

Backend Test

API Document Version 1.0

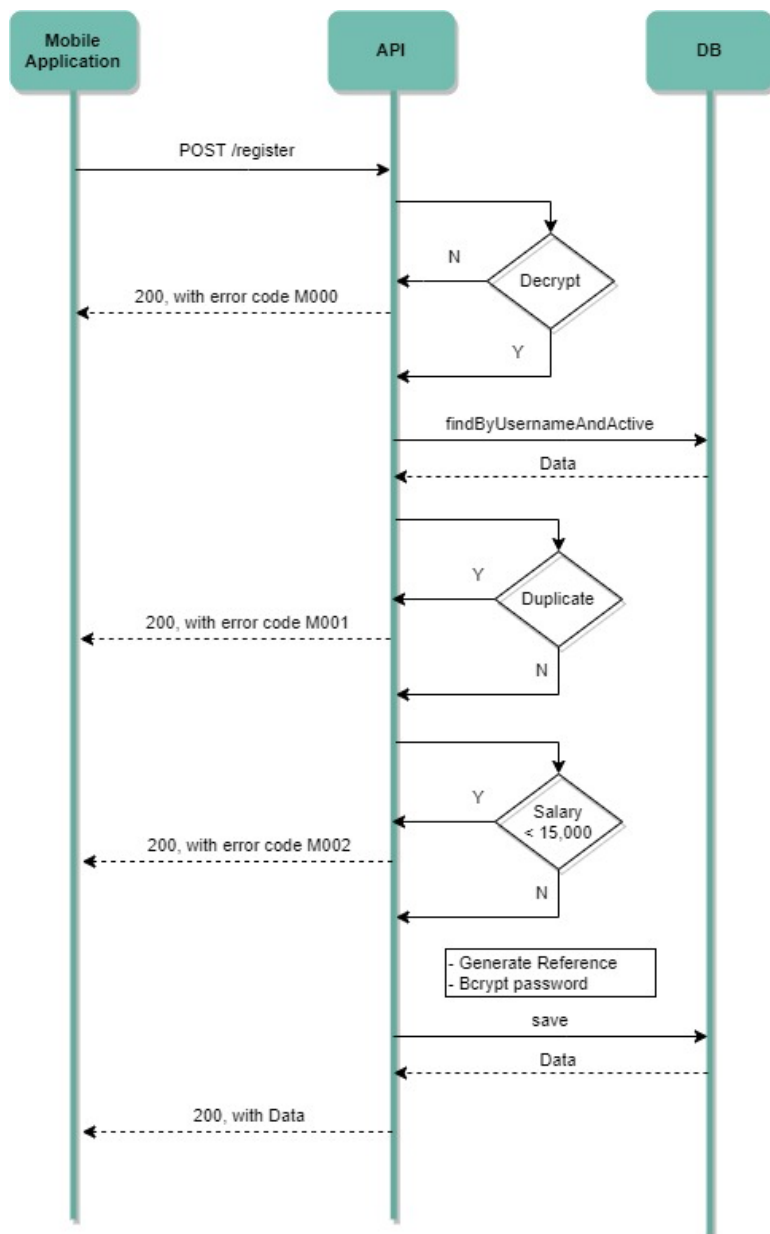
Table of Contents

1. WorkFlow.....	3
1.1 register	3
1.2 getCustomerDetail.....	4
2. API Specification	5
2.1 authentication.....	5
4.1 register	6
4.2 getCustomerDetail.....	8
3. Installation.....	10

1. WorkFlow

1.1 register

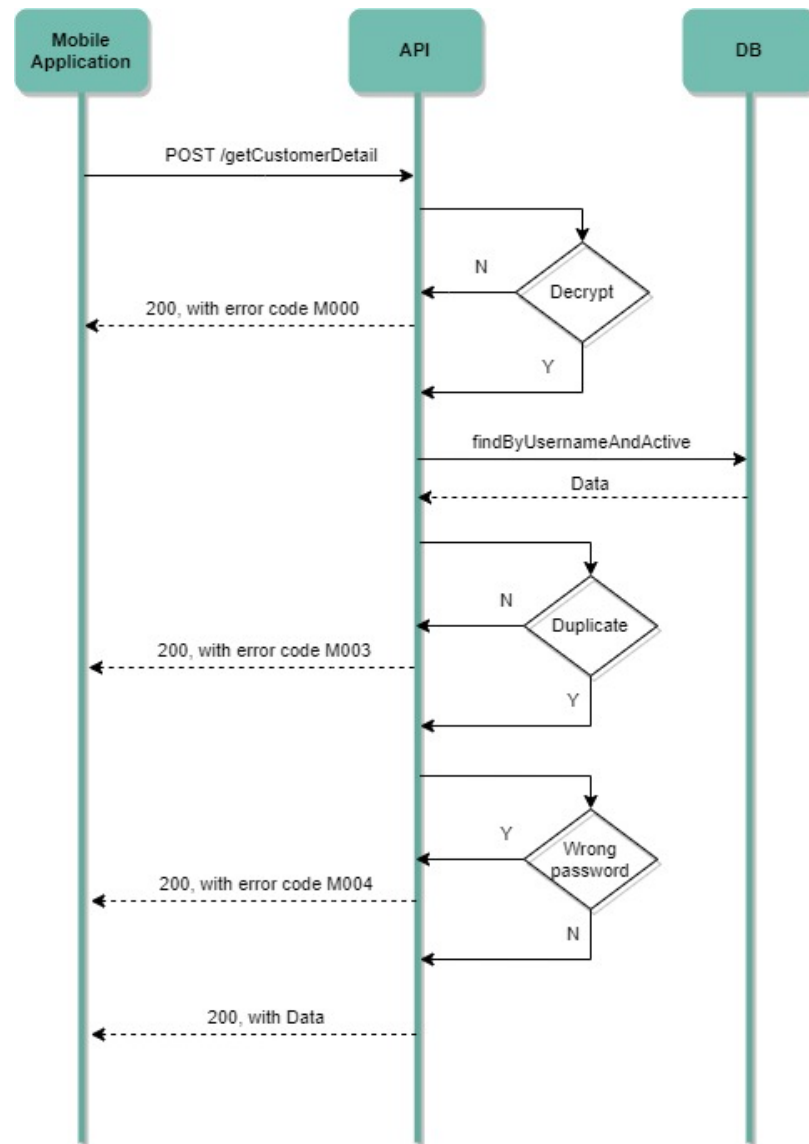
Flow ในการลงทะเบียน เริ่มต้นจาก mobile application เรียก api /register เพื่อสมัครใช้งาน โดยจะต้องนำข้อมูลทุก field มาทำ ASE Encrypt 128 ด้วย secret key ที่ได้ตกลงไว้กับ api และเมื่อ api ได้รับ request จาก mobile application จะทำการ decrypt หากไม่สามารถ decrypt ได้จะ return error code กลับไปให้ mobile application แต่หากสามารถ Decrypt ได้จะดำเนินการตรวจสอบตามเงื่อนไขที่กำหนดไว้ และสมัครใช้งานให้กับผู้ใช้งานในลำดับถัดไป โดยจะสร้าง reference ให้กับลูกค้า (yyyyMMdd<last 4 digit of phone number>) และเข้ารหัส password แบบ Bcrypt เพื่อเก็บข้อมูลลง Database



รูปที่ 1 WorkFlow การลงทะเบียน

1.2 getCustomerDetail

Flow ในการดึงรายละเอียดของผู้ใช้งาน เริ่มต้นจาก mobile application เรียก api /getCustomerDetail โดยจะต้องนำข้อมูลทุก field มาทำ ASE Encrypt 128 ด้วย secret key ที่ได้ตกลงไว้กับ api และเมื่อ api ได้รับ request จาก mobile application จะทำการ decrypt หากไม่สามารถ decrypt ได้จะ return error code กลับไปให้ mobile application แต่หากสามารถ Decrypt ได้จะดำเนินการตรวจสอบตามเงื่อนไขที่กำหนดไว้ และ retrun รายละเอียดผู้ใช้งานให้ตามลำดับ



รูปที่ 2 WorkFlow การดึงรายละเอียดผู้ใช้งาน

2. API Specification

2.1 authenticate

Purpose	เพื่อ Auth เอา Token
Content-Type	application/json
Method	POST
Authorization	-
Endpoint Env. SIT	https://localhost:8000/api/authenticate

Request Message (authenticate)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	key	String	M	CRC32(keyAuthen+yyyyMMDD+salt)	"0514720553"	keyAuthen ค่าที่ได้ตกลงกันไว้ LZ7iyweKfzMEM1iBu9P8JerlZ9g GVgpiEt1fK46SDcVvp7vtCfzxPK0 fqOFHLON
2	slat	String	M	salt	"12345"	

Response Message (authenticate)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	token	String	M	token		นำ token ที่ได้แนบมาใน Header ทุกครั้งที request ***หมายเหตุ Token มีอายุ 3 นาที

Example Request

```
{
  "key": "1853942656",
  "salt": "12345"
}
```

Example Response

Http status	Case	Case Description	Example
200	Success	success	{ "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWUiOiJTZXJ2aWNliwiZXhwIjoxNjMzODQxMTM2LCJpYXQiOiJlZ2MzM4NDA5NTZ9.mCKTQN2JNDLncPLW0gg1r4HrqmV9hdTa4f4WwzEF-H1tRi9XMyL_iYtqB5fY8tMel3pwYRU1T-4b5hRR1JWOA" }
401	Fail	กรณีที่ api CRC32 ด้วย salt แล้วได้ค่าไม่ตรงกับ key	

2.2 register

Purpose	เพื่อลงทะเบียนสมัครใช้งาน
Content-Type	application/json
Method	POST
Authorization	Bearer Token
Endpoint Env. SIT	https://localhost:8000/api/register

Request Message (register)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	data	String	M		"dUk+u/PyKO1gbdQem L2Ce..."	
AES Encrypt 128 all field below and add encrypted value to field data						
1.1	username	String	M	username	"test"	
1.2	password	String	M	password	"aaa"	
1.3	address	String	M	address	"test"	
1.4	phone	String	M	phone	"0901111111"	
1.5	salary	Integer	M	salary	100000	

*** AES Encrypt 128 Mode ECD ด้วย secret key "aesPrivateKey123"

หากต้องการทดสอบยิง api ให้นำ field ทั้งหมดไป encrypt ได้ที่

<https://www.devglan.com/online-tools/aes-encryption-decryption>

Response Message (register)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	api_status	Integer	M	สถานะ	1	1 = สำเร็จ 0 = ไม่สำเร็จ
2	api_message	String	M	ข้อความ	"Success"	ในกรณีที่ api_status = 0 mobile application สามารถนำ api_message ไปแสดงให้ผู้ใช้งานได้เลย
3	data	Object	M	ข้อมูล		ในกรณีที่ api_status = 0 field data จะเท่ากับ null
3.1	reference	String	M	reference	"202110103781"	
3.2	customer_type	String	M	customer_type	"Platinum"	

Example Request

```
{
  "data":
    "dUk+u/PyKO1gbdQemL2CeYeRqPh5Tm1vn2BFbZvqyY2SAjsWNHcL/Ucy7seZBOgxaYaHWkbNuAD2Dd5jPUvIr4
    +Bp0unJ8S9HbVezNs6HmVg7kcXxrTuG4q6SA14Cto18xT/TFy6TdfJQfJ1N30eeR1egoWi0TVqE1ATt0qx8Tw="
}
```

Example Response

Http status	Case	Case Description	Example
200	Success	success	{ "api_status": 1, "api_message": "Success", "data": { "reference": "202110103781", "customer_type": "Platinum" } }
200	Fail	กรณีผู้ใช้งานกรอกเบอร์โทรศัพท์ผิดรูปแบบ	{ "api_status": 0, "api_message": "หมายเลขโทรศัพท์ไม่ถูกต้อง กรุณากรอกใหม่อีกครั้ง [M005]", "data": null }
200	Fail	กรณีมีผู้ใช้งานแล้วในระบบ	{ "api_status": 0, "api_message": "มีผู้ใช้งานแล้วในระบบ [M001]", "data": null }
200	Fail	กรณีมีเงินเดือนต่ำกว่า 15000 บาท	{ "api_status": 0, "api_message": "เงินเดือนต่ำกว่าที่กำหนด [M002]", "data": null }
401	Fail	Bearer Token ไม่ถูกต้องหรือหมดอายุ	{ "timestamp": "2021-10-10T05:17:37.467+00:00", "status": 401, "error": "Unauthorized", "message": "", "path": "/api/register" }

2.3 getCustomerDetail

Purpose	เพื่อดึงรายละเอียดผู้ใช้งาน
Content-Type	application/json
Method	POST
Authorization	Bearer Token
Endpoint Env. SIT	https://localhost:8000/api/getCustomerDetail

Request Message (getCustomerDetail)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	data	String	M		"dUk+u/PyKO1gbdQem L2Ce..."	
AES Encrypt 128 all field below and add encrypted value to field data						
1.1	username	String	M	username	"test"	
1.2	password	String	M	password	"aaa"	

*** AES Encrypt 128 Mode ECD ด้วย secret key "aesPrivateKey123"

หากต้องการทดสอบยิง api ให้นำ field ทั้งหมดไป encrypt ได้ที่

<https://www.devglan.com/online-tools/aes-encryption-decryption>

Response Message (getCustomerDetail)

Seq	Field Name	Type	M/O	Description	Sample Value	Remark
1	api_status	Integer	M	สถานะ	1	1 = สำเร็จ 0 = ไม่สำเร็จ
2	api_message	String	M	ข้อความ	"Success"	ในกรณีที่ api_status = 0 mobile application สามารถนำ api_message ไปแสดงให้ผู้ใช้งานได้เลย
3	data	Object	M	ข้อมูล		ในกรณีที่ api status = 0 field data จะเท่ากับ null
3.1	username	String	M	username	"test"	
3.2	address	String	M	address	"test"	
3.3	phone	String	M	phone	"0901111111"	
3.4	customer_type	String	M	customer_type	"Platinum"	
3.5	create_date	String	M	create_date	"2021-10-10 12:26:25"	

3.6	reference	String	M	reference	"202110103781"	
-----	-----------	--------	---	-----------	----------------	--

Example Request

```
{
  "data":
    "dUk+u/PyKO1gbdQemL2CeYeRqPh5Tm1vn2BFbZvqyY3Y0UiWtEdiDOSRLAC1VOU12gwQDoOZaLstTPtdso499w
  =="
```

Example Response

Http status	Case	Case Description	Example
200	Success	success	{ "api_status": 1, "api_message": "Success", "data": { "username": "aaa6", "address": "", "phone": "0907363781", "customer_type": "Platinum", "create_date": "2021-10-10 12:26:25.0", "reference": "202110103781" } }
200	Fail	กรณีไม่พบผู้ใช้งาน	{ "api_status": 0, "api_message": "ไม่พบผู้ใช้งาน [M003]", "data": null }
200	Fail	กรณี Password ไม่ถูกต้อง	{ "api_status": 0, "api_message": "รหัสผ่านไม่ถูกต้อง [M004]", "data": null }
401	Fail	Bearer Token ไม่ถูกต้องหรือหมดอายุ	{ "timestamp": "2021-10-10T05:17:37.467+00:00", "status": 401, "error": "Unauthorized", "message": "", "path": "/api/register" }




3. Installation

3.1 clone repo <https://github.com/phasathon/BackendTest.git>

3.2 จากนั้น run “mvn package” เพื่อ build ไฟล์ .jar

3.3 สร้าง folder ใหม่ที่ได้ไม่ได้อยู่ใน repo และนำไฟล์ backend-0.0.1.jar และ start.sh ไปวาง

3.4 สร้าง folder config ใน folder ข้อ 3.3 และนำไฟล์ errorMapping.json ไปวาง

 config	10/10/2564 13:30	File folder	
 backend-0.0.1.jar	9/10/2564 22:52	Executable Jar File	42,797 KB
 start.sh	10/10/2564 13:33	Shell Script	1 KB

3.5 เปิดไฟล์ backend-0.0.1.jar (ด้วยโปรแกรม WinRAR) จากนั้นเปิดไฟล์
BOOT-INF/classes/application.yml เพื่อแก้ไข database connection จากนั้นกด Save

3.6 run คำสั่ง sh start.sh เพื่อ start program port 8000

```

10/10/2021 13:35:21 /drives/c/Users/ /eclipse-workspace/backend/target P master sh start.sh
10/10/2021 13:35:24 /drives/c/Users/ /eclipse-workspace/backend/target P master

Spring
:: Spring Boot ::
(v2.4.1)

2564-10-10 13:35:28.497 INFO 65060 --- [main] com.backend.test.BackendApplication : Starting BackendApplication v0.0.1 using Java 1.8.0_261 on LAPTOP-B0I9VJNH with PID 65060
(cc\Users\GHB-BIG\workspace\backend\target\backend-0.0.1.jar started by GHB-BIG in C:\Users\GHB-BIG\workspace\backend\target)
2564-10-10 13:35:28.511 INFO 65060 --- [main] com.backend.test.BackendApplication : No active profile set, falling back to default profiles: default
2564-10-10 13:35:31.961 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data modules found, entering strict repository configuration mode!
2564-10-10 13:35:31.974 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JDBC repositories in DEFAULT mode.
2564-10-10 13:35:32.187 INFO 65060 --- [main] .RepositoryConfigurationExtensionSupport : Spring Data JDBC - Could not safely identify store assignment for repository candidate interface com.backend.test.repository.BackendRepository. If you want this repository to be a JDBC repository, consider annotating your entities with one of these annotations: org.springframework
.data.relational.core.mapping.Table
2564-10-10 13:35:32.191 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 194 ms. Found 0 JDBC repository interfaces.
2564-10-10 13:35:32.235 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data modules found, entering strict repository configuration mode!
2564-10-10 13:35:32.241 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2564-10-10 13:35:32.449 INFO 65060 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 188 ms. Found 1 JPA repository interfaces.
2564-10-10 13:35:33.829 INFO 65060 --- [main] trationDelegatesBeanPostProcessorChecker : Bean 'org.springframework.ws.config.annotation.DelegatingWsConfiguration' of type [org.spr
ingframework.ws.config.annotation.DelegatingWsConfiguration$SEnhanerBySpringGLIBS$4efb737a] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for au
to-proxying)
2564-10-10 13:35:34.041 INFO 65060 --- [main] .w.s.a.s.AnnotationActionEndpointMapping : Supporting [WS-Addressing August 2004, WS-Addressing 1.0]
2564-10-10 13:35:34.208 INFO 65060 --- [main] trationDelegatesBeanPostProcessorChecker : Bean 'org.springframework.security.access.expression.method.DefaultMethodSecurityExpresio
nHandler$552768b' of type [org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler] is not eligible for getting processed by all BeanPostProcessors (for
examples: not eligible for auto-proxying)
2564-10-10 13:35:34.219 INFO 65060 --- [main] trationDelegatesBeanPostProcessorChecker : Bean 'methodSecurityMetadataSource' of type [org.springframework.security.access.method.De
legatingMethodSecurityMetadataSource] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2564-10-10 13:35:35.583 INFO 65060 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8000 (http)
2564-10-10 13:35:35.924 INFO 65060 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2564-10-10 13:35:35.935 INFO 65060 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.41]
2564-10-10 13:35:36.243 INFO 65060 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2564-10-10 13:35:36.244 INFO 65060 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 7500 ms
2564-10-10 13:35:36.971 INFO 65060 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2564-10-10 13:35:38.886 INFO 65060 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2564-10-10 13:35:39.082 INFO 65060 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH0000204: Processing PersistenceUnitInfo [name: default]
2564-10-10 13:35:39.408 INFO 65060 --- [main] org.hibernate.Version : HHH0000412: Hibernate ORM core version 5.4.25.Final
2564-10-10 13:35:39.906 INFO 65060 --- [main] o.hibernate.annotations.common.Version : HCHA0000080: Hibernate Commons Annotations (5.1.2.Final)
2564-10-10 13:35:40.882 INFO 65060 --- [main] o.hibernate.dialect.Dialect : HHH0004080: Using dialect: org.hibernate.dialect.SQLServer2012Dialect
2564-10-10 13:35:43.545 INFO 65060 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH0004090: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2564-10-10 13:35:43.572 INFO 65060 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2564-10-10 13:35:43.800 WARN 65060 --- [main] jpaBaseConfigurationsJpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performe
d during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2564-10-10 13:35:45.358 INFO 65060 --- [main] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.context.request.async.WebAs
yncManagerIntegrationFilter$63d102a2, org.springframework.security.web.context.SecurityContextPersistenceFilter$6d23017e, org.springframework.security.web.header.HeaderWriterFilter$6e746f57,
org.springframework.security.web.authentication.logout.LogoutFilter$2066604, com.backend.test.config.JdbcRequestFilter$729991e, org.springframework.security.web.savedrequest.RequestCacheAware
Filter$1817f1eb, org.springframework.security.web.servletapi.SecurityContextHolderRequestFilter$301ec38b, org.springframework.security.web.authentication.AnonymousAuthenticationFilter$60
c45ee6e, org.springframework.security.web.session.SessionManagementFilter$7f36662c, org.springframework.security.web.access.ExceptionTranslationFilter$1f6c9cd8, org.springframework.security.w
eb.access.intercept.FilterSecurityInterceptor$3016f5d6]
2564-10-10 13:35:45.804 INFO 65060 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2564-10-10 13:35:46.769 INFO 65060 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8000 (http) with context path ''
2564-10-10 13:35:46.790 INFO 65060 --- [main] com.backend.test.BackendApplication : Started BackendApplication in 20.117 seconds (JVM running for 22.095)

```

3.6 create table in sql server

```

CREATE TABLE [dbo].[tb_customer_data](
[id] [int] IDENTITY(1,1) NOT NULL,
[reference] [nvarchar](50) NOT NULL,
[username] [varchar](50) NOT NULL,
[password] [nvarchar](100) NOT NULL,
[address] [nvarchar](max) NOT NULL,
[phone] [varchar](10) NOT NULL,
[salary] [int] NOT NULL,
[customer_type] [varchar](20) NOT NULL,
[active] [tinyint] NOT NULL,
[create_date] [datetime] NOT NULL,
[update_date] [datetime] NULL
)

```