#### Intellij IDEA

Jet Brains Account: cosyu & cos3900223

Community version is free and for JVM and Android development while Ultimate version (fee-charge and need license) is for web and enterprise development.

##### Create Gradle Project with Spring

For web development, it creates Gradle project with SpringBoot usually, after Gradle project created, IDE will download related libraries and Tomcat (bundled with Ultimate version) will be started automatically with port 8080. Developers can create REST Controller and run the SpringBoot Application which is with Main method, then open <http://localhost:8080> with browser to test the server is started or not.

Related downloaded libraries will be shown like this:

Text

Description automatically generated

Gradle structure on the right side will be shown like this:

Graphical user interface, text

Description automatically generated

##### Change JDK

Open File/Project Structure/Project Settings/Project/SDK

##### Install Plugin

open File/Settings/Project/Plugins

##### Add Application Server

open File/Settings/Application Server

##### WAR/JAR File

Upon creating the web application with Spring, war is for web application deployment while jar is for running with java command.

Upon building web project successfully, developer can find the war/jar file in output folder of corresponding project, like this

Graphical user interface, text, application

Description automatically generated

WAR/JAR file only contains the class files & configure files in src/main folder. Others are setting/files for the IDE.

Property/XML file in JAR is allowed to modify which WAR does not allow to change

For JAR is created by Spring Boot, it is a standalone programme which is bundled with Application Server (i.e. Tomcat), it can access <http://localhost:8080/> after the JAR is run.

##### Build the project to local Maven repository

Go to project directory with cmd, run following gradle command(need to install gradle and set path first in system environment first)

gradle --refresh-dependencies clean build publishToMavenLocal

If it is success, it will build jar in local repository so that it can be used by other projects with implementation(“…”)

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

##### Gradle

Gradle is bundled with Intellij, it’s default gradle user home is C:\Users\xxxx\.gradle, dependencies for project will be downloaded and saved in C:\Users\xxxx\.gradle\caches, gradle user home can be changed by adding GRADLE\_USER\_HOME path in system environment

Gralde will use following script to determine which repository will be retrieved for dependencies download

repositories **{**

**//** Use local maven repository first, and it will download dependencies from central maven repository if cannot find them

mavenLocal()mavenCentral()

//Only download dependencies from central maven repository if they had not downloaded before

mavenCentral()  
**}**

##### Project Configure File

XML or property file is saved in main/resources and it will be bundled with generated war/jar file in WEB-INF\classes path

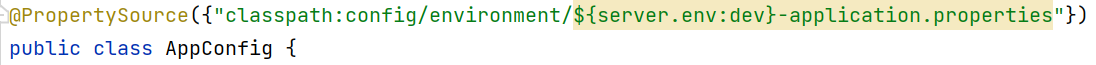
Graphical user interface, application

Description automatically generated

Graphical user interface, table

Description automatically generated with medium confidence

Programme will read the property file like this, classpath mean \WEB-INF\classes\ in jar/war file



Property file can be parameter instead of particular file path, it will be shown like this, it needs to pass parameter for programme running. Generally speaking, it will use this approach to read external(not bundled with the JAR/WAR file) configure files for web application.

@PropertySources({  
 @PropertySource("file:${common.properties}"),  
 @PropertySource("file:${secret.properties}")  
})

##### Generate Entity by Intellj(Ultimate)

Required plugin: *Java EE: Persistence (JPA)* (bundled)

Add compileOnly('javax.persistence:javax.persistence-api:2.2') to build.gradle file to enable JPA support for project

Open DataBase table on the right side, select Data Source / your data base type, fill in the database connection information

Graphical user interface, application

Description automatically generated

Open View/Tools Window/Persistence to open Persistence

<https://www.jetbrains.com/help/idea/persistence-tool-window.html>

##### Useful Short Key

Double click Shift, to search files in the whole project

Ctrl+Shift+F, to search content in the whole project

Ctrl+Alt+S, to open setting dialog

Alt+Enter, to import class

Ctrl+Alt+O, to optimize imports

Alt+Shirt+F9, to build and debug main method

##### Add Argument for Main Method

Alt+Shift+F9 and selection configuration, input parameters (split by space) in Programme Arguments

Graphical user interface, text, application, email

Description automatically generated

##### Add Environment Variables for Project

Alt+Shift+F9 and selection configuration, input environment variables (split by ;), this is for variable configure file such as @PropertySource(**"file:${common.properties}"**)

The var name is common.properties, usually to read external configure files which is not bundled with programme.

Graphical user interface, text, application, email

Description automatically generated

##### Add JAM Argument for Project

Alt+Shift+F9 and selection configuration, click Modify options/Add VM options, input highlight parameter, it will change the arguments when the JVM starts

Graphical user interface, text, application, email

Description automatically generated

##### Add JAM Argument for build.gradle

bootRun **{** jvmArgs =  
 [  
 **"-Dcommon.properties=..\\CEDB-TSW-common-library\\config\\common.properties"**,  
 **"-Dsecret.properties=..\\CEDB-TSW-common-library\\config\\secret.properties"** ]  
**}**

##### .gradle folder & .idea folder

It is created by Intellij IDEA when the project is open, this folder should not be committed to Git

##### .gradle folder & .idea folder

It is created by Intellij IDEA when the project is build, this folder should not be committed to Git

##### Save console log to specify file

Shift+Alt+F9 to open Debut dialog, click edit configuration, select Main Application tag, click Modify options, click Logs/Save console to out file. The specify file will be override if Spring Application restarts, otherwise, it will append log.

It is very useful for debug.

##### Boot Jar & Jar & War

Boot Jar is executable jar and Jar is library/executable jar.

A JAR file extension is .jar and is created with jar command from command prompt (like javac command is executed). Generally, a JAR file contains Java related resources like libraries, classes etc.JAR file is like winzip file except that Jar files are platform independent.

A WAR file is simply a JAR file but contains only Web related Java files like Servlets, JSP, HTML.

To execute a WAR file, a Web server or Web container is required, for example, Tomcat or Weblogic or Websphere. To execute a JAR file, simple JDK is enough.

For a library, build command is same as jar command in Intellij, bootJar command is not working

For a gradle java application with main method, running build command will generate executable jar and library jar, running jar command only generate library jar, running bootJar command only generate executable jar

To generate war file, it needs to add war plugin, then run bootWar command to generate executable war file, run war command to generate war file

Graphical user interface, text, application, email

Description automatically generated

#### Spring

##### Ioc/Dependency Injection

JavaBean (e.g. Service, Repository) is created and managed by Spring container (i.e. ApplicationContext/BeanFactory) instead of application. It uses @Autowired Annotation to inject JavaBean.

##### @Component

To define a class as JavaBean which is created and managed by Spring container

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/12317/#outline__2_2_3>

##### @ComponentScan

To scan all class to be JavaBean in specify package

##### @Bean

To define a method will return a JavaBean which is created and managed by Spring container

<https://matthung0807.blogspot.com/2019/05/spring-bean.html>

##### @Qualifier

When you need more control of the dependency injection process,

@Qualifier can be used. @Qualifier can be specified on individual constructor arguments or method parameters. This annotation is used to avoid confusion which occurs when you create more than one bean of the same type and want to wire only one of them with a property

<https://springframework.guru/spring-framework-annotations/>

##### @Primary

To tell Spring which bean will be injected first

<https://www.baeldung.com/spring-primary>

##### @Rescource

It is not Spring’s annotation (needs to import javax.annotation.Resource) and similar with @Autowire

##### @PropertySource

To specify property/configure file to load

@Configuration

@PropertySource("calsspath:application.properties")

public class AppConfig {

@Value("${app.zone:Z}")

String zoneId;

}

It can be multiple

@PropertySources({  
 @PropertySource("classpath:common.properties"),  
 @PropertySource("classpath:secret.properties")  
})

##### @Value

Once there is class reading property file by using @PropertySource (e.g. AppConfigure.class), other classes can inject values of property file by using @Value

##### @Scope

To define a bean’s scope, there are total 5 values.

Singleton: default value, it will create only one object by Spring

Prototype: it will create new bean every time

Request: it will create new bean for each http request

Session: it will create new bean for each http session

GlobalSession: it will create new bean for global http session.

<https://blog.csdn.net/jacklearntech/article/details/40157861?spm=1001.2101.3001.6650.6&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-6.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-6.pc_relevant_default&utm_relevant_index=13>

##### @Transactional

<https://iter01.com/61414.html>

By default, it can retrieve data which has not committed by transaction to DB

##### @Transactional for Class & @Transactional for Method

Spring applies the class-level annotation to all public methods of this class that we did not annotate with @Transactional. However, if we put the annotation on a private or protected method, Spring will ignore it without an error.

<https://stackoverflow.com/questions/23132822/what-is-the-difference-between-defining-transactional-on-class-vs-method>

All DML/DDL in SQL needs transaction except select, otherwise, it will throw TransactionRequiredException

##### @RequestParameter & @RequestBody & @ResponseBody

@RequestParameter is to receive parameters from request header for application/x-www-form-urlencoded content type

@RequestBody is to receive parameters from request body for application/json or application/xml content type, client side submit parameters with json format and they will be parsed in backend.

@RequestBody returns ResponseEntity<?> for controller method

<https://kknews.cc/code/jkrn93l.html>

Demo (include file upload):



##### @Async

It is used to indicated to method is async method, the class must be created and maintained by Spring otherwise, the async method will not work.

Caller cannot try catch the exception threw by async method which is without return value, it needs class which implements AsyncUncaughtExceptionHandler, for the async method with return value, try catch the future.get() to catch the exception which is thrown by async method in the main thread. @Transaction will not work for method with @Async



@Aysnc with @Transactional

If the @Async annotation is being used extra care should be taken with respect to transaction. When a @Transactional Spring @Component calls a method annotated with @Async the call to the asynchronous method is being scheduled and executed at a later time by a task executor and is thus handled as a 'fresh' call, i.e. without a transactional context. If the @Async method (or the @Component in which it is declared) is not @Transactional by itself Spring will not manage any needed transactions.

In order to make Spring manage the transaction of the @Async method either the @Component or the method itself should declare the @Transactional annotation, this way Spring will manage the transaction even if a method is being executed asynchronous.

##### @JsonProperty

It is used to rename fields for serializable or deserialize

<https://blog.csdn.net/liliang_11676/article/details/80210065>

@JsonProperty("stuName")  
private String name;

##### @JsonIgnoreProperties

It is used to specify fields will be ignored for serializable or deserialize, same as @JsonIgnore

@JsonIgnoreProperties({ "age"})//it means specify fields will be ignored for serializable or deserialize, same as @JsonIgnore  
public class Student {

##### @JsonFormat

It is used to specify date time format

@JsonFormat(pattern="MM-dd-yyyy HH:mm:ss")//specify the date time format  
private Date updateTime;

##### @JsonInclude

It is used to specify filed will NOT be included for serializable or deserialize if value is blank

@JsonInclude(JsonInclude.Include.*NON\_NULL*)//this filed will NOT be included for serializable or deserialize if value is blank  
private String hobby;

##### @JsonSerialize && @JsonDeserialize

It is used to embed customer logic for serialize and deserialize

@JsonSerialize(using = GenderSerialize.class)  
@JsonDeserialize(using = GenderDeserialize.class)  
private int gender;

public class GenderSerialize extends JsonSerializer<Integer> {  
  
 @Override  
 public void serialize(Integer value, JsonGenerator gen, SerializerProvider serializers) throws IOException {  
  
 if(value == 1){  
 gen.writeString("M");  
 }else{  
 gen.writeString("F");  
 }  
 }  
}

public class GenderDeserialize extends JsonDeserializer<Integer> {  
  
 @Override  
 public Integer deserialize(JsonParser p, DeserializationContext ctxt) throws IOException, JsonProcessingException {  
  
 if("M".equals(p.getText())){  
 return Integer.*valueOf*(1);  
 }else{  
 return Integer.*valueOf*(0);  
 }  
  
 }  
}

##### @Column

It is used to map the field to corresponding DB field

@Column(name = "USER\_ID",nullable = false)  
private long userId;

It can be optional if the field is using Java var name standard, i.e. userId -> USER\_ID

##### @OneToMany && @ManyToMany

Case 1: One organization has multiple external users

Organisation:  
@OneToMany(mappedBy = "organisation")  
private List<ExternalUser> employees = new ArrayList<>();  
  
ExternalUser(include ORGANISATION\_ID):  
@ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })  
//@JoinColumn(name = "ORGANISATION\_ID") optional  
private Organisation organisation;

Case 2: One manager (external user) has multiple employees (external user)

@JsonManagedReference(value = "manager")  
@OneToMany(mappedBy = "manager")  
private List<ExternalUser> employees = new ArrayList<>();  
  
@JsonBackReference(value = "manager")  
@ManyToOne(fetch = FetchType.*LAZY*)  
private ExternalUser manager;

Case 3: Multiple external user has one internal user which does not have external user list

@JoinColumn(name = "IDENTITY\_CHECK\_CLAIM\_BY")  
@ManyToOne(fetch = FetchType.*LAZY*)  
private InternalUser identityCheckClaimBy;

##### @JoinTable

It is used for many to many, e.g. One user has multiple roles, one role has multiple users, USER\_ROLE is relationship mapping table which include USER\_ID, ROLE\_ID

User:  
@JoinTable(name = "USER\_ROLE", inverseJoinColumns = { @JoinColumn(name = "ROLE\_ID") })  
@ManyToMany(fetch = FetchType.EAGER)  
protected Set<Role> roles = new HashSet<>();  
  
Role:  
@JoinTable(name = "USER\_ROLE", inverseJoinColumns = { @JoinColumn(name = "USER\_ID")})  
@ManyToMany(fetch = FetchType.EAGER)  
protected Set<User> users = new HashSet<>();

##### @EntityListener

攔截entity存取操作

<https://matthung0807.blogspot.com/2021/06/spring-data-jpa-entitylistener-annotation-intercept-persist-operation.html?m=0>



##### Java Lombok

Lombok 是一個 Java library，可以透過簡單的注解省略 Java 的 code，像是 setter、getter、logger…等，目的在消除冗長的 code 和提高開發效率

<https://kucw.github.io/blog/2020/3/java-lombok/>

##### Spring AOP

AOP is dynamic proxy for the business method. It will execute specify method e.g. access checking, begin transaction, commit transaction, rollback transaction. Round the business method. @Transactional is example of AOP.

##### DB Property Setting

spring.datasource.url=jdbc:mariadb://127.0.0.1:3306/tsw\_p2\_local?useSSL=false&autoReconnect=true  
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

spring.datasource.username=tsw\_p2\_local  
spring.datasource.password=tsw\_p2\_local

Graphical user interface, application

Description automatically generated

##### JPA DataSource for Spring & Multiple Datasources

<https://www.tpisoftware.com/tpu/articleDetails/2637>

@Configuration

@EnableJpaRepositories( //

basePackageClasses = TableADao.class, // Dao Package

entityManagerFactoryRef = "aEntityManagerFactory", //

transactionManagerRef = "aTransactionManager"

)

@EnableTransactionManagement

public class AConfig {

@Bean("aDataSource")

@Primary

@ConfigurationProperties("app.datasource.a")

public DataSource aDataSource() {

return DataSourceBuilder.create().build();

}

@Bean("aEntityManagerFactory")

@Primary

public LocalContainerEntityManagerFactoryBean aEntityManagerFactory(

@Qualifier("aDataSource") DataSource aDataSource,

EntityManagerFactoryBuilder builder) {

return builder //

.dataSource(aDataSource) //

.packages(TableA.class) // Entity Package

.persistenceUnit("aDs") //

.build();

}

@Bean("aTransactionManager")

@Primary

public PlatformTransactionManager aTransactionManager(

@Qualifier("aEntityManagerFactory") LocalContainerEntityManagerFactoryBean aEntityManagerFactory) {

return new JpaTransactionManager(aEntityManagerFactory.getObject());

}

}

@Configuration

@EnableJpaRepositories( //

basePackageClasses = TableBDao.class, // Dao Entity

entityManagerFactoryRef = "bEntityManagerFactory", //

transactionManagerRef = "bTransactionManager"

)

@EnableTransactionManagement

public class BConfig {

@Bean("bDataSource")

@ConfigurationProperties("app.datasource.b")

public DataSource bDataSource() {

return DataSourceBuilder.create().build();

}

@Bean("bEntityManagerFactory")

public LocalContainerEntityManagerFactoryBean bEntityManagerFactory(

@Qualifier("bDataSource") DataSource bDataSource,

EntityManagerFactoryBuilder builder) {

return builder //

.dataSource(bDataSource) //

.packages(TableB.class) // Entity Package

.persistenceUnit("bDs") //

.build();

}

@Bean("bTransactionManager")

public PlatformTransactionManager bTransactionManager(

@Qualifier("bEntityManagerFactory") LocalContainerEntityManagerFactoryBean bEntityManagerFactory

) {

return new JpaTransactionManager(bEntityManagerFactory.getObject());

}

}

After creating data sources successfully, repository will use corresponding data source automatically

##### JPA Repository Save does not work

The transaction is not committed.

It needs to use JpaTransactionManager instead of DataSourceTransactionManager

for data source creating

Problematic record(s) will lead to fail to commit data to DB (if all save methods use one transaction, it will be set to be roll-back only once one record is failing to save) when use save function. To fix this, each JPA.save should be in a new transaction.

##### JPA Pageable & Sort

 Be remember that there is no ; for sql so as JPA can pending limit x,y to the sql

@GetMapping("/page")  
public String page(@RequestParam(value = "page",required = true) int page){  
  
 RoleRepository repository = ApplicationContextProvider.*getBean*(RoleRepository.class);  
 Pageable pageable = PageRequest.*of*(page, 5, Sort.*by*("NAME").ascending());  
 List<Role> list = repository.findRole(pageable);  
 for(Role item : list){  
 System.*out*.println(item.getName());  
 }  
  
 return "complete";  
  
}

##### JPA Save Enum

@GetMapping("/enum")  
@Transactional(transactionManager = "txnManager", rollbackFor = Exception.class)  
public String enum2(){  
  
 executeSQL("SET FOREIGN\_KEY\_CHECKS=0;");  
 Role.RoleBuilder builder = Role.*builder*();  
 builder.name("test456").subAdminAssignable(true).department(Department.*CEDB*)  
 .switchable(1).deleted(false).recordVersion(0);  
 RoleRepository repository = ApplicationContextProvider.*getBean*(RoleRepository.class);  
 repository.save(builder.build());  
  
 executeSQL("SET FOREIGN\_KEY\_CHECKS=1;");  
 return "complete";  
}

##### JPA DTO

For HQL in DTO, it needs to use java variable instead of DB column, e.g. phoneNo instead of phone\_no, otherwise, it cannot create query.

DTO must have constructor with all properties.

DTO can be entity for the repository, it needs @Entity @Id for the DTO class and make sure the DTO’s package is scanned by the database configure

@Bean(name="p1EntityManagerFactory")  
public LocalContainerEntityManagerFactoryBean entityManagerFactory(  
 EntityManagerFactoryBuilder builder,  
 @Qualifier("p1DataSource") DataSource dataSource) {  
  
 return builder  
 .dataSource(dataSource)  
 .packages("hk.gov.cedb.tsw.p1.domain")  
 //.packages("hk.gov.cedb.tsw.p1.domain","hk.gov.cedb.tsw.p1.dto")  
 .persistenceUnit("p1PersistenceUnit")  
 .build();  
}

##### Fail to Run Main Method

Possible Reason:

1. Port 8080 is used
2. VPN/Putty is on, IDEA is fail to access Maven central repository
3. Fail to access to development network (e.g. DB connection, Minio connection), Spring cannot create bean.

##### Spring MVC

Diagram

Description automatically generated

<https://cloud.tencent.com/developer/article/1116654?from=article.detail.1800493>

##### Spring Start Up Flow

<https://www.cnblogs.com/summerday152/p/13639896.html>

It uses ClassPathXmlApplicationContext or AnnotationConfigApplicationContext , the flow is same

Diagram

Description automatically generated

Diagram

Description automatically generated

##### Spring Main Application

It is entry of application, remember that this class should be put in the root package so that ALL classes under the root package can be scanned by Spring

##### Spring CommandLineRunner

It is used to execute specify logic(only one time) after Spring Application initialization (Main method will be executed before it)

<https://www.cnblogs.com/chenpi/p/9696310.html>

@Component  
**public class** ApplicationStartupRunner **implements** CommandLineRunner {  
 **protected final** Log **logger** = LogFactory.*getLog*(getClass());  
  
 @Override  
 **public void** run(String... args) **throws** Exception {  
 **logger**.info(**"ApplicationStartupRunner run method Started !!"**);  
 }  
}

Developer can also follow approach to get the same achievement.

@Component  
**public class** ApplicationStartup {  
  
 @EventListener(ContextRefreshedEvent.**class**)  
 **public void** contextRefreshedEvent(){  
 System.***out***.println(**"run ApplicationStartup"**);  
 }  
}

##### Spring ApplicationRunner

It is used to execute specify logic(only one time)/access application arguments after Spring Application initialization (Main will be executed before it, CommandLineRunner will be executed after it)

@Slf4j  
@Component  
**public class** MyApplicationRunner **implements** ApplicationRunner {  
  
 @Override  
 **public void** run(ApplicationArguments args) **throws** Exception {  
 ***log***.info(**"ApplicationStartupRunner run method Started !!"**);  
 }  
}

##### Spring Session (Redis)

Spring Session is used to managed session for cluster server. It uses Filter to intercept request and save session(s) to database (e.g. Mysql, Redis) instead of application server.

Session will be created automatically if client side send request first time or the current session

is expired, but developer is used to save login user in the session.

If current session is destroy, it will create another new session and save it to Redis (if application is using Redis to save session) if client send request again.

<https://codingnote.cc/zh-tw/p/179345/>

原理:

<https://www.cnblogs.com/54chensongxia/p/12096493.html>

implementation("org.springframework.boot:spring-boot-starter-data-redis")// for session repository  
implementation("org.springframework.session:spring-session-data-redis")//for session

Spring will create Redis connection and save session to Redis DB automatically, developer only needs to config the Redis in property file, please note that Spring application will fail to start up if the Redis config is missing or incorrect.

To start Redis server (Window):

1. Open Redis folder with CMD. e.g. C:\Program Files\Redis-5.0.14.1
2. Type redis-server.exe redis.windows.conf

To set password (Window):

1. Open Redis folder with CMD. e.g. C:\Program Files\Redis-5.0.14.1
2. Type redis-cli.exe -h 127.0.0.1 -p 6379
3. Type config set requirepass 123456

To start Redis client (Window):

1. Download tools in <https://github.com/qishibo/anotherredisdesktopmanager/releases>
2. Create connection

Graphical user interface, text, application

Description automatically generated

##### Spring Security

Spring Security is a framework that focuses on providing both authentication and authorization to Java applications

Spring Security is using Filter to intercept request

<https://ithelp.ithome.com.tw/articles/10250502>

<https://yen0304.github.io/p/spring-security2%E4%BD%BF%E7%94%A8spring-security%E8%87%AA%E5%AE%9A%E7%BE%A9%E4%BD%BF%E7%94%A8%E8%80%85%E5%B8%B3%E5%AF%86%E4%BD%BF%E7%94%A8jdbc%E4%B8%B2%E6%8E%A5%E8%B3%87%E6%96%99%E5%BA%AB/>

<https://mp.weixin.qq.com/s/vxw4EeJ-LWxQtD0xlyB0nA>

implementation("org.springframework.boot:spring-boot-starter-security")

Spring will perform authentication automatically once the Spring Security is added into application

Text

Description automatically generated

##### Spring Listener

<https://www.cnblogs.com/rickiyang/p/12001524.html>

##### Spring JPA Audit

It is used to inserted createBy,createDate,lastModifyBy,lastModifyDate to DB by Spring. The framework will use current system time for createDate && lastModifyDate and current login user(provide by AuditorAware) for createBy && lastModifyBy.

<https://www.jdon.com/springboot/spring-data-jpa-auditing.html>

Remember to enable JPA audit in database configure

@EnableJpaAuditing(auditorAwareRef = "tswAuditorAware")  
public class DatabaseConfiguration {

##### Injection for persistence object

Hibernate 5.3 introduces a new service contract called ManagedBeanRegistry.

[tps://github.com/hibernate/hibernate-orm/blob/master/hibernate-core/src/main/java/org/hibernate/resource/beans/spi/ManagedBeanRegistry.java](https://github.com/hibernate/hibernate-orm/blob/master/hibernate-core/src/main/java/org/hibernate/resource/beans/spi/ManagedBeanRegistry.java)

This service contract would enable users of Spring to define Spring managed beans for various persistence objects such as Attribute Converters, Event Listeners, and Revision Listeners allowing Hibernate to lookup and use those beans as needed.

**Issue Links:**

* [#20850](https://github.com/spring-projects/spring-framework/issues/20850) Support for Hibernate ORM 5.3 (***"is depended on by"***)
* [#21548](https://github.com/spring-projects/spring-framework/issues/21548) org.springframework.orm.hibernate5.SpringBeanContainer ignores the fallback bean instance producer
* [#21696](https://github.com/spring-projects/spring-framework/issues/21696) Support dependency injection in JPA AttributeConverter for EclipseLink
* [#21540](https://github.com/spring-projects/spring-framework/issues/21540) LocalSessionFactoryBean and HibernateTransactionManager for JPA EntityManagerFactory setup
* [#21494](https://github.com/spring-projects/spring-framework/issues/21494) Propagate read-only status to Hibernate Session through setDefaultReadOnly
* [#21368](https://github.com/spring-projects/spring-framework/issues/21368) Apply Hibernate Integrator through LocalSessionFactoryBean
* [#21581](https://github.com/spring-projects/spring-framework/issues/21581) Hibernate 5: restore ability to inject "cacheRegionFactory" of LocalSessionFactoryBean

##### ApplicationContextAware

It is used to get ApplicationContext



<https://www.cnblogs.com/loong-hon/p/10917755.html>

##### ResetTemplate

The original Spring REST client with a synchronous, template method API.

<https://www.cnblogs.com/54chensongxia/p/11414923.html>

##### WebMvcConfigurer

WebMvcConfigurer是一个接口，提供很多自定义的拦截器，例如跨域设置、类型转化器等等。可以说此接口为开发者提前想到了很多拦截层面的需求，方便开发者自由选择使用。由于Spring5.0废弃了WebMvcConfigurerAdapter，所以WebMvcConfigurer继承了WebMvcConfigurerAdapter大部分内容。

<https://developer.aliyun.com/article/617307>

##### HttpSessionListener

It is used to detect session created and destroyed



To use WebServlet,WebFilter,WebListener, it needs @ServletComponentScan in main application

@SpringBootApplication  
@ServletComponentScan//it needs this annotation if application is using WebServlet, WebFilter, WebListener  
public class MainApplication {

##### I18N (Multiple Language)

Spring Boot自動配置MessageSource做i18n多國語言訊息，不用再自己配置MessageSource的bean且預設會尋找classpath根目錄下名稱為messages的properties作為訊息來源。如果想要使用其他名稱或路徑，可在Spring Boot配置檔application.properties設定。

<https://www.tpisoftware.com/tpu/articleDetails/2347>



@GetMapping("/locale")  
public String locale(HttpServletRequest request, HttpServletResponse response) throws Exception{  
 Locale locale = LocaleContextHolder.*getLocale*();  
 return messageSource.getMessage("current.locale", null, locale);  
}

##### Interceptor && Filter

<https://www.readfog.com/a/1636486590853386240>

Filter is servlet component which cannot use Spring resource, it recommends to use Interceptor which is Spring component.

Diagram

Description automatically generated

Table

Description automatically generated

#### Java

##### String… args & String[] args

They have the same purpose but just (String… args)provides more readability and easiness to use. String… args means an arbitrary number of values required include no parameter while String[] args means the parameter is array and mandatory.

##### Protected Method

Protected method in java can be called by subclass or object which is in the same package,

##### For Loop

List<String> list = **null**;  
 *for(String item : list){ // it will throw exception for null List  
   
}*

##### Java Generic

Generic is to indicate the parameter type is dynamic. E.g. private T x, which x is parameter name and T is dynamic parameter/object type, it can be Integer, Float , customer class and so on.

<https://ethan-imagination.blogspot.com/2018/11/javase-gettingstarted-017.html>

Example:

**public static** <T> List<T> parseArray(String response, Class<T> Object){  
 List<T> list = **null**;  
 **return** list;  
}

<T> behind static is to define this method is using generic, it cannot be removed, otherwise, it will compile error.

List<T> is return type

Class<T> object means T.class

call this method:

List<User> userList = *parseArray*(**"json response"**, User.**class**);

-------

This class has generic parameter.

class CollectionGeneric<T extends Collection> {

private T collection;

CollectionGeneric(T collection){

this.collection = collection;

}

}

CollectionGeneric c = **new** CollectionGeneric(**new** ArrayList());c.show();  
CollectionGeneric<ArrayList> c1 = **new** CollectionGeneric<>(**new** ArrayList());c1.show();  
CollectionGeneric c2 = **new** CollectionGeneric<>(**new** ArrayList());c2.show();  
CollectionGeneric<ArrayList> c3 = **new** CollectionGeneric(**new** ArrayList());c3.show();  
CollectionGeneric<ArrayList> c4 = **new** CollectionGeneric<ArrayList>(**new** ArrayList());c4.show();

##### Exception & RuntimeException

Diagram

Description automatically generated

Exception 是 Throwable 的一個子類。**Exception 表示合理的應用程式可能想要捕獲的條件。**\*\*編譯器會檢查 Exception 異常。\*\*此類異常，要麼通過 throws 進行聲明拋出，要麼通過 try catch 進行捕獲處理，否則不能通過編譯。

RuntimeException 是 Exception 的一個子類。RuntimeException 是那些可能在 Java 虛擬機正常運行期間拋出的異常的超類。\*\*編譯器不會檢查 RuntimeException 異常。\*\*當程序中可能出現這類異常時，倘若既沒有通過 throws 聲明拋出它，也沒有用 try catch 語句捕獲它，程序還是會編譯通過。

##### Session & Cookie

Session is used to identity user, server will create and save session and return session id to client by cookie(session id is saved in client side by cookie) when client accesses server first time, after that, client will send request with this session id so that server can identity client. General speaking, the login information (i.e. username) is saved in session upon authorization is completed. To judge the client is login or not, it can get login information from the session which is obtained by session id sent from client side.

To send cookie to client, it should use

response.addCookie(cookie);

<https://www.liaoxuefeng.com/wiki/1252599548343744/1328768897515553>

##### Filter

Graphical user interface

Description automatically generated

<https://www.liaoxuefeng.com/wiki/1252599548343744/1266264823560128>

##### Cross Site Request Forgery Token (CSRF)

<https://tech-blog.cymetrics.io/posts/jo/zerobased-cross-site-request-forgery/>

Text

Description automatically generated

Spring Security uses Filter to intercept the request to generate/compare CSRF token for validation.

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/296977/>

<https://www.cnblogs.com/pengdai/p/12164754.html>

##### JSON Web Token (JWT)

<https://jwt.io/introduction>

Diagram

Description automatically generated

Session & Cookie

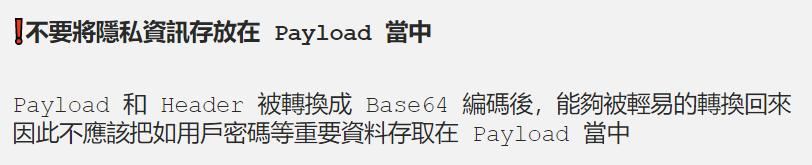


JWT

Chart, timeline

Description automatically generated





<https://medium.com/%E9%BA%A5%E5%85%8B%E7%9A%84%E5%8D%8A%E8%B7%AF%E5%87%BA%E5%AE%B6%E7%AD%86%E8%A8%98/%E7%AD%86%E8%A8%98-%E9%80%8F%E9%81%8E-jwt-%E5%AF%A6%E4%BD%9C%E9%A9%97%E8%AD%89%E6%A9%9F%E5%88%B6-2e64d72594f8>

Demo

<https://www.javainuse.com/spring/boot-jwt>

##### CORS

Cross-Origin Resources Sharing

<https://shubo.io/what-is-cors/>

##### Cookie

Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server. A cookie's value can uniquely identify a client, so cookies are commonly used for session management.

A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number. Some Web browsers have bugs in how they handle the optional attributes, so use them sparingly to improve the interoperability of your servlets.

The servlet sends cookies to the browser by using the HttpServletResponse.addCookie method, which adds fields to HTTP response headers to send cookies to the browser, one at a time. The browser is expected to support 20 cookies for each Web server, 300 cookies total, and may limit cookie size to 4 KB each.

The browser returns cookies to the servlet by adding fields to HTTP request headers. Cookies can be retrieved from a request by using the HttpServletRequest.getCookies method. Several cookies might have the same name but different path attributes.

##### HttpServletRequest & HttpServletResponse

It can use following code the get HttpServletRequest & HttpServletResponse

**public static** HttpServletRequest getRequest() {  
 **return** ((ServletRequestAttributes) RequestContextHolder.*getRequestAttributes*()).getRequest();  
}  
  
**public static** HttpServletResponse getResponse() {  
 **return** ((ServletRequestAttributes)RequestContextHolder.*getRequestAttributes*()).getResponse();  
}

##### Dynamic Proxy

Dynamic Proxy is used to proxy(use reflection) ALL interface/class to execute customer business logic before/after the actual method.

<https://segmentfault.com/a/1190000040571286>

##### Reflection

<http://www.51gjie.com/java/777.html>

Java中，无论生成某个类的多少个对象，这些对象都会对应于同一个Class对象，这个Class对象是由JVM生成的，通过它能够获悉整个类的结构。要想使用反射，首先需要获得待操作的类所对应的Class对象

获取class对象的3种方法

1. 已经得到一个类的实例，可以使用如下方式来得到Class对象.

Class c = 对象名.getClass();

2. 如果在编译期知道类的名字，可以使用如下方法.

Class c = java.awt.Button.class;

或

Class c = Integer.TYPE;

3. 如果类名在编译期不知道, 但是在运行期可以获得, 可以使用下面的方法

Class c = Class.forName(str); //注意:str是类的全路径

**Class c is the same object for above method to create Class.**

##### Multiple Thread

public class Main {  
 public static void main(String[] args) {  
 Thread t = new Thread(new MyRunnable());  
 t.start(); *// 启动新线程* }  
}  
  
class MyRunnable implements Runnable {  
 @Override  
 public void run() {  
 System.*out*.println("start new thread!");  
 }  
}

<https://www.liaoxuefeng.com/wiki/1252599548343744/1306580710588449>

##### Callable

The Callable interface is similar to Runnable, in that both are designed for classes whose instances are potentially executed by another thread. A Runnable, however, does not return a result and cannot throw a checked exception.

##### Java encryption vs hashing

Encryption is a two-way function that includes encryption and decryption whilst hashing is a one-way function that changes a plain text to a unique digest that is irreversible

##### Java Synchronized

<https://pdai.tech/md/java/thread/java-thread-x-key-synchronized.html>

##### Java Lock

<https://www.cnblogs.com/dolphin0520/p/3923167.html>

##### Java Thread Safe

<https://www.baeldung.com/java-thread-safety>

##### Java Collection

Graphical user interface, text, application

Description automatically generated

<https://www.runoob.com/java/java-collections.html>

##### Java ThreadLocal

It is used to save Object/Parameter for thread

<https://www.liaoxuefeng.com/wiki/1252599548343744/1306581251653666>

##### Set JVM Parameter

To change jvm parameter (e.g. default charset, time zone), it needs to use following command to start a jar

java -Dfile.encoding=UTF-8 ./build/libs/TestStart-1.0.jar, it changes the default charset for JVM, new string in Java will use default charset(same as the running machine) to create string if it does not specify charset, it may show 爛字 for Chinese if it use default charset.

#### Design Pattern

##### Abstract Factory Pattern

Concrete factory is for concrete product, client NO need to know how to create concrete product (i.e. wrap, it is very important concept in OOP which only provides method/interface for client), instead, client only know use the concrete factory which should implements the abstract factory to get concrete product, if client wants for another concrete product, it only needs to change concrete factory.

Diagram

Description automatically generated

<https://ithelp.ithome.com.tw/articles/10208955>

##### Builder Pattern

<https://www.jianshu.com/p/3d1c9ffb0a28>

<https://ithelp.ithome.com.tw/articles/10204732>

Diagram

Description automatically generated

Text

Description automatically generated

##### Proxy Pattern

It used to do action for client and add customer business logic. E.g. Spring transaction

Diagram

Description automatically generated

<https://ithelp.ithome.com.tw/articles/10205659>

##### Adapter Pattern

Diagram

Description automatically generated

##### Observer Pattern

It is used to notify when object changes state.

Diagram

Description automatically generated

<https://ithelp.ithome.com.tw/articles/10204117>

#### SQL

##### MySQL Table “my\_table” was not locked with Lock Tables

If in one session, you locked one table but want to select from another table, you must either lock that table too or unlock all tables.

MySQL enables client sessions to acquire table locks explicitly for the purpose of cooperating with other sessions for access to tables, or to prevent other sessions from modifying tables during periods when a session requires exclusive access to them. A session can acquire or release locks only for itself. One session cannot acquire locks for another session or release locks held by another session.

Locks may be used to emulate transactions or to get more speed when updating tables. This is explained in more detail later in this section.

LOCK TABLES explicitly acquires table locks for the current client session. Table locks can be acquired for base tables or views. You must have the LOCK TABLES privilege, and the SELECT privilege for each object to be locked.

For view locking, LOCK TABLES adds all base tables used in the view to the set of tables to be locked and locks them automatically. If you lock a table explicitly with LOCK TABLES, any tables used in triggers are also locked implicitly, as described in Section 13.3.5.2, “LOCK TABLES and Triggers”.

UNLOCK TABLES explicitly releases any table locks held by the current session. LOCK TABLES implicitly releases any table locks held by the current session before acquiring new locks.

##### Use IFNULL for nullable column

**SELECT** ID,USERNAME,ENG\_FIRST\_NAME,**IFNULL**(**NAME**,'') **AS** **NAME**

**FROM** tsw\_p1\_local.prolonged\_registration\_audit **WHERE** **NAME** **IS** **null**

##### Substring

**SELECT** **left**('1234567890', 3) **AS** t1, **right**('1234567890', 3) **AS** t2 ;

##### Select \* into table1 from table2 & Insert into table2(\*,\*) select \*,\* from table1

Select \* into will create table2 and copy data from table1 (this is not supported by MySql)

Insert into will copy data from table1 which must be existing.

##### Delete & Truncate

<https://www.geeksforgeeks.org/difference-between-delete-and-truncate/>

In the TRUNCATE command, the transaction log for each deleted data page is not recorded. Unlike the DELETE command, the TRUNCATE command is fast. We cannot roll back the data after using the TRUNCATE command.

Truncate will not roll back in Spring transaction(it will commit transaction implicitly instead by Spring transaction) even an exception is thrown while Delete can roll back

##### Schema

Text

Description automatically generated

##### Display Create Table Script

**SHOW** **CREATE** **TABLE** tsw\_p2\_local.subscription;

##### Check lock table

**SELECT** \* **FROM** innodb\_locks;

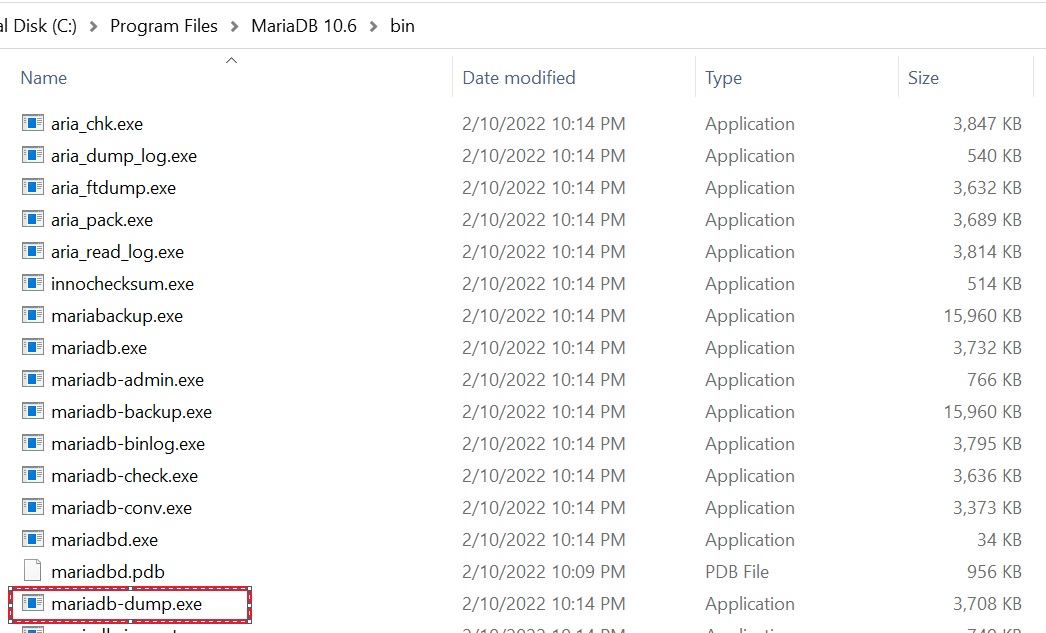
**SELECT** \* **FROM** innodb\_lock\_waits;

**SELECT** \* **FROM** innodb\_trx;

##### Export Data by Command

It is used to exported database/table by command

Open folder which includes sql dump exe, e.g.



Run following command in cmd to export tables(includes data)

mysqldump -u username -p DBName **table1 table2** > C:\Users\Dell\Desktop\Notes\dcm.**sql**

Example to dump DB without data

mysqldump -P 3306 -h 192.168.8.65 -u tsw\_dev -p --no-data --routines --no-create-db tsw\_dev > C:\Users\Dell\Desktop\Notes\tsw\_dev\_ddl.sql

##### Import Data by Command

Open folder which includes sql dump exe(Refer to Sql Dump) and run following command:

mysql -u root -p tsw\_p1\_local < C:\\Users\\Dell\\IdeaProjects\\TSW-P2\\test.sql

root is username, tsw\_p1\_local is DB name

##### Plugin client\_ed25519 could not be loaded

Copy following file to plugin folder



#### Gradle

##### Jib Plugin

Jib builds optimized Docker and [OCI](https://github.com/opencontainers/image-spec) images for your Java applications without a Docker daemon - and without deep mastery of Docker best-practices. It is available as plugins for [Maven](https://github.com/GoogleContainerTools/jib/blob/master/jib-maven-plugin) and [Gradle](https://github.com/GoogleContainerTools/jib/blob/master/jib-gradle-plugin) and as a Java library.

<https://github.com/GoogleContainerTools/jib/>

#### WinSCP

It is used to upload files from Window to Linux

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

#### Docker

##### Open Container Initiative (OCI)

To define Runtime Specification & Image Specification

<https://ithelp.ithome.com.tw/articles/10216215/>

#### Linux

##### Command

<https://blog.techbridge.cc/2017/12/23/linux-commnd-line-tutorial/>

cd foldername to open folder of current directory

cd .. to move to the super directory

cd ../foldername to move to super directory and then move the specify directory

cd / to move to root directory

pwd to show current path

ls -la to list all directories/files with detail of current directory, or use ls -la \*.js to show files which are end with .js

cat static-portal.yaml to view content of specify file, or use cat static\*.yaml

touch test.xml to create file in current directory

copy/mv file1 test2/file2 to copy/move file1 in current directory to test2 directory and named as file2

mv file1 file2 to rename file1 to file2

rm file1 to delete file1

rm -d directory1 to delete directory1(must be empty)

rm -r directory1 to delete directory1(can be not empty)

tail -f file.log to keep showing the last updated content of file (e.g. check updated log)

wget http:… to download something from URL

sudo apt install openjdk-17-jre-headless to install JDK

java –version to check JDK version

docker –version to check Docker version

. test.sh to execute test.sh in current directory

vim test.xml to view file, input i to entry edit mode, input Esc to exit edit mode, input :wq to save and exit, input :q! to exit without save.

clear to clear screen

shutdown -h now to shut down Linux

copy files from A(login with A) to B:

scp /home/tswadmin1/command.txt <ssh username>@<B host>:/home/tswadmin1/command.txt

SSH to another server:

ssh <username>@<host>

A picture containing graphical user interface

Description automatically generated

##### exec

The Linux **exec** command executes a Shell command without creating a new process. Instead, it replaces the currently open Shell operation. Depending on the command usage, **exec** has different behaviors and use cases.

<https://phoenixnap.com/kb/linux-exec>

#### Cloud Native

<https://datadrivenai.wordpress.com/2019/11/01/%E4%BB%80%E9%BA%BC%E6%98%AF-cloud-native%EF%BC%9F-%E7%82%BA%E4%BB%80%E9%BA%BC-cloud-native-%E9%80%99%E9%BA%BC%E7%86%B1%E9%96%80%EF%BC%9F/>

Cloud Native 用來表示已經包裝好容器 (Containerize) 的應用、軟體，也就是說 Cloud Native 是用來指應用程式已經被容器包裝好可以直接 deploy 成一個 Microservice 或是變成一個很有彈性的架構經過 DevOps 或是 CI/CD 的流程可以直接變成一個新的服務

#### GitOps

Old CI/CD

Diagram

Description automatically generated

New CICD

Diagram

Description automatically generated

<https://www.hwchiu.com/gitops.html>

#### K8s

Container is isolated unit for resource (i.e. CPU, memory, OS) for application, docker is tool for container, Pod contains one or more containers, which is virtual resources for K8s. Application is presented as image which put in docker.

Graphical user interface, application

Description automatically generated

Node is working station, one node contains multiple pods. Master Node is to manager to all Nodes.

Schedule in Master Node is to assign Pod to Node according to the resource of Node.

Controller Manager in Master Node is to make Pod work as expected. e.g. make sure 10 Pods is working as expected.

Container Runtime is container manager of Pods of Node, it will download image from Docker Hub and start the container. Kubelet communicates with API server in Master Node and manages Pods through Container Runtime, such as restart/stop Pod.

Diagram

Description automatically generated

Diagram

Description automatically generated

NodePort Service is router for request, it will send request to corresponding Pod by app name. It exposes service to Internat.

LoadBalancer Service is for load balancer.

ClusterIP is for micro service communication, it is assigned by K8s.

Diagram

Description automatically generated

ReplicaSet is to ensure high availability for k8s. E.g. start a Pod automatically if it is down.

Diagram

Description automatically generated

ConfigMap API is a mechanism to provide config information for container, it can save single property in key-value format as well as config file path or JSON object.

Volume is to save persistent data for container in Pod, it will be destroyed when the corresponding Pod is destroyed.

Docker 作為容器工具可以把：業務邏輯容器、數據庫容器、儲存容器、隊列容器使得軟件可以拆分成若干個標準化容器，然後像搭積木一樣組合起來，讓彼此通信，從而形成微服務。因此微服務很適合用 Docker 容器實現，每個容器承載一個服務。一台計算機同時運行多個容器，從而就能很輕鬆地模擬出複雜的微服務架構

#### Nexus Repository

It is used to create Maven private repository for company

<https://blog.csdn.net/chetuodao0481/article/details/100849458?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-2.pc_relevant_default&utm_relevant_index=5>

#### Jetkin

<https://www.youtube.com/watch?v=SbMabIXQd_A&list=PLmOn9nNkQxJE_3wrOfHdL1dWRY6CCHBnh&index=1>

#### Git

Diagram

Description automatically generated

Text, letter

Description automatically generated

git init is to create local repository for current folder

git add filename is to add the specify file to local repository, you can use git add . to add ALL files to local repository

git commit -m “commit comment” to commit the files to local repository

git remote add origin <https://.....git> is to connect with remote repository, origin is remote name, you can use git remote to display all remote name

git pull origin master to pull code from remote branch to local repository

git push -u(optional) origin master is to push files from local repository to remote repository

git clone <https://......git> is to download files from git, please note that git will not create local repository for current folder when it executes this command.

git status is to list the files you've changed and those you still need to add or commit.

#### Firewall

Text

Description automatically generated

<https://www.youtube.com/watch?v=jHg-Wx--a3I>

#### WAF & LTM

A WAF or web application [firewall](https://www.cloudflare.com/learning/security/what-is-a-firewall/) helps protect web applications by filtering and monitoring [HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/) traffic between a web application and the Internet. It typically protects web applications from attacks such as [cross-site forgery](https://www.cloudflare.com/learning/security/threats/cross-site-request-forgery/), [cross-site-scripting (XSS)](https://www.cloudflare.com/learning/security/threats/cross-site-scripting/), file inclusion, and [SQL injection](https://www.cloudflare.com/learning/security/threats/sql-injection/), among others

LTM is form Local Traffic Manager

#### NIDS & HIDS

Network-based Intrusion Detection, System Host-based Intrusion Detection System

NIDS is to monitor network while HIDS is to monitor system and files

<http://www.netqna.com/2014/04/network-based-intrusion-detection-system.html>

#### DMZ

Demilitarized Zone

<https://www.youtube.com/watch?v=dqlzQXo1wqo>

#### WAN & LAN

Wide Area Network & Local Area Network

<https://kknews.cc/zh-hk/tech/4qml8yv.html>

#### VLAN

Virtual Local Area Network

<https://www.jannet.hk/virtual-lan-vlan-zh-hant/>

#### Router & Switch

<https://bluemuta38.pixnet.net/blog/post/45543357>

SWITCH會記錄封包中的MAC位址, 傳送的資料封包每一個都必須經過SWITCH判斷決定要送往哪一台電腦,所以當電腦A傳送資料給電腦B時，其他電腦並不會也收到資料，而且這個時候別的電腦也可以同時互相傳送資料.

而ROUTER最主要的工作就是判別收到的IP封包要往哪一個介面(WAN/LAN)送

#### Proxy & VPN

Proxy has cache and hide real IP, VPN is able to hide real IP and encrypt content

<https://nordvpn.com/zh-tw/blog/vpn-proxy-bijiao/>

<https://www.ithome.com.tw/news/5006>

#### Telnet & SSH

Telnet is a terminal emulation programm that is used to access remote server, so that it can execute command from client side. All commands are sent in clear text and no encryption. For Window system, user need to enable Telnet client before use it.

SSH is encryption of Telnet

<https://www.youtube.com/watch?v=tZop-zjYkrU>

#### Putty

Putty is a tool to provide SSH connection between two computers

#### Access to Hub Development Environment from Home

|  |  |  |
| --- | --- | --- |
| VPN/Putty | Git | DB |
| On/On | Success with proxy, fail without proxy | Fail |
| On/Off | Fail | Fail |
| Off/Off | Fail | Fail |

#### Access to Hub Development Environment from Hub

|  |  |  |
| --- | --- | --- |
| VPN/Putty | Git | DB |
| On/On |  |  |
| On/Off |  |  |
| Off/Off |  |  |

#### Mail Server, SMTP, POP3, IMAP

Simple Mail Transfer Protocol, SMTP, protocol to send email

Post Office Protocol Version 3, POP3, protocol to receive email from mail server, only support online mode, it will download email from mail server and on copy leave on the server

Internet Message Access Protocol, IMAP, protocol to receive email from mail server, support online & offline mode, it will cache email from mail server instead of download and a copy will leave on server as well as sync email/folder to different devices.

<https://www.stockfeel.com.tw/%EF%BD%8Dail-%E4%BC%BA%E6%9C%8D%E5%99%A8%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F%E5%A6%82%E4%BD%95%E9%81%8B%E7%94%A8%EF%BC%9F/>

Different between POP3 and IMAP

<https://www.youtube.com/watch?v=SBaARws0hy4>

#### Web Server & Application Server

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Type of Server

<https://www.stockfeel.com.tw/%E4%BC%BA%E6%9C%8D%E5%99%A8%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F%E6%9C%89%E5%93%AA%E4%BA%9B%E7%A8%AE%E9%A1%9E%EF%BC%9F/?utm_source=graph_search>

#### DNS Server

Domain Name System

<https://www.stockfeel.com.tw/dns-%E4%BC%BA%E6%9C%8D%E5%99%A8%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F%E5%A6%82%E4%BD%95%E9%81%8B%E7%94%A8%EF%BC%9F/>

#### DHCP

Dynamic Host Configuration Protocol, is to assign(lease) IP for computer, router have DHCP service build in it.

<https://www.stockfeel.com.tw/dhcp%E4%BC%BA%E6%9C%8D%E5%99%A8%E6%98%AF%E4%BB%80%E9%BA%BC%EF%BC%9F%E5%A6%82%E4%BD%95%E9%81%8B%E7%94%A8%EF%BC%9F/>

<https://www.youtube.com/watch?v=e6-TaH5bkjo>

#### FTP & SFTP & TFTP

File Transfer Protocol, is used to transfer files between client and server thorough Internet

Secure File Transfer Protocol, is used to transfer encrypt files between client and thorough Internet, both FTP & SFTP use TCP to transfer files, it guarantees the deliverable of file transferring.

Trivial File Transfer Protocol, is used to transfer files (e.g. configure file) to network device(e.g. firewall, router) in local network, not used to transfer files over the internet. It uses UPD and does not guarantee the deliverable of file transferring.

<https://www.youtube.com/watch?v=tOj8MSEIbfA>

#### HTTP & HTTPS & SSL & TLS

Hyper Text Transfer Protocol, is standard for viewing web page, all information is sent in clear text, actually it uses TCP to transfer data.

Secure Hyper Text Transfer Protocol, all information is sent with encryption.

Secure Socker Layer, is used to ensure security (data encryption) for HTTPS transfer, an SSL certificate is used to authenticate the identity of a website.

Transport Layer Security, successor of SSL.

<https://www.stockfeel.com.tw/%E4%BD%A0%E4%B8%8D%E5%8F%AF%E4%B8%8D%E7%9F%A5%E7%9A%84%E7%B6%B2%E8%B7%AF%E8%A1%8C%E9%8A%B7%E7%9F%A5%E8%AD%98%EF%BC%88ii%EF%BC%89-ssl/?utm_source=graph_search>

<https://www.youtube.com/watch?v=hExRDVZHhig>

<https://tw.alphacamp.co/blog/http-https-difference>

#### TCP & UDP

Transfer Control Protocol

User Datagram Protocol

<https://nordvpn.com/zh-tw/blog/tcp-udp-bijiao/>

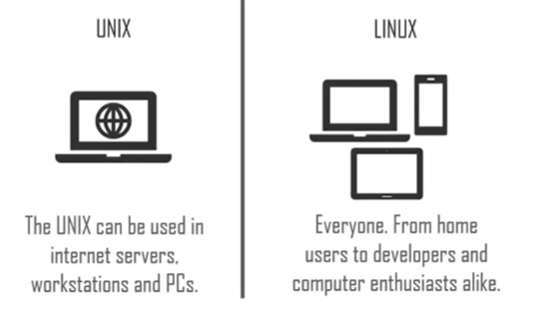
<https://www.youtube.com/watch?v=uwoD5YsGACg>

#### Linux & Unix

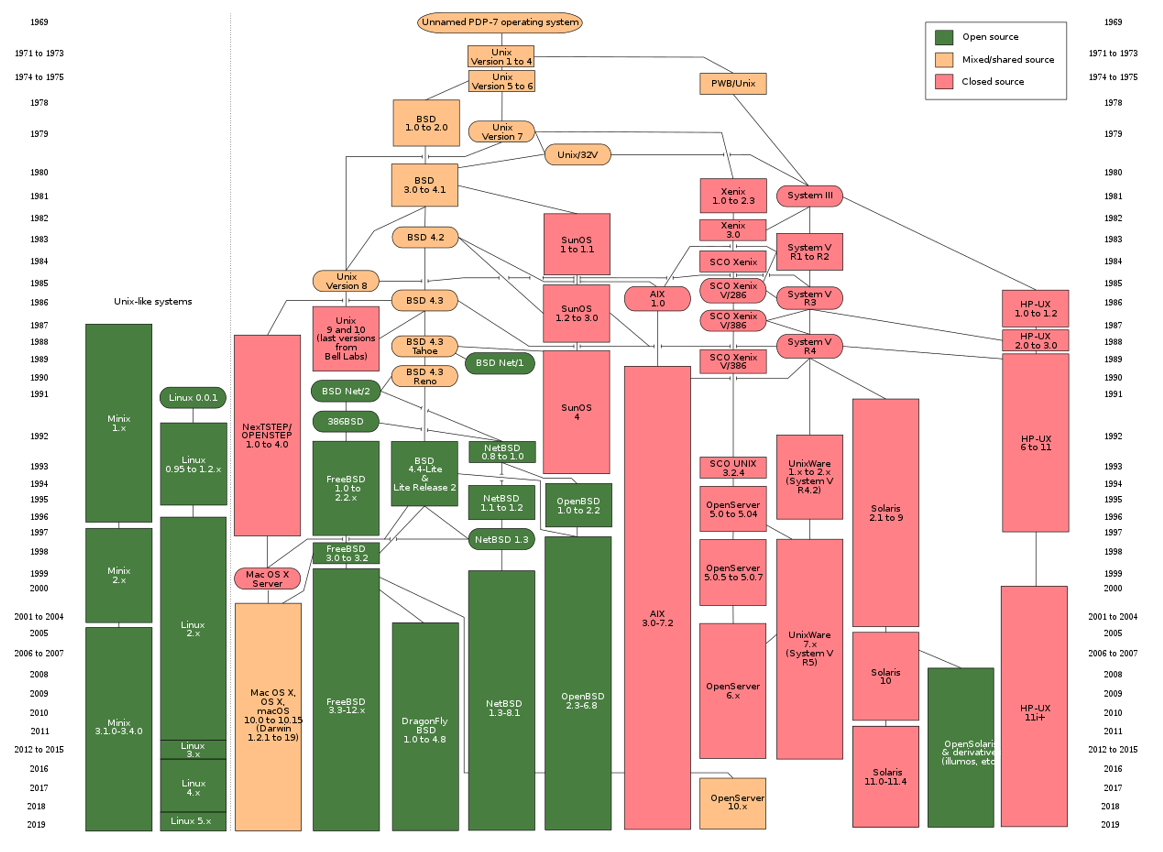
Unix is commercial system and closed source while Linux is free and open source, Linux is Unix-Link system, which has the same function as Unix.

Linux主要應用於網絡服務器（各種各樣的網絡服務解決方案）因為比較穩定，對硬體資源需求比較低，科學運算（各種各樣的科研項目解決方案）、軟件開發平台，嵌入式系統（日常生活中各種各樣的智能電器）

Unix is used in server, workstation, PCs



Branch of Unix



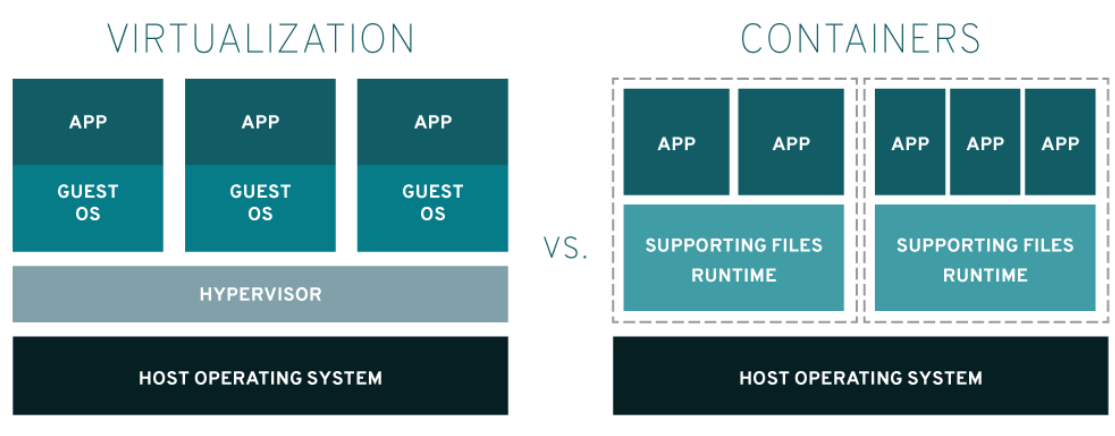
<https://www.youtube.com/watch?v=7L5JMj82vYU>

#### VM & Container

虛擬機器是實體電腦的虛擬呈現或模擬環境。它們通常稱為訪客，而用來執行的實體機器則稱為主機。

[虛擬化](https://www.ibm.com/tw-zh/cloud/learn/virtualization-a-complete-guide)可讓您在單一實體機器上建立多個虛擬機器，每個虛擬機器都有自己的作業系統 (OS) 和應用程式。虛擬機器無法與實體電腦直接互動。而是需要藉助一個叫做 [Hypervisor](https://www.ibm.com/tw-zh/cloud/learn/hypervisors) 的輕量型軟體層，在虛擬機器和底層的實體硬體之間進行協調。Hypervisor 負責將實體運算資源（例如處理器、記憶體及儲存設備）配置給每個虛擬機器。它讓虛擬機器之間相互分開，這樣就不會互相干擾。

容器通常以兆字節為度量單位。它們封裝的內容大小不會超過一個應用及其運行所需的所有文件，並且通常用於封裝執行特定任務的單個功能（稱為微服務）。容器因其輕量級特性及共享的操作系統（OS），非常便於在多個環境之間移動





<https://www.ibm.com/tw-zh/cloud/learn/virtual-machines>

<https://www.redhat.com/zh/topics/virtualization/what-is-a-virtual-machine>

<https://www.redhat.com/zh/topics/containers/containers-vs-vms>

#### Restful API

Representational State Transfer API, it is design style instead of standard.

<https://www.youtube.com/watch?v=PlaKAMShvHc>

Text

Description automatically generated