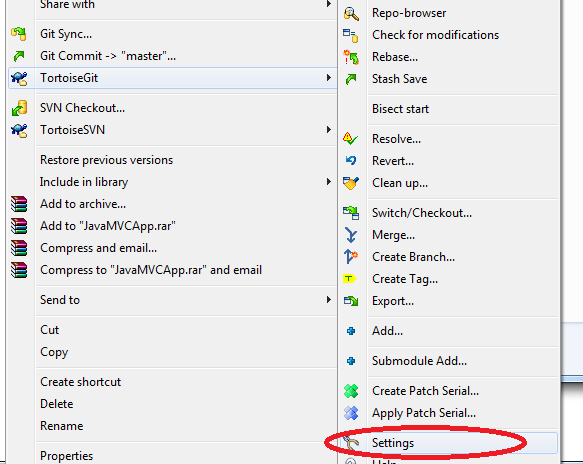
Java MVC Application

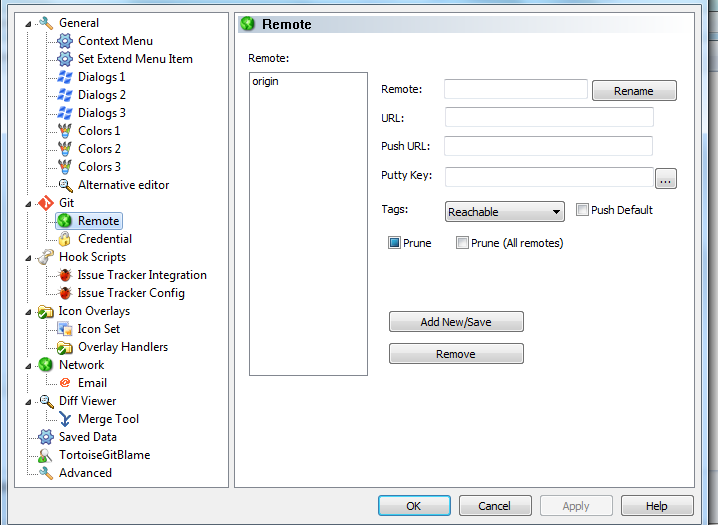
Part III Custom MVC Framework - Integrating with FRONTEND

1. Sincronizare fork de pe userul propriu cu repository-ul de start ( <https://github.com/ZeroToHero-2014/JavaMVCApp> )

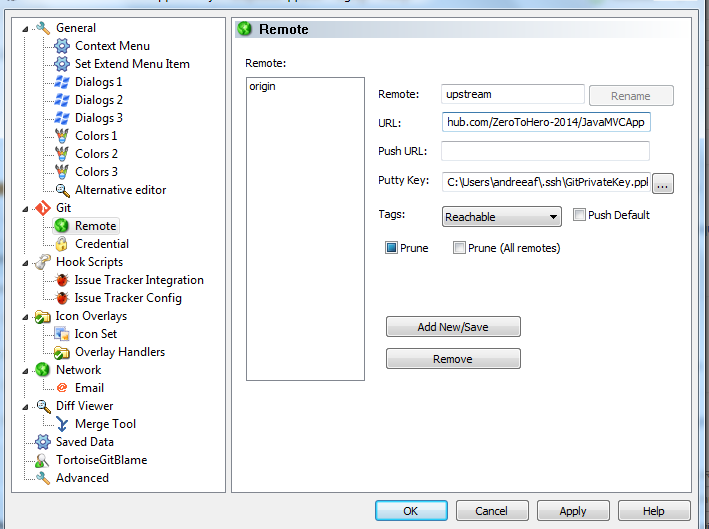
In structura de directoare de pe calculatorul vostru, navigati care clona fork-ului vostru. Click dreapta -> Tortoise Git. Alegeti optiunea *Settings*.



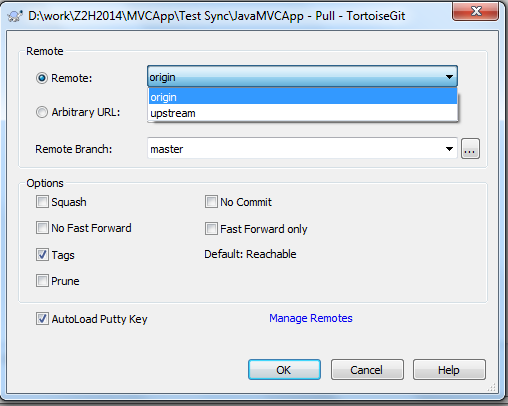
In feresatra *Settings*, alegeti *Remote*:



Adaugati un nou remote clonei voastre, cu numele *upstream*:



Folositi TortoiseGit pentru a face *Pull* pe clona fork-ului vostru, alegand remote-ul *upstream* (modificarile locale nu vor fi afectate; daca sunt conflicte, acestea vor trebui rezolvate):



1. In linie de comanda, navigati catre directorul *webapp* al fork-ului vostru (clona acestuia), acolo unde este prezent fisierul *bower.json*. Rulati comanda *bower install* . Care este scopul acestei comenzi?
2. Accesati link-ul <http://localhost:8080/app/#/employeeslist>. Realizati modificarile necesare astfel incat sa afiseze lista tuturor angajatilor din baza de date.
3. Accesati link-ul <http://localhost:8080/app/#/employeeview/100>. Realizati modificarile necesare astfel incat sa afiseze detaliile angajatului cu id-ul 100.
4. Adaugati metoda *deleteOneEmployee()* in clasa *EmployeeController.java*. Implementati metoda astfel incat sa stearga un angajat din baza de date. Id-ul angajatului este primit ca parametru. Respectati layerele aplicatiei (controller – service – dao) la implementarea functionalitatii.
5. Accesati link-ul <http://localhost:8080/app/#/employeedelete/1> . Realizati modificarile necesare astfel incat, prin pagina care raspunde la aceasta adresa, sa putem sterge un angajat.
6. Accesati link-ul <http://localhost:8080/app/#/employeeadd>. Realizati modificarile necesare astfel incat sa putem selecta in pagina informatiile necesare pentru salvarea unui angajat (Departament, Job si Manager). Implementati ApplicationControllere-le si celelalte metode necesare in partea de backend a aplicatiei.
7. La apasarea butonului *SAVE* din pagina de adaugare a unui angajat este necesara transmiterea informatiilor selectate in pagina catre server. Transmiteti informatiile in format JSON, pe request body. Ce metoda HTTP va fi folosita in acest caz?
8. In clasa *EmployeeController.java*, adaugati metoda *saveEmployee* care sa insereze un angajat.
9. Gasiti o modalitate prin care DispatcherServletul, in etapa de dispatch, poate transfera informatiile primite la punctul **8** catre metoda creata la punctul **9**.