlášť a standardně, např.1.9.0 => 1.10.0 => 1.11.0.

Jakmile se vydá očíslovaná verze programu, NESMÍ se měnit a každá další úprava nebo oprava je vydána pod novou verzí.
MAJOR verze s hodnotou 0 (0.y.z.) je určena pro počáteční vývoj. Cokoliv se může změnit a API v tomto formátu by NEMĚLO být považováno za stabilní.

Verze 1.0.0 definuje veřejně vydané API. Způsob, jakým se dále navyšuje číslo verze je ovlivněné tímto API a jeho změnami.

Číslo PATCH (Z) MUSÍ být navýšené jenom pokud byly implentované zpětně kompatibilní opravy chyb. Oprava chyb je definována jako interní změna opravující nežádoucí chování programu.

Číslo MINOR (Y) MUSÍ být zvýšené, když byla do API přidána nová, zpětně kompatibilní funkcionalita nebo pokud byla jakákoliv funkcionalita odebrána (jako zastaralá) i pokud neovlivňuje samotný API kód. MŮŽE zahrnout i změnu PATCH verze. Číslo PATCH verze se musí vynulovat vždy, když se změní MINOR verze.

Číslo MAJOR (X) MUSÍ být zvýšené, když byly přidané změny, které způsobily zpětnou nekompatibilitu. MŮŽE zahrnout i změny v rámci MINOR a PATCH verze. Číslo MINOR i PATCH se MUSÍ vynulovat vždy, když se změní MAJOR verze.
Předběžné verze (angl. pre-release) MOHOU být označeny přidáním pomlčky a sérií identifikátorů oddělených tečkou hned za číslo PATCH verze. Identifikátory MUSÍ obsahovat pouze ASCII alfanumerické znaky a pomlčku [0-9A-Za-z-], NESMÍ být prázdné a číselné identifikátory NESMÍ obsahovat úvodní nulu. Předběžné verze mají nižší prioritu jako

související normální verze. Předběžná verze je nestabilní a nemusí splňovat požadavky a závislosti jako normální verze. Např.: 1.0.0-alpha, 1.0.0-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92.

Metadata MOHOU být označené ve verzi přidáním znaku plus (+) a sérií identifikátorů oddělených tečkou hned za číslo PATCH a nebo pomocí předběžné verze. Identifikátory MUSÍ obsahovat pouze ASCII alfanumerické znaky a pomlčku [0-9A-Za-z-], NESMÍ být prázdné a číselné identifikátory NESMÍ obsahovat úvodní nulu. Metadata by NEMĚLA hrát roli při volbě priority verze. Např.: verze 1.0.0-alpha+001, 1.0.0+20130313144700, 1.0.0-beta+exp.sha.5114f85 mají všechny stejnou prioritu.

Priorita se vztahuje na to, jak se verze vzájemně porovnávají. Priorita MUSÍ být určována rozdělením verze na MAJOR, MINOR, PATCH a identifikátory předběžných verzí – přesně v tomto pořadí (s metadaty se nepočítá).

Priorita je daná prvním rozdílem při porovnání zleva doprava přičemž čísla MAJOR, MINOR a PATCH jsou vždy porovnávána jako čísla. Např.: 1.0.0 < 2.0.0 < 2.1.0 < 2.1.1. Pokud jsou čísla MAJOR, MINOR a PATCH stejná, předběžná verze má menší prioritu, než normální.

Např.: 1.0.0-alpha < 1.0.0.Priorita pro dvě předběžné verze, které se shodují v číslech MAJOR, MINOR a PATCH musí být počítána zleva doprava od tečky oddělených identifikátorů a to do doby, dokud se nenajde rozdíl a to následujícím způsobem: (1) Identifikátory obsahující pouze číslice, jsou porovnané číselně a identifikátory s písmeny nebo pomlčkami jsou porovnávané lexikálně, zařazené podle ASCII. (2) Číselné identifikátory mají vždy nižší prioritu jak nečíslené. (3) Jsou-li všechny předchozí identifikátory v předběžné verzi stejné, tak větší množství identifikátorů značí vyšší prioritu, než s menším počtem. Např.: 1.0.0-alpha < 1.0.0-alpha.1 < 1.0.0-alpha.beta < 1.0.0-beta < 1.0.0-beta.2 < 1.0.0-beta.11 < 1.0.0-rc.1 < 1.0.0.

Proč používat Sémantické verzování?

Sémantické verzování není revoluční myšlenka a když vydáváte software, tak už pravděpodobně děláte něco podobného. Problém je, že “něco podobného” nestačí. Bez dodržování daných formálních specifikací jsou čísla pro managment závislostí v podstatě na nic. Tím, že výše uvedeným myšlenkám dáváme přesnou a jasnou definici, je lehčí komunikovat záměry Vašeho softwaru jeho uživatelům. Jakmile jsou záměry jasné a flexibilní (ale ne příliš), specifikace závislostí může začít.

Bibliografie

Jakub Křížka, Tom Preston-Werner(22.03.2020). Sémantické verzování 2.0.0. Načteno z semver.org: https://semver.org/lang/cs/