

**3. lekcija: Sencha Architect**

**Sadržaj:**

Šta je Sencha Arhitect……………………………………………………………………………………………………………….3

Models………………………………………………………………………………………………………………………………………4

Stores………………………………………………………………………………………………………………………………………..5

Views……………………………………………………………………………………………………………………….…………….. 6

Controllers………………………………………………………………………………………………………………………………10

**Šta je Sencha Architect**

Sencha Architect je framework baziran na HTML5 tehnologiji koji se koristi za lakše pravljenje veb aplikacija.

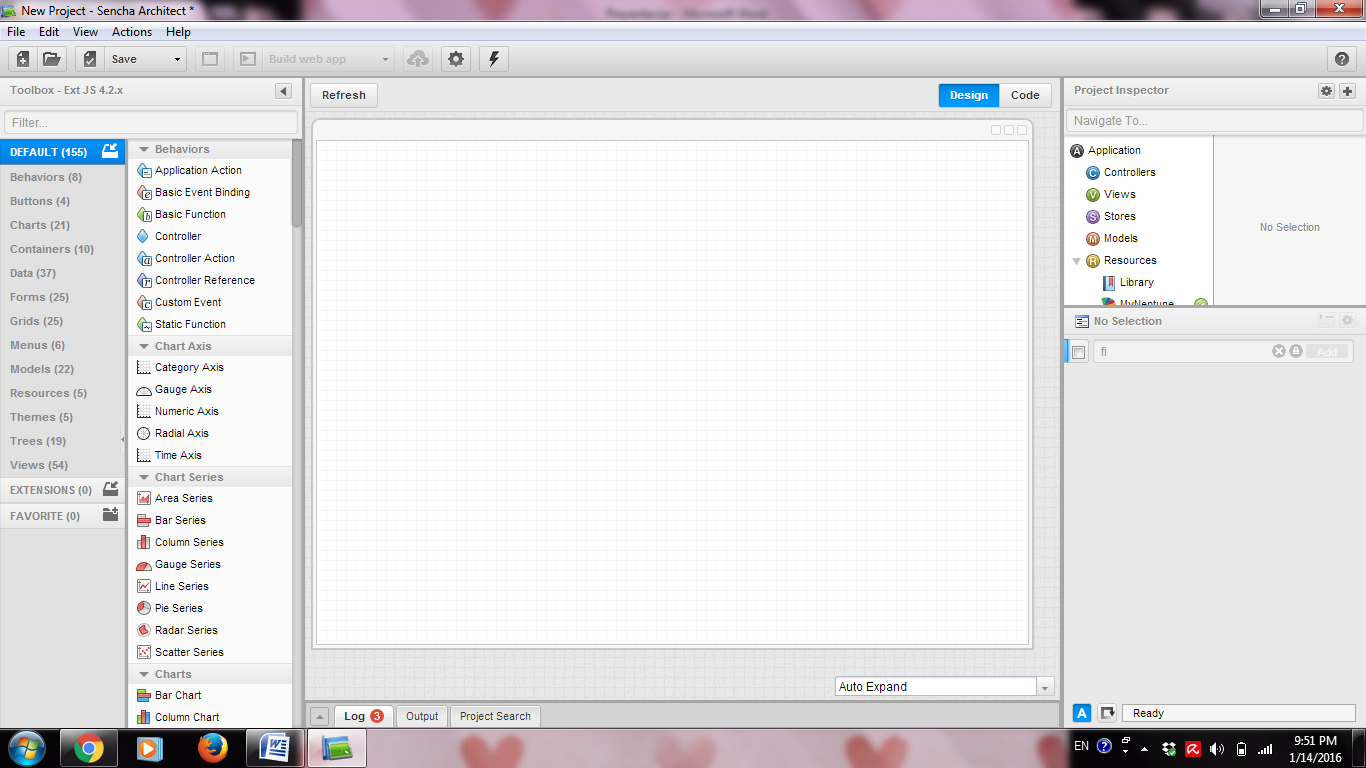
Sencha obiluje gotovim komponentama modernog dizajna i takođe autogeneriše veliki deo koda.

Sa leve strane u odeljku Toolbox se nalaze sve postojeće komponente, koje se mogu koristiti jednostavnim principom drag&drop.

+U sredini se nalazi trenutni izgled naše aplikacije, sa dve mogućnosti:

* Desing (pokazuje nam ceo interfejs aplikacije)
* Code (prikazuje ceo kod aplikacije)

Sa desne strane se nalaze Project Inspector i odeljak Config.

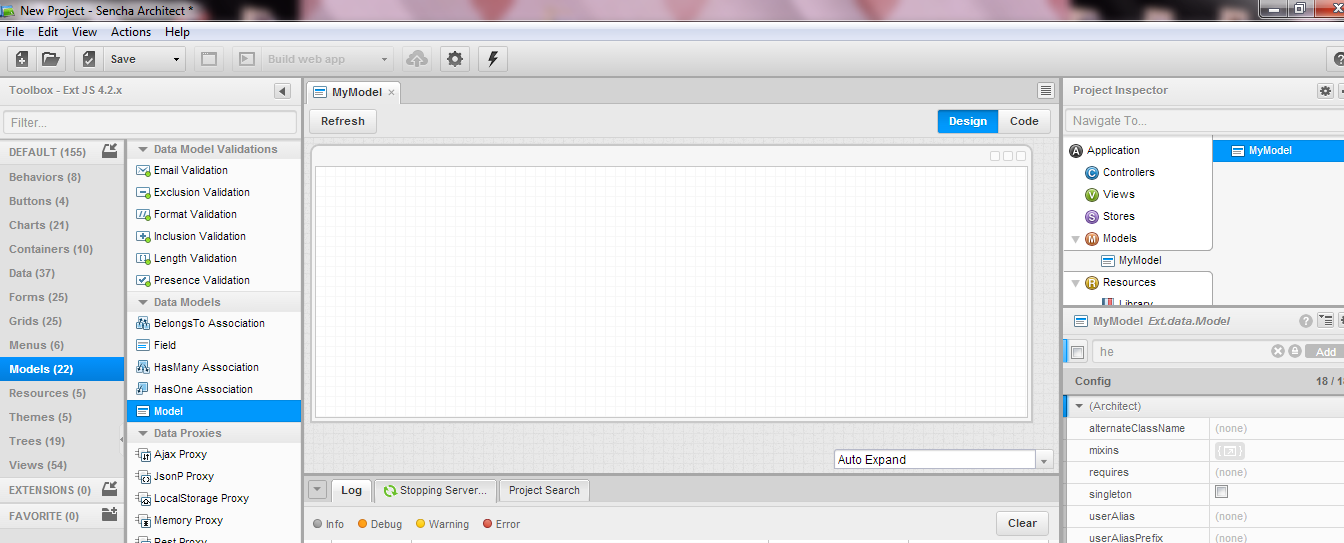
****

**Models**

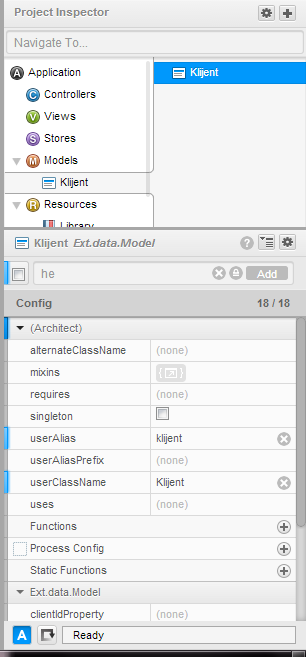
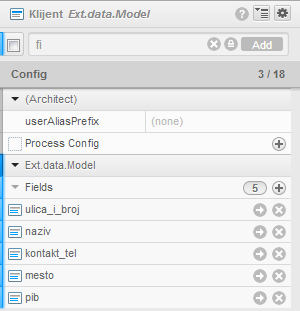
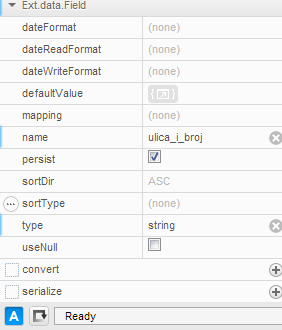
Iz Toolbox-a prevlačimo komponentu Model i ubacujemo je u Project Inspector u odeljak Models.

Najvažniji atributi u Modelu koje menjamo u odeljku Config su: userClassName, userAlias i Filds.

Pod atributom Filds dodajemo sva potrebna polja, tj. ova polja ce biti ekvivalentna kolonama tabele za koju pravimo interfejs (naravno tabela iz baze podataka).

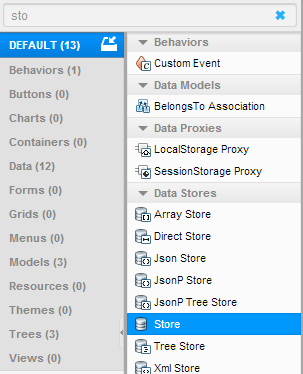
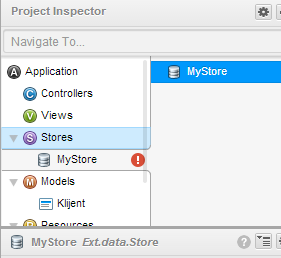
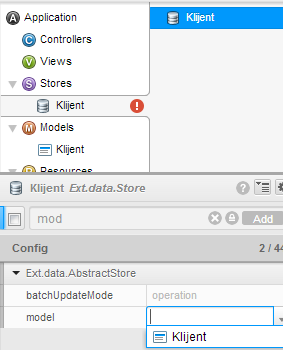
****

Za svako polje se mora navesti atribut Type, ekvivalentno koloni iz baze koju on oslikava. Ako je polje u bazi tipa String, i polje u modelu mora biti istog tipa String.

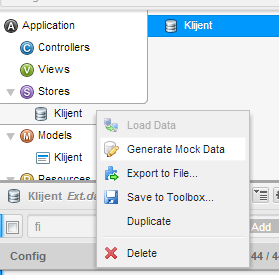
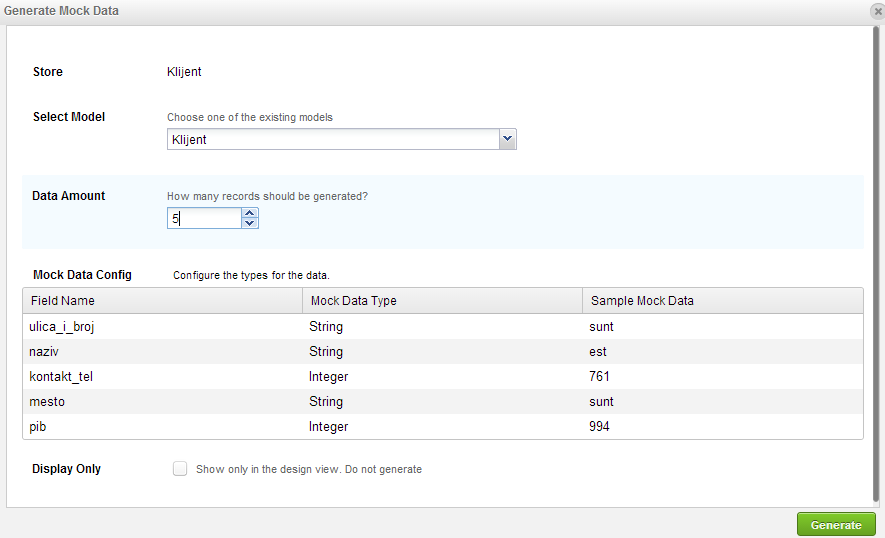
**  **

**Stores**

Istim principom “drag&drop” iz Toolbox-a prevlačimo komponentu Store u Project Inspector u odeljak Stores. U odeljku Config najbitniji atributi koji se trebaju promeniti za MyStore su: userClassName I model (povezujemo postojeći model).

**  **

Desnim klikom na klijent Stores biramo opciju Generate Mock Date. Ovo radimo da bi pri povezivanju sa gridom mogli proveriti da li je sve u redu, u smislu da li smo sve dobro povezali sa gridom i da li grid učitava postojeće podatke iz našeg stora.

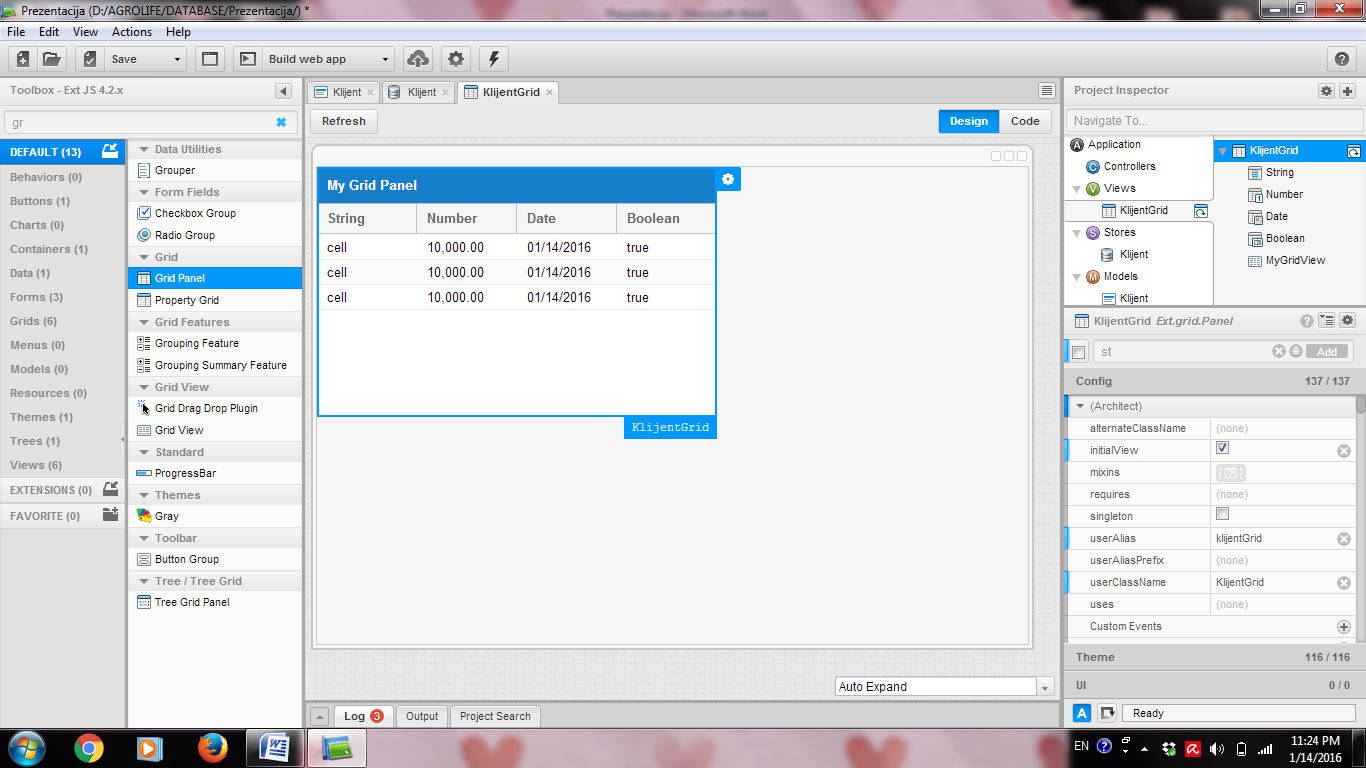
****

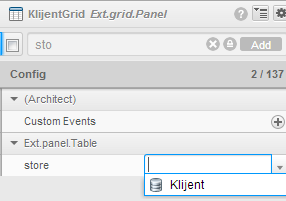
U prozoru koji nam je iskočio biramo po kojem modelu želimo da nam generiše podatke, zatim količinu podataka(u ovom primeru 5) i klikom na Generate završavamo.

**Views**

U Views odeljku Project Inspector smeštamo sve komponente koje korisnik može da vidi ( grid, form, toolbar, button itd).

Prevlačenjem komponente Grid Panel u Views i klikom na istu u Project Inspector najvažnije je promeniti sledeće attribute: userClassName (pr. KlijetnGrid), userAlias (pr. klijentGrid), Store (povezati sa postojećim).

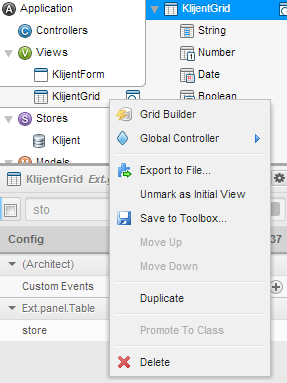
****

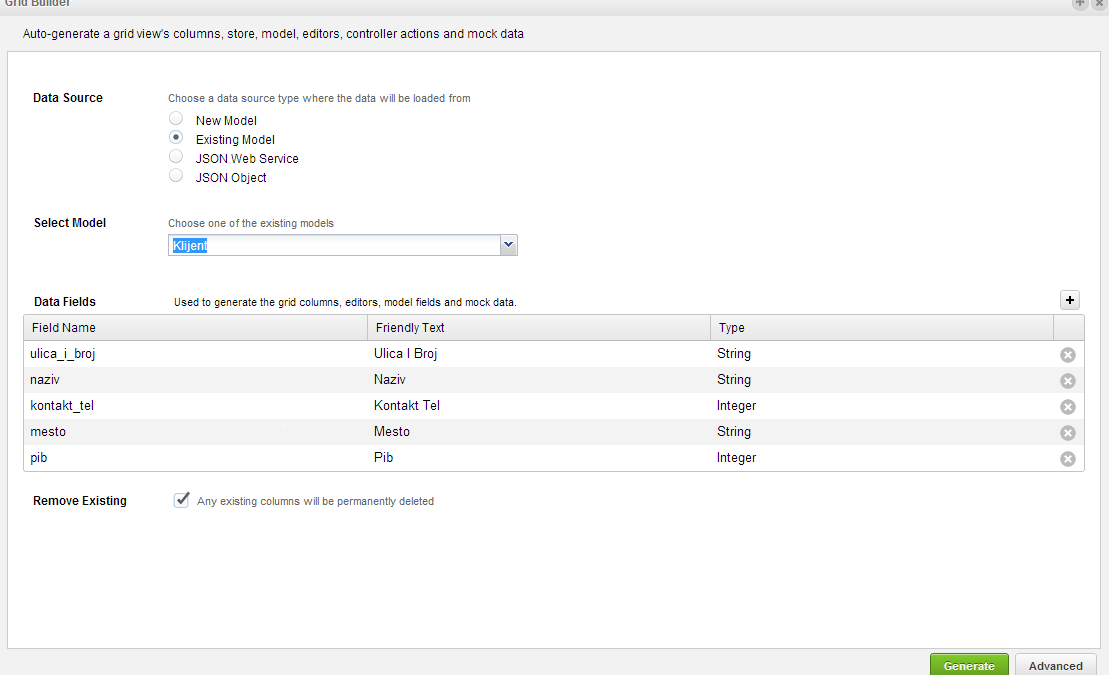
****

Desnim klikom na KlijentGrid odabirom opcije Grid Builder iskače prozor u kome menjamo sledece opcije.

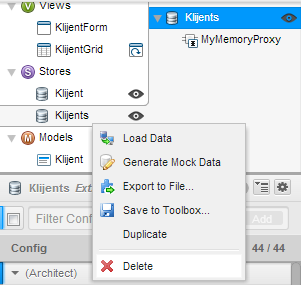
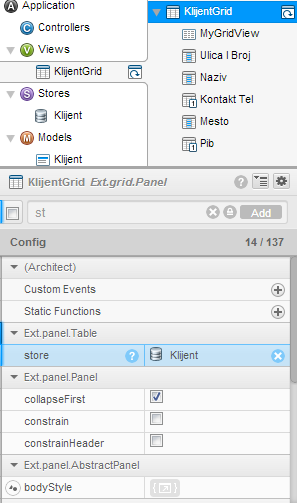
U Data Source štikliramo Existing Model (postojeći model), iz padajućeg menija biramo koji postojeći model želimo.

Pod poljem Remove Existing štikliramo Any existing columns will be permanently deleted, da bi nam obrisao sve neželjenje kolone koje već postoje u gridu.

****

****

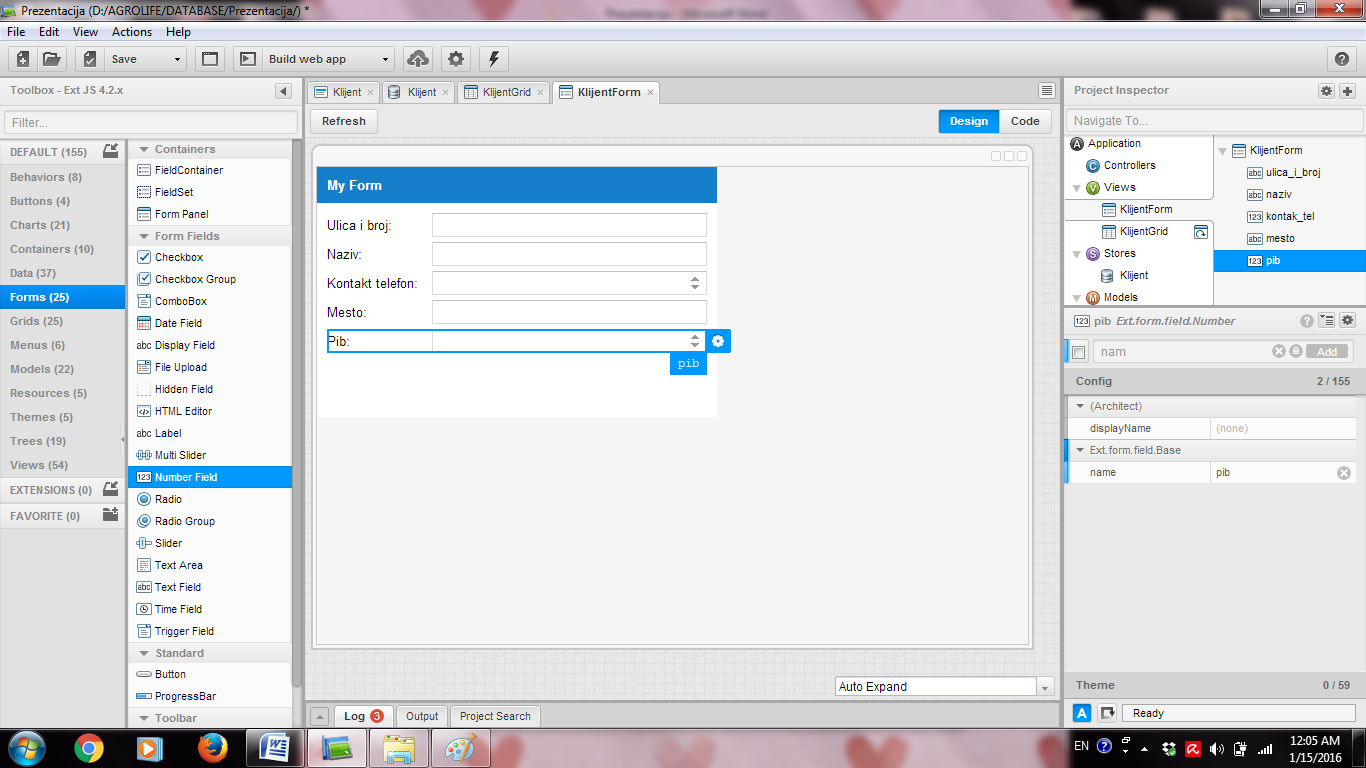
Grid Builder nam pravi poseban Stores koji treba izbrisati (Klijent**s**), i u Config pod atributom store povezati onaj koji smo mi pravili.

** **

Ponavljamo postupak za Form Panel, takodje u Config podešavamo userClassName i userAlias.

Iz Toolbox-a prevlačenjem ubacujemo potrebna polja (TextFild,NamrFild,Label,…).

Za svako polje je vazno podesiti atribu name, on se koristi da bi u kodu kontrolera mogao pronaći to polje.

****

**Controllers**

Prevlačenjem komponente Controler i klikom na nju u Project Inspectoru u odeljku Config treba promeniti sledeće atribute (userClassName i userAlias takođe treba dodeliti svakom kontroleru):

* **init**

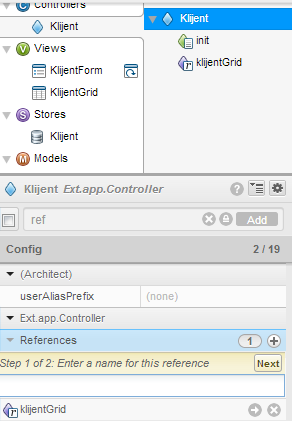
treba samo dodati na znak plus (init-u se kasnije ručno dodaje kod, on nam služi da dodamo slušaoce na dugmad)

* **References**

Takođe dodati svaku referencu na znak plus (reference dodajemo tako sto kada kliknemo na plus od nas traži da upišemo “ref”, tj.naziv reference, zatim selector )

Referencu koristimo da bi se znalo na kojoj komponenti ćemo primeniti kontroler, na primer za neki određeni grid, koristimo njegov userAlias kao selector.

Primer.



Ako želimo da se referenciramo na grid kojem smo dali userClassName (KllijentiGrid), i userAlias (klijentiGrid), za selector ćemo navesti klijentiGrid.

* **Functions**

Na znak plus dodajemo funkciju,koju ćemo preko slušaoca u init-u kasnije pozvati.

Klikom na plus, od nas traži da unesemo ime funkcije i da joj dodamo parametre.

Ime funkcije je jako važno,zato što je kasnije u init-u pozivamo preko imena.

