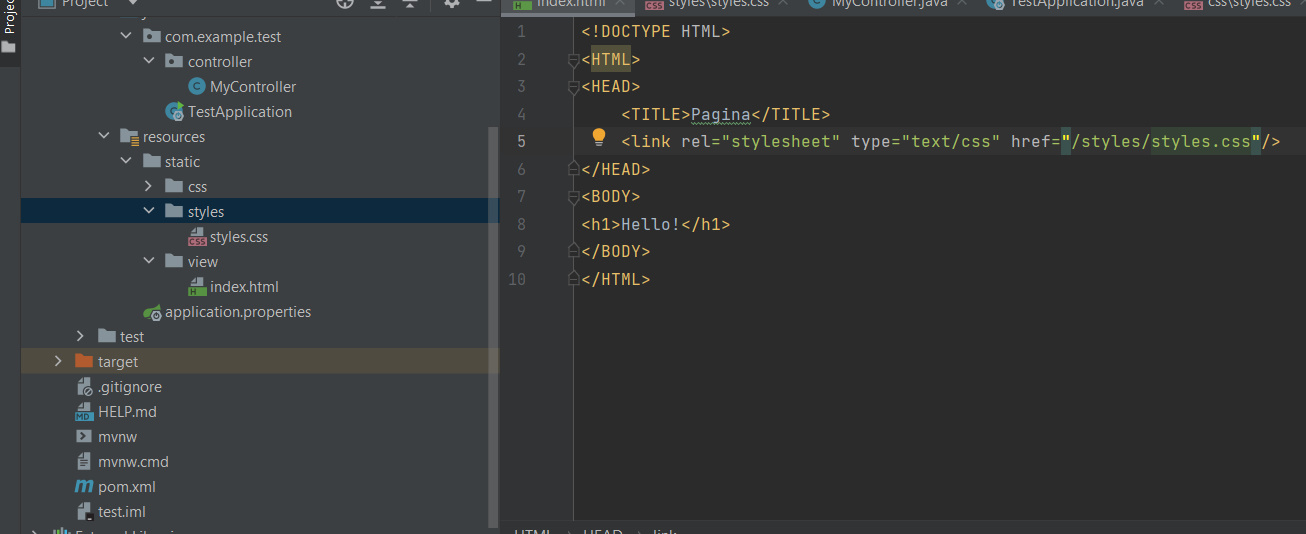
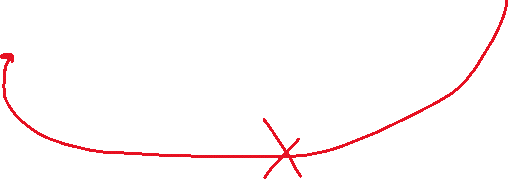
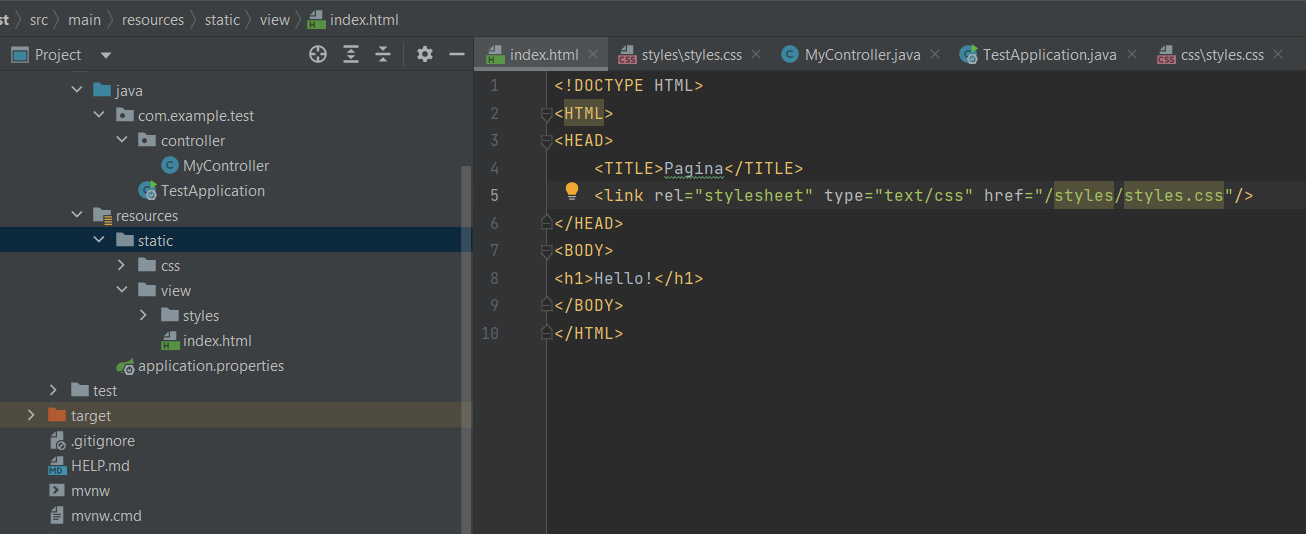
**Add CSS and other**

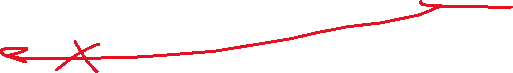
* In Spring Boot e mai usor de a adauga pagini CSS.
* Paginile web, in mod default au acces la folderul /static din /resources, si deci orice fisier sau folder scris in href=”” va fi cautat automat **doar** in /resources/static/





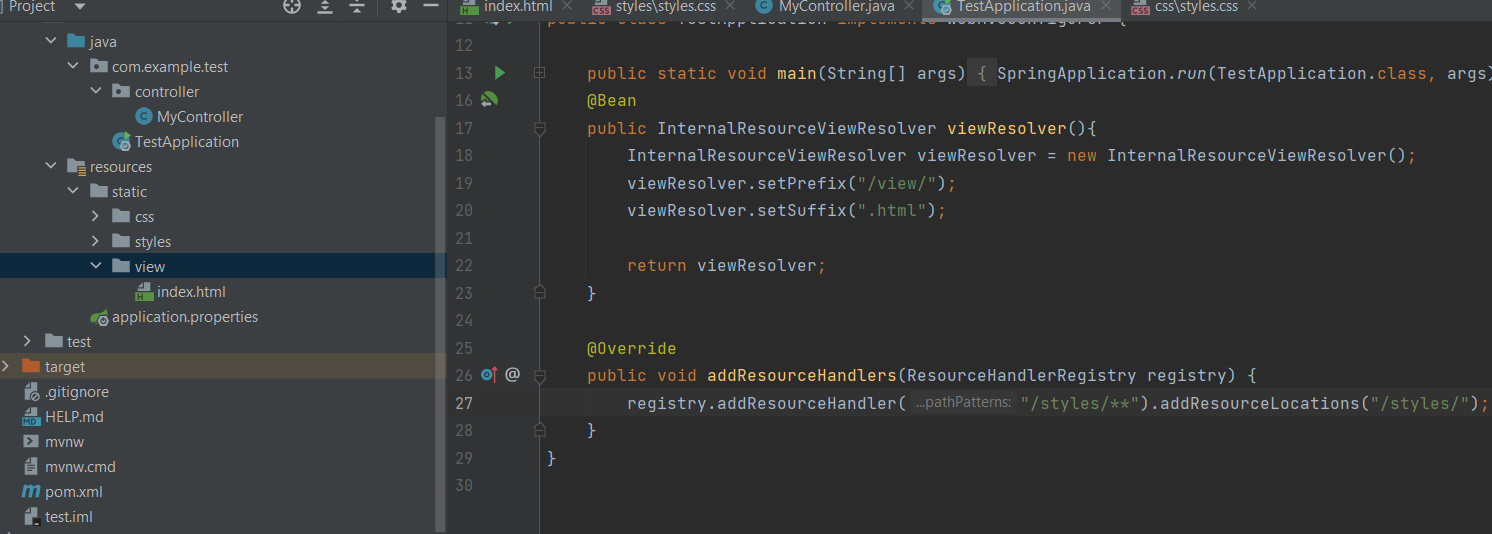
* **Fisierele mereu vor fi cautate in /static/, nu in folderul in care se afla pagina web!Spre deosebire de cum era cu Spring simplu, iata de ce asa ceva nu va merge:**





* **Atentie cum suprascriem metoda addResourceHandlers pentru situatia de sus:**
* public void addResourceHandlers(ResourceHandlerRegistry registry) {  
   registry.addResourceHandler("/styles/\*\*").addResourceLocations("classpath:/static/view/styles/");  
  }

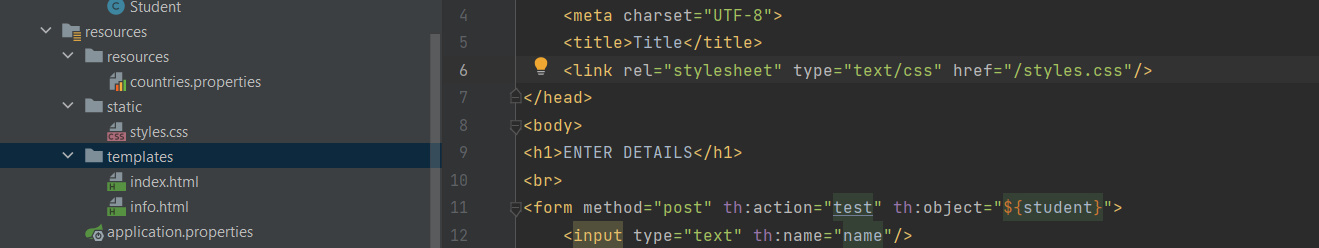
**Deci, nu uitam ca metoda data are nevoie de fullpath, caci ea se va ocupa de gasirea si trimiterea fisierului, nu va inlocui pur si simplu /styles/ cu acel link pus de noi. Dar, pentru asta, e nevoie de classpath: ca sa stie ca ne referim anume la resources din classpath**





**deci, /styles/ nu va fi pur si simplu inlocuit cu /styles/, ci se va prelua requestul si va fi cautat in dependenta de ce scriem acolo, dar noi nu am dat fullpath si spring boot nu stie unde sa caute /styles/, caci doar pagina web cauta totul in /resources/static/**

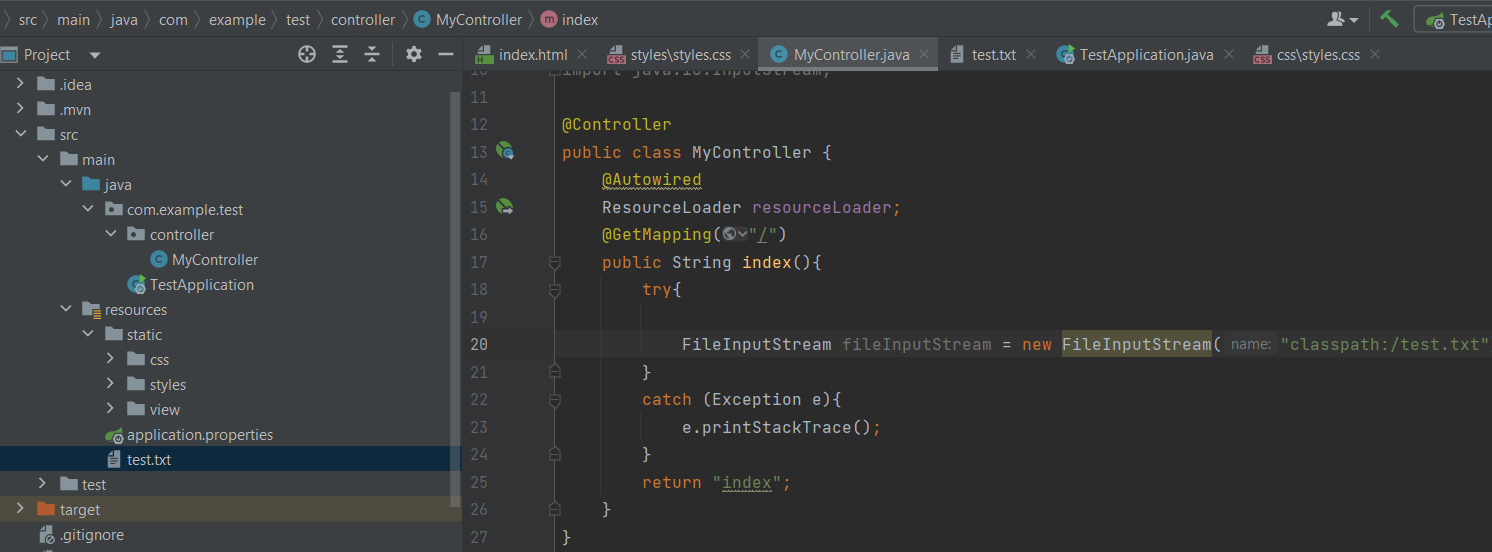
**Chiar daca folosim thymeleaf, unde paginile .HTML sun stocate in templates, orice scriem cu href=”” oricum mereu va fi cautat in /static!!!Doar th:href va face ca ele sa fie cautate in resources si public si MET-INF/resources si static automat**

****



**Read direct from Code**

* Anotatiile automat cauta resursele in resources, dar nu si codul Java
* De ex, fisierele ce le deschidem cu cod nu vor fi cautate automat in resources asa cum era in Spring normal! Nu asa merg chestiile intr-un jar.





* Pentru a rezolva problema data, ne trebuie sa injectam un obiect de tip **ResourceLoader**, pe care Spring il creaza automat ca sa putem accesa direct in Cod resursele din /resources/
* Apoi cream un obiect de tip **Resource** pentru a obtine o resursa completa cu .getResource, si folosim .getFile pentru a obtine fisierul:
* @Controller  
  public class MyController {  
   @Autowired  
   ResourceLoader resourceLoader;  
   @GetMapping("/")  
   public String index(){  
   try{  
   Resource resource = resourceLoader.getResource("classpath:/test.txt");  
   FileInputStream fileInputStream = new FileInputStream(resource.getFile());  
   }  
   catch (Exception e){  
   e.printStackTrace();  
   }  
   return "index";  
   }  
  }
* **Atentie!classpath: e obligatoriu!**