

## SEQUENZA DI PASSI PER AGGIUNGERE UNA NUOVA VIEW ALL'APP

1. Creare un nuovo file html nella cartella templates con il nome della view che desiderate (es. profilo.html).

NB i file html dei template non devono contenere tag html body o head. Questi saranno aggiunti dalla structureview o dal router durante il rendering.

2. Creare un nuovo file js con il nome della view in js/views/pages (es. ProfiloPageView.js) copiare in questo il contenuto del file MyView.js
3. Sostituire al nome della variabile MyView quella della vostra pagina... la variabile si trova in riga 7 e 52... alla riga 9 cambiare il nome del costruttore della view con lo stesso della variabile.
4. Dove si trova:

```
this.template = Utils.templates.submyview
```

sostituire submyview col il nome del vostro template

(es. that.template = Utils.templates.profilo)

E togliere le righe riguardanti il caricamento della sottoview:

```
var that = this;
```

```
instance.on("dataReady", function(){  
    // load the precompiled template if we have a data  
    that.template = Utils.templates.myview;  
    console.log("entro qui dentro");  
});
```

N.B. ovviamente dobbiamo cambiare la render di questo file in base alle nostre necessità.

5. ora aprire template.js

qui sono definite le template da associare alle view

```
define({  
    structure: "templates/structure.html",  
    myview: "templates/myview.html",  
    submyview: "templates/submyview/submyview.html"  
});
```

aggiungere la coppia chiave valore per il vostro template

ad es. visto che profilo si troverà in templates/profilo.html" aggiungo nella define :

```
define({
```

```
structure: "templates/structure.html",
myview: "templates/myview.html",
submyview: "templates/submyview/submyview.html",
profilo : "templates/profilo.html"
});
```

così il template sarà accessibile nella view tramite `Utils.templates.profilo`

#### 6. ora andare in `router.js`

nelle prime righe sono dichiarate come variabili le view javascript insieme ai model e alle variabili globali quali `backbone` etc.

```
var StructureView = require("views/StructureView");
var MyView = require("views/pages/MyView");
var MapView = require("views/pages/MapView")
```

qui aggiungere:

```
var ProfiloPageView = require("views/pages/ProfiloPageView");
```

poi andare nelle routes

```
routes: {
  // the default is the structure view
  "": "home",
  "myview": "myView",
  "map": "map"
},
```

ed aggiungere la route per la vostra pagina

```
routes: {
  // the default is the structure view
  "": "home",
  "myview": "myView",
  "map": "map",
  "profilo" : "showprofilo"
},
```

ora aggiungere prima della funzione showStructure (l'ultima) una nuova funzione dichiarata come segue

```
showprofilo: function () {
```

```
    var page = new ProfiloPageView();
```

```
    this.changePage(page);
```

```
},
```

il nome ProfiloPageView() è di esempio varia a seconda di come viene dichiarata la variabile in cima a router.js

ora se lanciate l'app e sull'url aggiungete alla fine #profilo vi troverete sulla pagina profilo