@[Entity|Document]  
public class A {  
 @Id [@GeneratedValue(strategy=GenerationType.$)] private long $;  
 @Temporal(TemporalType.$) private Date $  
 @OneToMany(cascade = CascadeType.$, mappedBy=”attribuut\_B”, fetch = FetchType.$)  
 private List<T> $  
 public $ (…){} <- lege constructor VERPLICHT  
 }  
@[Entity|Document]  
public class B {  
 @ManyToOne(fetch = FetchType.$, optional = [true|false])  
 @JoinColumn(name = “A\_id”, nullable = [true|false])  
 @OnDelete(action = OnDeleteAction.$)  
 A a;  
}  
public interface $ extends [MongoRepository|CrudRepository]<T, V>{  
 @Query(“SELECT x from y x WHERE x.attribuut = :parameter”)  
 [List|Optional]<Klasse> $ (@Param(“parameter”) [int|String|…] parameter);  
 // Bij enum heel het pad vermelden: package.enumklasse.keuze  
}  
@Component  
public class $Service {  
 @Autowired private Repository repository;  
 public $ $ { return repository.$ }  
}  
@RestController  
public class $Controller {  
 @Autowired private $Service service;  
 @RequestMapping(value=”/path”, method=RequestMethod.$)  
 1) public $ $ () -> GET /path  
 2) public $ $(@PathVariable(“abc”) $ $) -> /path/{abc} (ook aanpassen in value)  
 3) public $ $ (@RequestBody Object o) -> verwacht JSON of gelijkaardig  
 4) public $ $ (@RequestParam(“$”) $ $) -> form-encoded data  
 {  
 // aanspreken service  
 }  
  
}