atitit golan java php lan compare.docx jpa

|  |  |  |
| --- | --- | --- |
|  | Golan | java |
| Var define | sql := "drop ” | Var sql=”” |
| If else | **if** err != nil {  } | If (){} |
| foreach | **for** i := 0; i <= 9999; i++ {    } | Foreach() |
| 异常处理 | **函数作用域，for循环里使用需要使用匿名函数包起来**  **func** insertOne( ) {  // 处理异常的函数 same befFunExit()  **defer** **func**() {｝  inTry() | Try catch 块作用于 |
| Db query | //查询操作  **func** Query(db \*sql.DB) {  rows, \_ := db.Query(sql)    **for** rows.Next() {  **var** name string  rows.Scan(&name)  fmt.Println(name)  //fmt.Println(json.Marshal()  }  } | Map properties = Maps.*newLinkedHashMap*();  properties.put("javax.persistence.jdbc.driver", "com.mysql.cj.jdbc.Driver");    properties.put("openjpa.RuntimeUnenhancedClasses", RuntimeUnenhancedClassesModes.***SUPPORTED***); // if needed  EntityManagerFactory factory = Persistence.*createEntityManagerFactory*("wmsPersisteUnitName", properties);  EntityManager em = factory.createEntityManager();  Query q = em.createNativeQuery("select 1 as age,'ati' name;select 2;", Map.**class**);  List<Map> li = q.getResultList();  System.***out***.println(JSON.*toJSONString*(li)); |
| Db update | res, \_ := db.Exec(sql)  // CheckErr(err)  //查询删除多少条信息  // if(res)  // {  num, \_ := res.RowsAffected()  // CheckErr(err)  fmt.Println(num) //1 if creted d | EntityManager em = factory.createEntityManager();  EntityTransaction transaction = em.getTransaction();  transaction.begin();    System.out.println(sql);  Query createNativeQuery = em.createNativeQuery(sql);  int executeUpdate = createNativeQuery.executeUpdate();  System.out.println(executeUpdate);  transaction.commit(); |
| Rest接口 | http.HandleFunc("/a", **func**(w http.ResponseWriter, r \*http.Request) {  fmt.Fprintf(w, "Hello, aaaaa", html.EscapeString(r.URL.Path))  })  log.Fatal(http.ListenAndServe(":8080", nil)) | @RestController  @EnableAutoConfiguration  **public** **class** RestApplication {  **public** **static** **void** main(String[] args) {  SpringApplication.*run*(RestApplication.**class**, args);  }  @RequestMapping("/")  String home() {  **return** "Hello World!";  } |
|  |  |  |