Najdôležitejšie JPA anotácie

Anotácia	Poznámka
@Entity	anotácia entitnej triedy
@Table(name = "OSOBA")	nepovinný, vhodný ak chceme sami definovať meno tabuľky entitnej triedy
@Id	
@Column(name = "ID")	nepovinný, vhodný ak chceme sami definovať meno stĺpca
@GeneratedValue(strategy = GenerationType.AUTO)	potrebný len pre tabuľky s autogenerovaným kľúčom
@Temporal(TemporalType.DATE)	potrebný pre dátové členy typu Date
@ElementCollection	nepovinný, vhodný pre viacnásobné dátové členy základných dátových typov
@ManyToMany @ManyToOne @OneToMany	
@JoinColumn(name="vydavatel_fk")	vhodný spolu s @OneToMany pre definovanie mena FK stĺpca
@JoinTable(name = "BOOKAUTHOR" ,joinColumns = @JoinColumn(name = "book_fk"), inverseJoinColumns = @JoinColumn(name = "author_fk"))	nepovinný, definuje prepojovaciu tabuľku pri @ManyToMany asociácii
@OneToMany(fetch = FetchType.LAZY)	parameter **fetch** nie je povinný. Môže byť použitý aj s ostatnými typmi anotácií asociácií
@OneToMany(cascade=CascadeType.PERSIST)	parameter **cascade** nie je povinný. Môže byť použitý aj s ostatnými typmi anotácií asociácií
@OneToMany(orphanRemoval=true)	parameter **orphanRemoval** nie je povinný. Umožňuje automatické odstraňovanie osirotených detských komponent v kompozícii
@OneToMany(mappedBy = "vydavatelId")	parameter **mappedBy** označuje atribút na opačnej strane obojstrannej asocoácie. Môže byť použitý aj s ostatnými typmi anotácií asociácií
@Embeddable	anotácia embedovanej triedy
@Embedded	anotácia objektu embedovanej triedy
@EmbeddedId	anotácia zloženého kľúča
@MapsId("empid")	anotácia umožňuje definovať, ktorá položka zloženého kľúča je cudzí kľúč asociácie ManyToOne
@SecondaryTable(name="T_LOGIN")	anotácia pre definiciu sekundárnej tabuľlky
@Column(table= "T_LOGIN", column="heslo")	anotácia atribútu ukladaného v sekundárnej tabuľke
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)	definovanie strategie dedičnosti, anotuje sa koreňová trieda hierarchie entitných tried
@MappedSuperclass	anotácia abstraktnej triedy, ktorá slúži ako koreň hierarchie entitných tried

Vytvorenie entity managera

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("vsaPU");
EntityManager em = emf.createEntityManager();
```

Najdôležitejšie anotácie pre REST

Anotácia	Poznámka
@Path("menu")	relativna URL resoursu
@Path("{pid: [A-Z]{3}.*}")	relativna parametrizovatelna URL resoursu
getText(@PathParam("pid") String id)	a extrakcia parametrizovanej casti URL do argumentu metody
getParName(@QueryParam("meno") String s)	a extrakcia parametrov dotazu
@Consumes(MediaType .APPLICATION_XML)	media type dat, ktore prijima servis od klientov
@Produce(MediaType .APPLICATION_XML)	media type dat, ktore posiela servis klientom
@GET	metoda vyvolana HTTP GET requestom
@PUT	metoda vyvolana HTTP PUT requestom
@POST	metoda vyvolana HTTP POST requestom
@DELETE	metoda vyvolana HTTP DELETE requestom

Najdôležitejšie JAXB anotácie

Anotácia	Poznámka
@XmlRootElement(name = "book")	anotuje triedu, ktorú chceme serializovať do XML. Parameter name umožňuje definovať meno koreňového elementu
@XmlElement(name = "title")	nepovinná anotácia dátoveho člena triedy. Parameter name umožňuje definovať meno element.
@XmlElementWrapper(name = "ponuka")	nepovinná, umožnuje definovať element, ktorý “obaluje” kolekciu elementov viacnásobného dátového člena.
@XmlTransient	nepovinná, umožňuje potlačiť serializáciu dátového člena

Rôzne ďalšie anotácie

Anotácia	Poznámka
@Singleton	**Pozor!** javax.inject.Singleton
@RequestScoped	
@Context UriInfo uriInfo @Context ServletContext ctx	

Vytvorenie a použitie REST klienta

```
Client client = ClientBuilder.newClient();
WebTarget webTarget = client.target(BASE_URI).path("message");
```

Odoslanie požiadavky PUT

```
webTarget.request(MediaType.TEXT_PLAIN)
         .put(Entity.entity("Hello", MediaType.TEXT_PLAIN));
```

Odoslanie požiadavky GET

```
String result = webTarget.request(MediaType.TEXT_PLAIN).get(String.class);
```