
Kapitola 1. Tutoriál SVN a Maven (pro řešitele projektů)

Obsah

Subversion	1
Úvod	1
Repozitory	1
Pracovní kopie	2
Odeslání změn	2
Řešení konfliktů	3
Další užitečné příkazy	3
Vytváření větví a značek	3
Základní zásady	3

Subversion

Úvod

Subversion je systém pro řízení verzí. Tento systém spravuje soubory a adresáře nějakého projektu v průběhu času. Systém si udržuje informace o každé změně a umožňuje získat libovolnou starší verzi projektu nebo prozkoumat historii změn. Systém také výrazně usnadňuje týmový vývoj, neboť umožňuje souběžnou práci vývojářů, přičemž hlídá jednotlivé změny, aby nedošlo ke kolizím.

Dokumentaci k systému naleznete v Subversion Book [<http://svnbook.red-bean.com/>].

Subversion je typ systému řízení verzí s centrálním serverem. Existují i distribuované systémy řízení verzí, např. GNU Arch [<http://www.gnu.org/software/gnu-arch/>].

Repozitory

Repozitory je centrální úložiště, kde je uložena hierarchie souborů projektu včetně historie všech změn. Je uložena na serveru a mohou k ní přistupovat klienti. Tito klienti mohou data do repozitory ukládat a tím je zpřístupnit ostatním klientům, nebo je mohou číst a získat tak data poskytnutá ostatními klienty.

Problém souběžné modifikace stejných souborů:

- Řešení typu Lock-Modify-Unlock
- Řešení typu Copy-Modify-Merge

Pracovní kopie

Pracovní kopie je lokální verze souborů z repozitory. Při práci s projektem používáme a modifikujeme tyto soubory a jejich změny potom hromadně posíláme do repozitory. Pracovní kopii získáme příkazem

```
svn co <url>
```

kde <url> je adresa repozitory. Pokud již máme pracovní kopii staženou a chceme pouze stáhnout případné změny, použijeme příkaz

```
svn update
```

Pokud v pracovní kopii uděláme nějaké změny a budeme se chtít vrátit k původní verzi, použijeme příkaz

```
svn revert <soubor>
```

kde <soubor> je jméno souboru, který chceme vrátit do původního stavu. Chceme-li do projektu přidat další projekt, použijeme příkaz

```
svn add <soubor>
```

a pokud jej chceme naopak odebrat, použijeme příkaz

```
svn delete <soubor>
```

Odeslání změn

Pokud v pracovní kopii provedeme nějaké změny, můžeme je do repozitory odeslat pomocí příkazu

```
svn commit -m "<popis zmeny>"
```

Před odesláním směn je nutné ověřit, že jde nová verze přeložit. Proto doporučuji následující postup:

```
svn update
maven test
svn commit -m "<popis zmen>"
```

Chceme-li před odesláním změn vidět, které soubory jsme změnili, použijeme příkaz

```
svn status
```

Chceme-li vidět i všechny provedené změny, použijeme příkaz

```
svn diff
```

Řešení konfliktů

Pokud dva vývojáři budou modifikovat stejný soubor naráz, dojde k takzvanému konfliktu. Pokud jsou změny na různém místě, subversion tento konflikt vyřeší automaticky.

Pokud jsou však na stejném místě, je nutné nejdříve provést update, pak konflikt vyřešit ručně a poté teprve provést odeslání změn do repozitory.

Další užitečné příkazy

- `svn info` - vypíše informace o pracovní kopii
- `svn log` - vypíše historii změn
- `svn status` - vypíše stav jednotlivých souborů v lokální kopii
- `svn copy` - zkopíruje soubor na jiné místo (tzv. lehká kopie)
- `svn move` - přesune soubor na jiné místo

Vytváření větví a značek

Pokud chceme vytvořit nějakou vývojovou větev, jednoduše zkopírujeme adresář trunk (příkazem `svn copy`) jako podadresář adresáře branches. Např.:

```
svn copy https://kore.fi.muni.cz/repos/fi/tools/2004/trunk \
https://kore.fi.muni.cz/repos/fi/tools/2004/branches/test1
```

Pokud chceme nějakou verzi označit, zkopírujeme adresář trunk do adresáře tags. Např.:

```
svn copy https://kore.fi.muni.cz/repos/fi/tools/2004/trunk \
https://kore.fi.muni.cz/repos/fi/tools/2004/tags/ver1.0
```

Základní zásady

- Do repozitory ukládáme vždy nějakou ucelenou změnu.
- Každou změnu smysluplně okomentujeme.
- Před uložením změny ověříme, že jde nová verze přeložit a že projde všemi testy.
- Do repozitory ukládáme pouze zdrojové kódy.