



НЭМЭГДСЭН ӨРТГИЙН
АЛБАН ТАТВАРЫН
СИСТЕМ



ТАТВАРЫН ЕРӨНХИЙ ГАЗАР **POS API** ГАРЫН АВЛАГА





Өөрчлөлт хийсэн түүх

Огноо	Албан тушаалтан	Тайлбар
2015-12-01	Б.Насанжаргал, Д.Очирпүрэв, С.Баттулга	PosAPI 2.1 Гарын авлагыг анхны байдлаар бичсэн
2015-12-07	Б.Насанжаргал, Д.Очирпүрэв, С.Баттулга	PosAPI 2.1 талбарын өөрчлөлт
2015-12-09	Б.Насанжаргал, Д.Очирпүрэв, С.Баттулга	Баримт засварлах тухай



Гарчиг

Танилцуулга	2
Цахим төлбөрийн баримт.....	5
PosAPI-г холбох заавар	7
PosAPI функцүүд.....	10
UString өгөгдлийн төрөл	10
Ustring PosAPI::checkAPI().....	10
Ustring PosAPI::getInformation()	11
Ustring PosAPI::callFunction(Ustring, Ustring).....	11
Ustring PosAPI::put(Ustring)	12
Ustring PosAPI::returnBill(Ustring).....	15
Ustring PosAPI::sendData()	16
Алдааны мэдээллийн тайлбар.....	18
Java програмчлалын хэлтэй холбох жишээ	20
C# .Net програмчлалын хэлтэй холбох жишээ	30



Танилцуулга

“Нэмэгдсэн Өртгийн Албан Татварын Тухай Хууль”-ийн шинэчилсэн найруулга 2015 оны 07 дугаар сарын 09 нд соёрхон батлагдсантай холбогдуулан бараа, ажил үйлчилгээний борлуулалтын баримтын мэдээллийг цахимаар цуглуулах зорилгоор Татварын Ерөнхий Газраас PosAPI санг хөгжүүлэн гаргалаа.

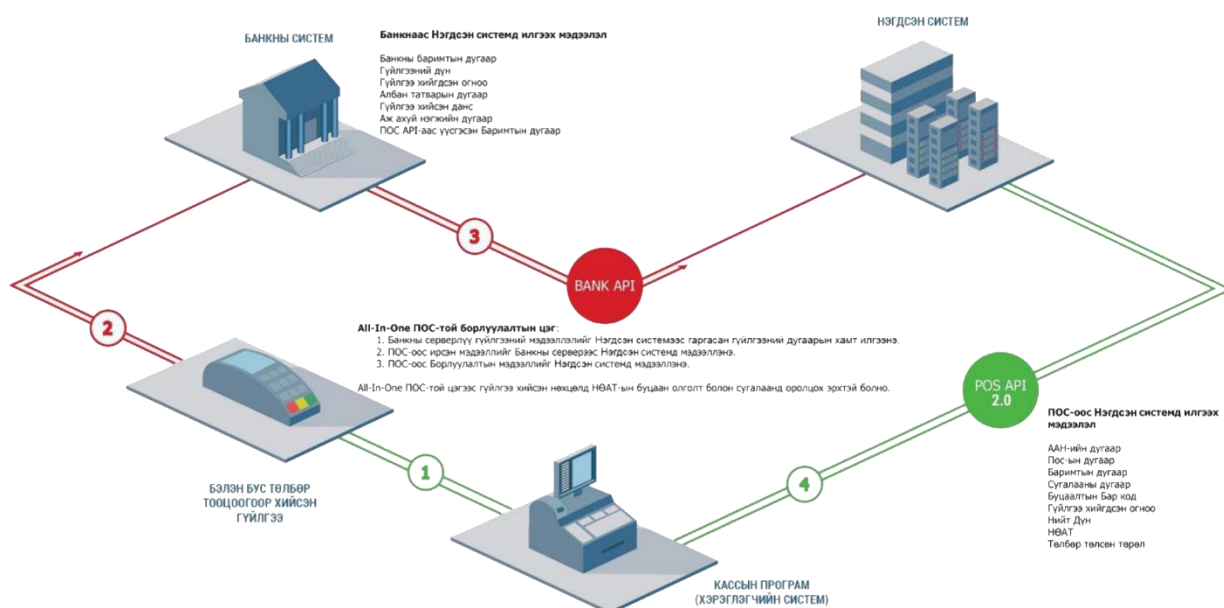
PosAPI нь татвар суутган төлөгч ААН, иргэнээс эцсийн хэрэглэгчид худалдаалагдаж буй бараа, ажил үйлчилгээний мэдээллийг татварын нэгдсэн системд цуглуулах хэрэглэгчийн системтэй уялдан ажиллах програм хангамжийн модуль юм.

Ажиллах зарчим

PosAPI-н ажиллах зарчим дараах байдалтай байна.

Хамрах хүрээ

1. НӨАТ суутган төлөгч
2. НХАТ суутган төлөгч



Зураг 1

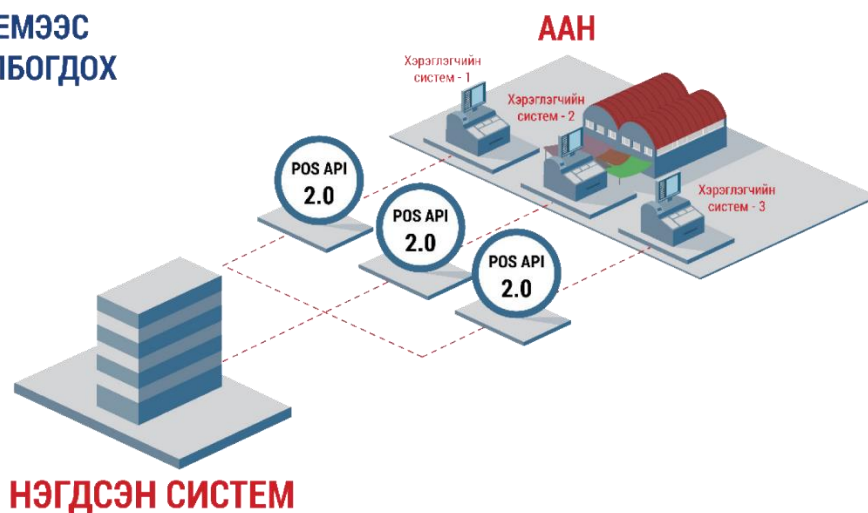


Ажиллах горим

PosAPI санг хэрэглэгчийн системээс нэгдсэн системд холбогдох горим болон төв серверээс нэгдсэн системд холбогдох горим гэсэн 2 горимоор ашиглаж болно.

1. Хэрэглэгчийн системээс нэгдсэн системд холбогдох горим гэдэг нь PosAPI санг хэрэглэгчийн систем бүрт зориулан татан авч суурилуулан ашиглах юм.

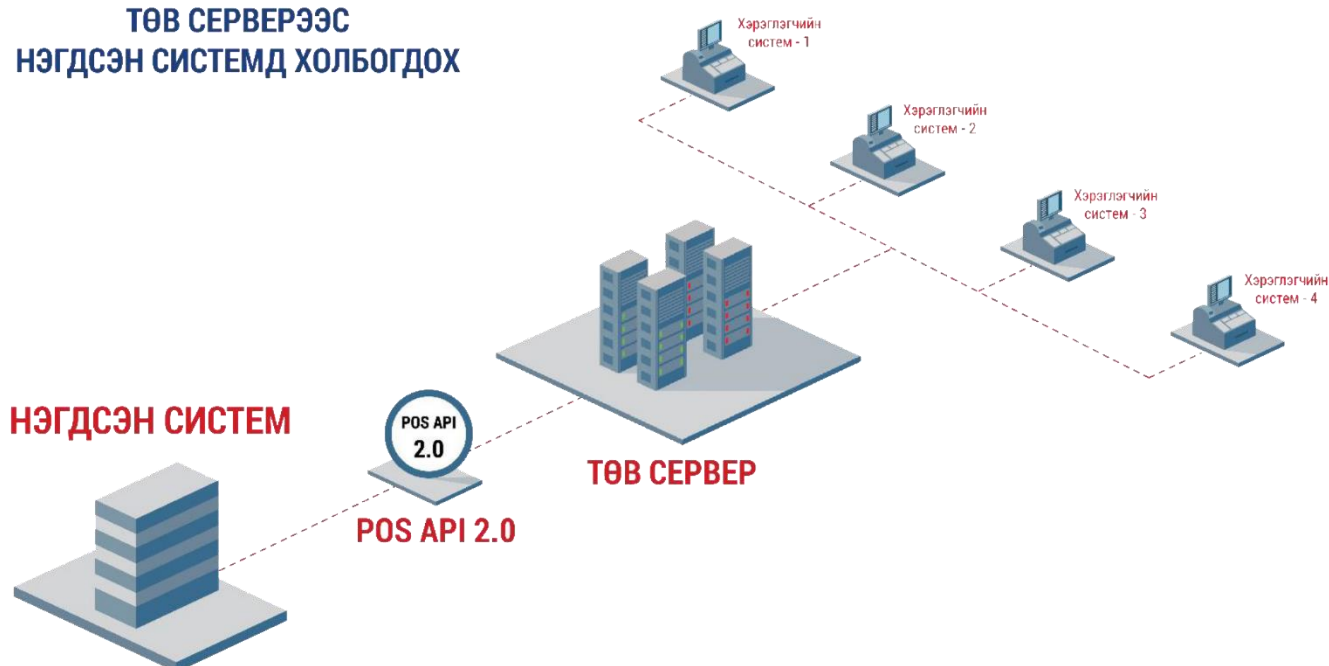
ХЭРЭГЛЭГЧИЙН СИСТЕМЭЭС НЭГДСЭН СИСТЕМД ХОЛБОГДОХ



Зураг 2

2. Төв серверээс нэгдсэн системд холбогдох горим гэдэг нь PosAPI санг төв сервер дээр суурьлуулан олон хэрэглэгчийн системийг холбон ашиглах юм

ТӨВ СЕРВЕРЭЭС НЭГДСЭН СИСТЕМД ХОЛБОГДОХ

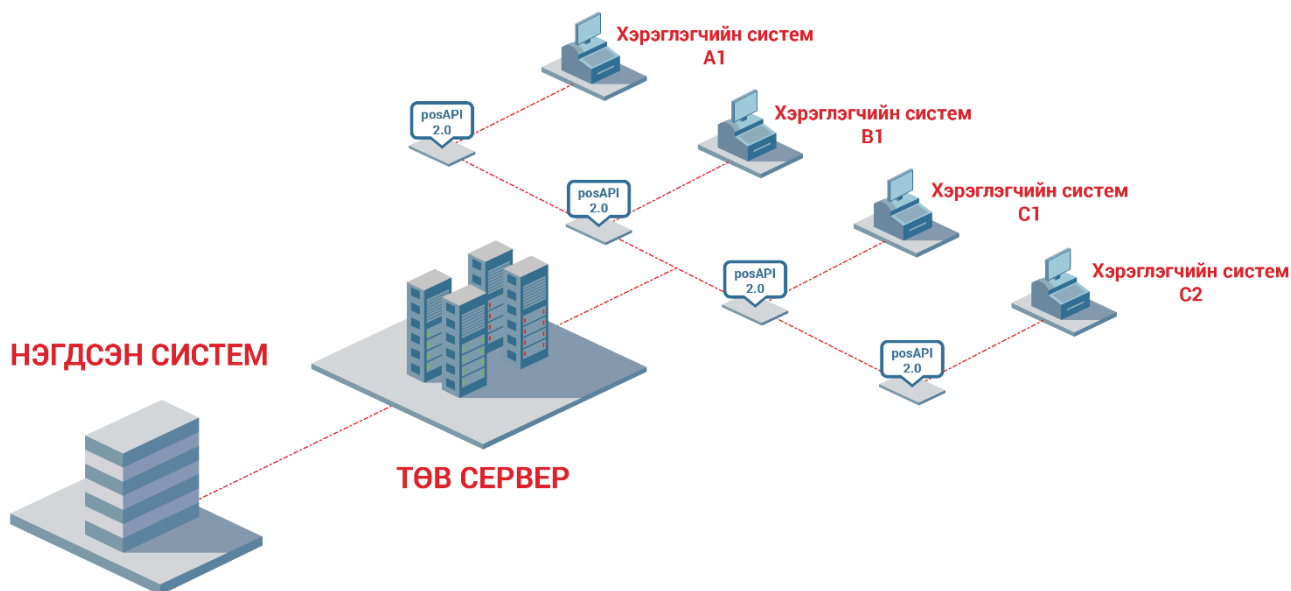


Зураг 3



Мөн төв серверээс хэд хэдэн ААН, иргэний хэрэглэгчийн системийг нэгдсэн системд холбох боломжтой юм.

ХУДАЛДАА ҮЙЛЧИЛГЭЭ БОРЛУУЛАГЧ ОЛОН БАЙГУУЛЛАГА ДУНДЫН ТӨВ СЕРВЕРЭЭС НЭГДСЭН СИСТЕМД ХОЛБОГДОХ.



Зураг 4

Төв серверээс хэрэглэгчийн системийг нэгдсэн системд холбох үед сүлжээ доголдоход сугалаа болон дахин давтагдашгүй төлбөрийн дугаар /ДДТД/ зэрэг мэдээллүүдийг хэрэглэгчийн системд үүсгэж өгөх процесс зогсоно гэдгийг анхаарна уу!!!

Техникийн үзүүлэлт

Pos API ажиллах системийн хамгийн бага үзүүлэлт дараах байдалтай байна.

Процессор /Processor/
Санах ой /RAM/
диск /HARD disk/
График үзүүлэлт

900 MHz болон түүнээс дээш үзүүлэлттэй байх
512 MB болон түүнээс дээш үзүүлэлттэй байх Хатуу
50GB сул зай байх
Direct-X 9 дээш үзүүлэлттэй



Цахим төлбөрийн баримт

НӨАТ-ын тухайн хуулинд цахим төлбөрийн баримтыг үндэслэн сугалаа болон НӨАТ-н 20% хүртэлх урамшуулал олгох, НӨАТ-падааныг цахим төлбөрийн баримтыг үндэслэн үйлдэнэ гэж тусгагдсан. Иймд PosAPI нь цахим төлбөрийн баримтыг “Байгууллагаас иргэнд /B2C/”, “Байгууллагаас байгууллагад /B2B/” гэж 2 ялгаатай төрлөөр бүртгэж тухай бүр тохирсон өгөгдлийг буцаана.

ИРГЭНД ОЧИХ БАРИМТ				
Сугалаатай				
Борлуулагч нь НӨАТ төлөгч мөн бол 20% урамшуулалтай				
Борлуулагч нь НӨАТ төлөгч биш бол 20% урамшуулалгүй				
Таван Буянт Трейд				
Борлуулагч:				
ТТД:	2584727			
ДДТД:	000002584727151027121518239828			
Огноо:	2015/10/27 11:03:51			
Касс:	№ 001			
Кассчин:	№ 14525			
Бараа				
Т/Ш	НӨАТ	НХАТ	НЭГЖ ҮНЭ	НИЙТ
Атар талх				
1	109.09	0.00	1200.00	1200.00
Сүү /Сүү ХХК/				
1	254.55	0.00	2800.00	2800.00
<u>ЭССЭ Тамхи</u>				
1	315.32	31.53	3500.00	3500.00
Пирог самартай				
2	640.00	0.00	3520.00	7040.00
Бонус:	0.00	Нийт үнэ:		14540.00
НӨАТ:	1318.95	НХАТ:		31.53
Бүгд үнэ:	14540.00			
Бэлэн бусаар:	4540.00	Бэлнээр:		10000.00
Нийт төлсөн:	14540.00	Хариулт:		0.00
Card No	RRN	App.Code	Terminal ID	Amount
9496 25** **** 5656	000249619875	245517	91110578	2000.00
9497 25** **** 7819	000754523545	325485	91110578	2540.00
CE 59394258				
000002584727151027121518239828				

БАЙГУУЛЛАГАД ОЧИХ БАРИМТ				
Сугалаа болон 20% урамшуулал олгохгүй				
Таван Буянт Трейд				
Борлуулагч:				
ТТД:	2584727			
ДДТД:	000002584727151027121518239828			
Огноо:	2015/10/27 11:03:51			
Касс:	№ 001			
Кассчин:	№ 14525			
Худалдан авагч:				
ТТД:	5254587			
Нэр:	Харуул Алтай СӨХ			
Бараа				
Т/Ш	НӨАТ	НХАТ	НЭГЖ ҮНЭ	НИЙТ
Атар талх				
1	109.09	0.00	1200.00	1200.00
Сүү /Сүү ХХК/				
1	254.55	0.00	2800.00	2800.00
Бонус:	0.00	Нийт үнэ:		4000.00
НӨАТ:	363.64	НХАТ:		0.00
Бүгд үнэ:	4000.00			
Бэлэн бусаар:	0.00	Бэлнээр:		4000.00
Нийт төлсөн:	10000.00	Хариулт:		6000.00
000002584727151027121518239828				

Уг баримтын загвар зураг нь зөвхөн жишээ зураг бөгөөд ААН-н цахим төлбөрийн баримт нь уг загварыг заавал хуулбарлах шаардлагагүйг анхаарна уу!!!



Төлбөрийн баримтын шаардлага

ААН-ийн цахим төлбөрийн баримтын загвар нь Монгол Улсын Стандартчилал хэмжил зүйн газраас баталсан “Татварын нэгдсэн системд ашиглах төлбөрийн баримтын ерөнхий шаардлага” /MNS 5005:2015/ стандартыг мөрдлөг болгоно.

Шаардлагатай талбарууд

Байршил	Талбар	Иргэнд	Байгууллагад
Баримтын дээд хэсэг	Борлуулагчийн ТТД	Тийм	Тийм
	Борлуулагчийн нэр	Тийм	Тийм
	Худалдан авагчийн ТТД	Үгүй	Тийм
	Худалдан авагчийн нэр	Үгүй	Тийм
	Дахин давтагдашгүй төлбөрийн дугаар	Тийм	Тийм
	Огноо	Тийм	Тийм
Баримтын доод хэсэг	Сугалааны дугаар	Тийм	Үгүй
	Qr Code	Тийм	Тийм
	Буцаалтын BarCode	Үгүй	Үгүй

Буцаалтын BarCode нь хэдийгээр заавал хэвлэх шаардлаггүй ч төлбөрийн баримтыг хүчингүй болгохдоо ДДТД нь 33 оронтой тоон утгыг ашиглах ба гараар бичих явцад алдах эрсдэл байгааг бодолцоно уу!!!

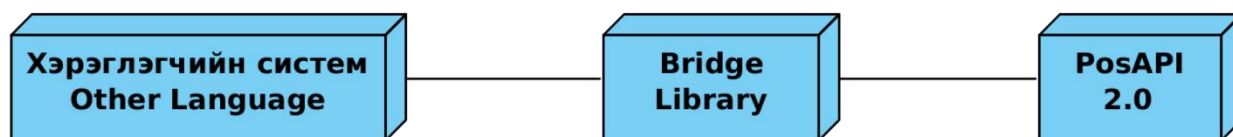
PosAPI-г холбох заавар

Холбох аргачлал

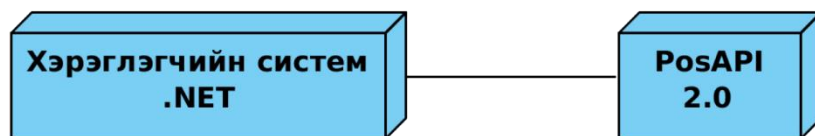
PosAPI 2.1 сан нь C++ хэл дээр бичигдсэн бөгөөд хэрэглэгчийн систем нь уг сантай холбогдоход PosAPI.h болон ExportLib.h header файлуудын тусламжтайгаар холбогдоно.

C++ хэл нь бусад програмчлалын хэлтэй шүүд холбогдох бүрэн боломжтой байдаг бөгөөд холбох үедээ дундын bridge library/зургаг 6/ үүсгэх шаардлага тулгардаг. Хэрэв .NET framework эсвэл Delphi гэх мэт windows үйлдлийн системийн орчинд ажилладаг програмчлалын хэл бол заавал дундын санг ашиглалгүй шүүд холболтыг/зургаг 7/ хийж болно.

Уг холболтуудын талаар энэ гарын авлагын төгсгөлд Java болон .NET C# програмчлалын хэлнүүд дээр жишээ аван тайлбарласан тул уг жишээнээс илүү дэлгэрүүлэн ойлгоно гэдэгт итгэлтэй байна.



Зургаг 6



Зургаг 7

Суурьлуулах, тохируулах

PosAPI 2.1 санг хөгжүүлэхдээ Qt5 болон OpenSSL 1.0 технологиудыг ашигласан тул холболт хийхэд уг сангуудаас тодорхой хэсэгүүд хэрэгцээтэй байдаг. Тухайн сангуудыг <https://ebarimt.mn/posApi> гэсэн хаягаар орон татан авч болно.

Windows үйлдлийн системд тохируулах

Шаардлагатай сангууд

icudt53.dll
icuin53.dll
icuuc53.dll
Qt5Core.dll
Qt5Network.dll
Qt5Script.dll
Qt5Sql.dll
libeay32.dll



ssleay32.dll
sqldrivers/qsqli.dll

Суулгах заавар

Дээрх DLL файлуудыг тухайн хэрэглэгчийн системийн “Working directory”-д хуулна. Мөн Java програмчиллын хэл дээр хөгжүүлэгдсэн хэрэглэгчийн системийн хувьд “C:\Windows\Sun\Java\Bin” directory-д хуулж болно. Мөн “C:\Windows\Sun\Java\Bin” directory нь үүсээгүй байж болох бөгөөд шаардлагатай гэж үзвэл гараар үүсгэж болно гэдгийг анхаарна уу.

Мөн Windows орчинд PosAPI 2.1 санг ажиллуулахдаа “Visual C++ Redistributable Package for Visual Studio 2013” програмыг заавал суулгасан байна. Учир нь PosAPI.dll болон ашиглагдаж буй нэмэлт сангууд нь Visual Studio 2013 C++ хэл ашиглан бичигдсэн байдаг тул заавал суусан байх ёстой.

JAVA програмчиллын хэл дээр хөгжүүлэгдсэн хэрэглэгчийн систем нь Windows орчинд ажиллуулах бол OpenSSL v1.0.2d Light заавал суулгасан байна. Windows XP орчинд ажиллах хэрэглэгчийн системийн хувьд OpenSSL v1.0.2d Light заавал суулгасан байна.

Мөн Windows XP SP3 байх шаардлагатай.

OpenSSL татах хаяг: <https://slproweb.com/products/Win32OpenSSL.html>

Жишээ нь:

Working Directory-д хуулсан хэлбэр

Name	Date modified	Type	Size
sqldrivers	2015/11/29 15:02	File folder	
DemoPos.exe	2015/11/29 07:36	Application	25 KB
icudt53.dll	2014/9/3 16:42	Application extens...	21,025 KB
icuin53.dll	2014/9/3 16:42	Application extens...	2,412 KB
icuuc53.dll	2014/9/3 16:42	Application extens...	1,675 KB
libeay32.dll	2015/11/29 21:21	Application extens...	1,611 KB
PosAPI.dll	2015/11/29 14:59	Application extens...	116 KB
Qt5Core.dll	2015/8/21 16:29	Application extens...	4,825 KB
Qt5Network.dll	2015/5/31 02:24	Application extens...	1,017 KB
Qt5Script.dll	2015/5/31 05:39	Application extens...	1,218 KB
Qt5Sql.dll	2015/5/31 02:24	Application extens...	196 KB
ssleay32.dll	2015/11/29 21:21	Application extens...	335 KB

Зураг 8

“Зураг 8” зургийн хувьд DemoPos.exe файл нь хэрэглэгчийн системийн ажлын файл гэж үзвэл уг файлыг агуулж буй directory-г Working Directory гэж нэрлэнэ. Уг хавтасанд шаардлагатай DLL сангуудыг хуулснаар ажлын файл шууд тухайн сангуудыг хайж олох, ашиглах боломжтой болдог.



Linux үйлдлийн системд тохируулах

Шаардлагатай сангууд

libcudata.so.53
libcuduc.so.53
libcui18n.so.53
libQt5Core.so.5.4.1
libQt5Network.so.5.4.1
libQt5Script.so.5.4.1
libQt5Sql.so.5.4.1
libcrypto.so.1.0.0
libssl.so.1.0.0

Суулгах заавар

Linux үйлдлийн системийн хувьд дээрх сангууд болон libPosAPI.so сан буюу PosAPI-г цугт нь /usr/lib гэсэн directory-д хуулж өгнө. Учир нь тухайн хавтас нь Linux үйлдлийн системийн сан хайх үндсэн directory юм. Харин OpenSSL сан буюу libcrypto.so.1.0.0, libssl.so.1.0.0 сангуудыг ерөнхийдөө бүх Linux үйлдлийн системийн хувьд суусан байдаг тул заавал суулгах шаардлагагүй.

Нэмэлт тайлбар

PosAPI-н тухайн байгууллагын хэрэглэгчийн систем бүрд өөр өөр хэлбэрээр ажиллахаар загварчлагдсан байдаг. Иймд нэг PosAPI-г олон хэрэглэгчийн системд суурьлуулан хэрэглэх нь тохиромжгүйгээс гадна ДДТД давхцаж үүсэх эрсдлийг буй болгоно.



PosAPI функцүүд

PosAPI-ийн бүх функцүүд нь JSON string ашиглан өгөгдөл хүлээн авах болон буцаадаг. Энэ нь програмууд болон програмчлалын хэл хооронд өгөгдөл дамжуулхад өгөгдлийн төрөл болон тоон утгын хувьд гарах алдааг арилгаж буй юм. Мөн JSON формат нь нийтэд тархсан бүх програмчлалын хэлнүүд дэмжин ажиллах боломжтой байдаг нь JSON сонгох хамгийн том шалтгаан юм.

UString өгөгдлийн төрөл

Windows болон Linux үйлдлийн систем нь Unicode тэмдэгтэй өөр өөр хэлбэрээр ажилладаг тул тэмдэгтийг хүлээн авахдаа ялгаатай ажиллах шаардлагатай. "UString" төрөл нь Windows орчинд std::wstring харин Linux орчинд std::string төрөл рүү хөрвөж ажиллана.

Ustring PosAPI::checkAPI()

Хэрэглэгчийн системийн тогтвортой ажиллагааг хангах шаардлагын улмаас PosAPI сангийн ажиллагааг шалгана. Хэрэглэгчийн системийг ажиллуулж буй үйлдлийн системийн хэрэглэгч нь заавал өөрийн HOME directory-той байх ёстой. Хэрэв уг шаардлагыг хангаагүй бол уг функц нь амжилтгүй гэсэн утгыг буцаана.

Оролтын утга

Байхгүй

Гаралтын утга

Гаралтын утгын формат

```
{
  "success":boolean,
  "database":{
    "success":boolean,
    "message":String
  },
  "config":{
    "success":boolean,
    "message":String
  },
  "network":{
    "success":boolean,
    "message":String
  }
}
```

Гаралтын утгын тайлбар

Нэр	Тайлбар
success	PosAPI сан нь хэвийн ажиллах эсэхийг илтгэнэ <i>true - ажиллагаа хэвийн</i> <i>false - ажиллагаа хэвийн бус</i>
database	Өгөгдөлийн баазтай харьцахад саад буй эсэхийг тодорхойлно.
success	<i>true - ажиллагаа хэвийн</i> <i>false - ажиллагаа хэвийн бус</i>
message	Хэрэв алдаатай гэж үзвэл алдааны мессежийг буцаана



config	Тохиргооны мэдээллүүдийг серверээс татаж тохируулсан эсэхийг тодорхойлоно.
success	<i>true - ажиллагаа хэвийн false - ажиллагаа хэвийн бус</i>
message	Хэрэв алдаатай гэж үзвэл алдааны мессежийг буцаана
network	Сүлжээ болон интернет холболт хэвийн байгааг татварын нэгдсэн системрүү холбогдож шалгана.
success	<i>true - ажиллагаа хэвийн false - ажиллагаа хэвийн бус</i>
message	Хэрэв алдаатай гэж үзвэл алдааны мессежийг буцаана

Ustring PosAPI::getInformation()

Хэрэглэгчийн систем нь нэгээс олон PosAPI ашиглаж буй үед харьцаж буй PosAPI-гийн мэдээллийг харах шаардлага тулгардаг. Уг функц нь уг асуудлыг шийдэж буй бөгөөд уг функцийг тусламжтайгаар хэрэглэгчийн систем нь тухайн ашиглаж буй PosAPI-гийн дотоод мэдээллүүдийг авна.

Оролтын утга

Байхгүй

Гаралтын утга

Гаралтын утгын формат

```
{
  "registerNo":String,
  "branchNo":String,
  "posId":String,
  "dbDirPath":String,
  "extraInfo":{
    "countBill": String
  }
}
```

Гаралтын утгын тайлбар

Нэр	Тайлбар
registerNo	PosAPI ашиглаж буй татвар суутган төлөгчийн дугаар. ААН бол 7 оронтой тоо, иргэн бол 12 оронтой тоо /2 серийн үсэгийг тоон утгаруу хөрвүүлсэн байдлаар/
branchNo	PosAPI ашиглаж буй татвар суутган төлөгчийн салбарын дугаар. 3 оронтой тоон утгаар /000, 001, 142 гэх мэт/
posId	PosAPI-г системд бүртгэсэн дугаар.
dbDirPath	PosAPI-гийн ашиглаж буй SQLite өгөгдлийн баазыг агуулж буй directory-гийн байршил
extraInfo	PosAPI-аас нэмэлтээр өгөх тайлбар, мэдээллүүд байна
countBill	Илгээгдээгүй үлдсэн баримтын тоо

Ustring PosAPI::callFunction(Ustring,Ustring)

PosAPI нь цаашид шинээр нэмэлт функцүүд нэмэгдэх бөгөөд нэмэгдсэн функцийг ашиглахын тулд заавал өөрийн PosAPI-г шинээр татах шаардлаггүй юм. Уг нэмэлт функцүүдийг уг функцийг

тусламжтайгаар дуудана.

Уг өөрчлөлтийг тухай бүр өөрсдийн цахим хуудас болон Уг гарын авлагын шинэчлэлээр зарлана.

Оролтын утга

1-рх оролтын утга:

Дуудах функцийн нэр

Оролтын формат

Тэмдэгт мөр /Нэмэгдсэн функцийн нэр байна/

2-рх оролтын утга:

Дамжуулах параметер

Оролтын формат

JSON String /Тухайн функцээс хамаарч өөр өөр байна/

Гаралтын утга

Гаралтын утгын формат

JSON String /Тухайн функцээс хамаарч өөр өөр байна/

Ustring PosAPI::put(Ustring)

Хэрэглэгчийн системээс борлуулалтын мэдээллийг хүлээн авч буцаан баримтын ДДТД, сугалааны дугаар, баримт хэвлсэн огноо, баримтын код, QRCode гэсэн утгуудыг нэмж боловсруулан буцаана.

Эдгээр буцаасан мэдээллүүдээс зөвхөн ДДТД болон төлбөрийн баримт хэвлсэн огноо гэсэн 2 талбараас бусад талбаруудын мэдээллийг хэрэглэгчийн системд хадгалахыг хориглоно. Мөн баримт бүртгэхийн тулд тухайн бүртгэлийн машин нь заавал сүлжээний төхөөрөмжтэй /LAN card/ байх ёстой. Хэрэв сүлжээний төхөөрөмжгүй эсвэл сүлжээний төхөөрөмж нь идэвхигүй /disabled/ горимтой бол баримт хэвлэх боломжгүйг анхаарна уу!!!

Сүлжээнд холбоогүй байж болно /Disconnected/

Оролтын утга

Оролтын утгын формат

```
{  
  "amount": String,  
  "vat": String,  
  "cashAmount": String,  
  "nonCashAmount": String,  
  "cityTax": String,  
  "districtCode": String,  
  "posNo": String,  
  "customerNo": String,  
  "billType": String,  
  "billIdSuffix": String,  
  "returnBillId": String,  
  "stocks": [  
    ...  
    {  
      "code": String,  
      "name": String,  
    }  
  ]  
}
```



```

        "measureUnit": String,
        "qty": String,
        "unitPrice": String,
        "totalAmount": String,
        "cityTax": String,
        "vat": String,
        "barCode": String
    }
    ...
],
"bankTransactions": [
    ...
    {
        "rrn": String,
        "bankId": String,
        "terminalId": String,
        "approvalCode": String,
        "amount": String
    }
    ...
]
}

```

Оролтын утгын тайлбар

Нэр	Нөхцөл	Тайлбар
amount	2 орны нарийвчлалтай тоо	Баримтын гүйлгээний нийт дүн
vat	2 орны нарийвчлалтай тоо	Баримтын НӨАТ-ын нийт дүн
cashAmount	2 орны нарийвчлалтай тоо	Бэлэн төлбөрийн дүн
nonCashAmount	2 орны нарийвчлалтай тоо	Бэлэн бүс төлбөрийн дүн
cityTax	2 орны нарийвчлалтай тоо	Нийслэл хотын албан татварын нийт дүн
districtCode	2 оронтой бүхэл тоо	Баримт хэвэлсэн орон нутгийн код /Татварын албаны орон нутгын код/
posNo	4-6 оронтой бүхэл тоо	Тухайн байгууллагын дотоод кассын дугаар
customerNo	7 оронтой бүхэл тоо эсвэл Монгол улсын иргэний регистер	Худалдан авагч байгууллагын ТТД эсвэл Иргэний регистерийн дугаар
billType	1 оронтой бүхэл тоо	Баримтын төрөл
billIdSuffix	6 оронтой бүхэл тоо	Баримтын ДДТД-ыг давхцуулахгүйн тулд олгох дотоод дугаарлалт. Тухайн өдөртөө дахин давтагдашгүй дугаар байна.
returnBillId	33 оронтой бүхэл тоо	Засварлах баримтын ДДТД
stocks		
code	Дурын тэмдэгт мөр	Бараа, үйлчилгээний код /байгууллагын дотоод код/
name	Дурын тэмдэгт мөр	Бараа, үйлчилгээний нэр /байгууллагын дотоод нэр/
measureUnit	Дурын тэмдэгт мөр	Хэмжих нэгж
qty	2 орны нарийвчлалтай тоо	Тоо, хэмжээ



unitPrice	2 орны нарийвчлалтай тоо	Нэгж үнэ
totalAmount	2 орны нарийвчлалтай тоо	Нийт үнэ
cityTax	2 орны нарийвчлалтай тоо	Нийслэл хотын албан татварын нийт дүн
vat	2 орны нарийвчлалтай тоо	НӨАТ-ын нийт дүн
barCode	Бүхэл тоо	Барааны зураасан код эсвэл бараа, ажил үйлчилгээний ангиллын код
bankTransactions		
rrn	12 оронтой бүхэл тоо	Бэлэн бус гүйлгээний баримтын дугаар
bankId	2 оронтой бүхэл тоо	Пос терминалын эзэмшигч банкны код
terminalId	Латин үсэг тоо холилдсон 6 ба түүнээс дээш тэмдэгт бүхий тэмдэгт мөр	Пос терминалын дугаар
approvalCode	10 оронтой латин үсэг тооноос бүрдсэн тэмдэгт мөр	Бэлэн бус гүйлгээний зөвшөөрлийн код
amount	2 орны нарийвчлалтай тоо	Бэлэн бус гүйлгээний дүн

Дээрх талбаруудаас хэд хэдэн талбарууд нь заавал бөглөх албагүй ба шаардлагатай гэж үзсэн системийн орчин эсвэл борлуулалтын төрөлд л бөглөнө.

Заавал бөглөхгүй талбарууд

Нэр	Тайлбар
billIdSuffix	Сервер шийдлээр буюу борлуулалтын мэдээ нь PosAPI-д хэд хэдээрээ зэрэг ирэх орчинд уг талбарыг заавал бөглөнө. Учир нь зэрэг ирж буй борлуулалтын мэдээнүүдээс улбаалан ДДТД нь давхцаж үүсэх магадлал өндөр юм.
customerNo	Байгууллага хоорондын борлуулалтыг бүртгэх үед заавал бөглөнө. Харин худалдан авагч хувь иргэнд борлуулж буй үед бөглөх шаардлагагүй. Учир нь төлбөрийн баримт нь НӨАТ-ын падааны орлож үүсэх тул худалдан авагч байгууллагын ТТД-ийг авах шаардлага тулгардаг.
billType	Баримтын төрөлийг дамжуулаагүй үед энгийн байгууллагаас хувь иргэнд борлуулалт гэж PosAPI ойлгон бүртгэж авдаг. Харин байгууллага хоорондын борлуулалт бол заавал бөглөж дамжуулна.

billIdSuffix талбар хэрэв “салбарын дугаар” бүхий PosAPI татаж авсан бол салбар дотроо, хэрэв салбарын дугаар оруулалгүй PosAPI татаж авсан бол тухайн байгууллагын хэмжээнд тухайн өдөртөө дахин давтагдашгүй дугаар байна.

Баримтын төрөл /billType талбар/

Утга	Харгалзах утга
1	Байгууллагаас хувь иргэнд борлуулсан бараа, ажил үйлчилгээний баримт
2	Байгууллагаас байгууллагын хооронд худалдан авсан бараа, ажил үйлчилгээний баримт
3	Байгууллагаас байгууллагын хооронд борлуулсан бараа, ажил үйлчилгээний баримт





Банкны код /bankId талбар/

Утга	Харгалзах утга
01	Монгол банк
02	Капитал банк
04	Худалдаа хөгжлийн банк
05	ХААН банк
15	Голомт банк
19	Тээвэр хөгжлийн банк
21	Ариг банк
22	Кредит банк

Утга	Харгалзах утга
26	Улаанбаатар хотын банк
29	Үндэсний хөрөнгө оруулалтын банк
30	Капитрон банк
32	ХАС банк
33	Чингис Хаан банк
34	Төрийн банк
36	Хөгжлийн банк
38	Богд банк

Аймаг/Дүүргийн код /districtId талбар/

Утга	Харгалзах утга
01	Архангай
02	Баян-Өлгий
03	Баянхонгор
04	Булган
05	Говь-Алтай
06	Дорноговь
07	Дорнод
08	Дундговь
09	Завхан
10	Өвөрхангай
11	Өмнөговь
12	Сүхбаатар
13	Сэлэнгэ
14	Төв
15	Увс

Утга	Харгалзах утга
16	Ховд
17	Хөвсгөл
18	Хэнтий
19	Дархан-Уул
20	Орхон
32	Говьсүмбэр
23	Хан-Уул
24	Баянзүрх
25	Сүхбаатар
26	Баянгол
27	Багануур
28	Багахангай
29	Налайх
34	Сонгинохайрхан
35	Чингэлтэй

Гаралтын утга

Гаралтын утгын формат

```
{
  "success":boolean,
  "registerNo":String,
  "billId":String,
  "date":String,
  "macAddress":String,
  "internalCode":String,
  "billType":String,
  "qrData":String,
  "lottery":String,
  "lotteryWarningMsg":String,
  ...
}
```

Гаралтын утгын тайлбар

Нэр	Тайлбар
success	Баримтыг бүртгэх процесс амжилттай болсон тухай илтгэнэ



	<i>true – хүчингүй болгосон</i> <i>false – хүчингүй болгож чадаагүй</i>
registerNo	PosAPI эзэмшигч ААН-ийн ТТД эсвэл Татвар суутган төлөгч иргэний хөрвүүлсэн ТТД
billId	Баримтын ДДТД 33 оронтой тоон утга /НӨАТ-ийн тухай хуулинд зааснаар/
date	Баримт хэвлэсэн огноо Формат: yyyy-MM-dd hh:mm:ss
macAddress	Баримтыг хэвлэсэн бүртгэлийн машиний MacAddress
internalCode	Баримтын дотоод код
billType	Баримтын төрөл
qrData	Баримтын баталгаажуулах Qr кодонд орох нууцлагдсан тоон утга
lottery	Сугалааны дугаар
lotteryWarningMsg	Сугалаа дуусаж буй эсвэл сугалаа хэвлэх боломжгүй болсон талаар мэдээлэл өгөх утга
errorCode	Хэрэв алдаа илэрсэн бол уг алдааны код
message	Алдааны мэдээллийн текстэн утга

Уг талбаруудын хамт анх дамжуулсан талбарууд цуг ирэх ба уг талбаруудаас зөвхөн billId болон date талбарыг л хадгалж болно. Учир нь уг 2 талбар баримтыг хүчингүй болгоход ашиглагдана.

Баримт засварлах:

Баримтыг засварлах буюу хэсэгчилсэн буцаалт хийх тохиолдолд returnBillId талбарт засварлах баримтын дугаарыг оруулан баримтын шинэ бичилтийг хийнэ. Баримт засварлаж буй тохиолдолд тухайн хэвлэгдсэн сугалаа хүчингүй болохгүй бөгөөд засварлагдсан баримтанд сугалаа болон qrCode хэвлэгдэхгүй.

Ustring PosAPI::returnBill(Ustring)

Хэрэглэгчийн системээс бүртгэгдэн гарсан баримтийг хүчингүй болгоно. Уг функцээр буцаасан нөхцөлд тухайн баримтан дээрх сугалаа болон дахин давтагдашгүй төлбөрийн дугаар/ДДТД/ зэргийг хүчингүй гэж үзнэ.

Оролтын утга

Оролтын утгын формат

```
{
  "returnBillId":String,
  "date":String
}
```

Оролтын утгын тайлбар

Нэр	Нөхцөл	Тайлбар
returnBillId	33 орон бүхий тоон утга	Хүчингүй болгох баримтын ДДТД
date	Формат:	Баримт хэвлэсэн огноо



yyyy-MM-dd hh:mm:ss

Гаралтын утга

Гаралтын утгын формат

```
{
  "success":boolean,
  "errorCode":Integer,
  "message":String
}
```

Гаралтын утгын тайлбар

Нэр	Тайлбар
success	Баримт хүчингүй болгох процесс амжилттай болсон тухай илтгэнэ <i>true – хүчингүй болгосон</i> <i>false – хүчингүй болгож чадаагүй</i>
errorCode	Хэрэв алдаа илэрсэн бол уг алдааны код
message	Алдааны мэдээллийн текстэн утга

Ustring PosAPI::sendData()

Баримтын мэдээллийг хуулинд заасны дагуу борлуулалт гэж үзэн бүртгэсэнээс хойш 72 цагийн дотор илгээх ёстой. Уг функц нь борлуулалтын мэдээллийг татварын нэгдсэн системд илгээнэ.

Мөн хэрэглэгчийн системд PosAPI-г анх удаа суурьлуулж буй эсвэл сугалааны багц дууссан нөхцөлд мөн уг функцийг ажиллуулж тохиргоо болон шинэ сугалааны багцийг татаж авна.

Хэрэв тухайн бүртгэлийн машинд PosAPI 2.0 санг шинээр суурьлуулсан бол sendData функцийг-ийг заавал нэг удаа хоосон дуудна. Ингэснээр НӨАТУС-д тухайн PosAPI-г бүртгүүлж, шаардлагатай тохиргооны мэдээллийг татна.

Оролтын утга

Байхгүй

Гаралтын утга

Гаралтын утгын формат

```
{
  "success":boolean,
  "errorCode":Integer,
  "message":String
}
```

Гаралтын утгын тайлбар

Нэр	Тайлбар
success	Борлуулалтын мэдээллийг татварын нэгдсэн системд амжилттай илгээж чадсан эсэхийг илтгэнэ. <i>true - ажиллагаа хэвийн</i> <i>false - ажиллагаа хэвийн бус</i>
errorCode	Хэрэв илгээх процессд алдаа гарсан бол уг алдааны код



message

Алдааны мэдээллийн текстэн утга



Алдааны мэдээлэлийн тайлбар

Код	Мессеж	Тайлбар
1000	Системийн алдаа.	Үйлдлийн системийн орчин бүрүү эсвэл алдаатай нөхцөлд илрэх алдаа
1001	Системийн алдаа.	
1002	Системийн алдаа.	
1003	Системийн алдаа.	
1004	Системийн алдаа.	
1005	Системийн алдаа.	
0	Key Not Found !!!	Тохиргооны мэдээлэл татагдаагүй, дутуу татагдсан эсвэл тохиргооны мэдээлэл бүрүү ирсэн нөхцөлд үүсэх алдаа
500	Мэдээлэл дамжуулалт хэт удаан байна. Мэдээлэл дамжуулах дээд хязгаар {0} секунд байна!!!	Сүлжээ хэт удаан эсвэл тасалдал ихтэй үед илрэх алдаа.
501	Мэдээлэл дамжуулалт хэт удаан байна. Мэдээлэл дамжуулах дээд хязгаар {0} секунд байна!!!	Системд мэдээлэл илгээх дээд хязгаар анхны байдлаар 30 секунд байна.
300	Бааз нээх эсвэл үүсгэх явцад алдаа гарлаа.\n	SQLite баазын файл үүсгэх эсвэл холболт хийх үед гарах алдаа.
301	Бааз үүсгэх явцад алдаа гарлаа. (INFO_DATA)\n	Ихэвчлэн home directory байхгүй эсвэл бичих эрхгүй хэрэглэгчээр дуудахад гарна.
302	Өгөгдөлийн баазтай харьцах явцад алдаа гарлаа!!!\n	Өгөгдлийн баазруу бүрүү query ажиллуулахад гарах алдаа.
303	Өгөгдөлийн баазтай харьцах явцад алдаа гарлаа!!!\n	
304	Өгөгдөлийн баазтай харьцах явцад алдаа гарлаа!!!\n	
305	Баазтай холболт хийгдээгүй байна!!!	SQLite баазтай холболт хийх үед гарах алдаа
400	Тохиргооны мэдээлэл эвдэрсэн байна!!! :	Тохиргооны мэдээлэл татагдаагүй, дутуу татагдсан эсвэл тохиргооны мэдээллийг хүчээр сольсон үед илрэх алдаа
401	Тохиргооны мэдээлэл эвдэрсэн байна!!! :	
402	Тохиргооны мэдээлэл эвдэрсэн байна!!! :	
403	Тохиргооны мэдээлэл эвдэрсэн байна!!! :	
100	Тохиргооны мэдээлэл олдсонгүй. Мэдээлэл илгээх хүсэлтийг ажлуулж тохиргооны мэдээллүүдийг татна уу!!!	Шинээр PosAPI татаж суурьлуулан шууд төлбөрийн баримт бүртгэх эсвэл төлбөрийн баримт буцаах үед илрэх алдаа. Уг тохиолдолд алдаан дээрх зааврыг дагна уу!!!
310	Баазтай харьцахад алдаа гарлаа!!!	Өгөгдлийн баазруу бүрүү query ажиллуулахад гарах алдаа.
311	Баазтай харьцахад алдаа гарлаа!!!	
312	Баазтай харьцахад алдаа гарлаа!!!	
313	Баазтай харьцахад алдаа гарлаа!!!	
314	Баазтай харьцахад алдаа гарлаа!!!	
315	Баазтай харьцахад алдаа гарлаа!!!	
316	Баазтай харьцахад алдаа гарлаа!!!	
317	Баазтай харьцахад алдаа гарлаа!!!	



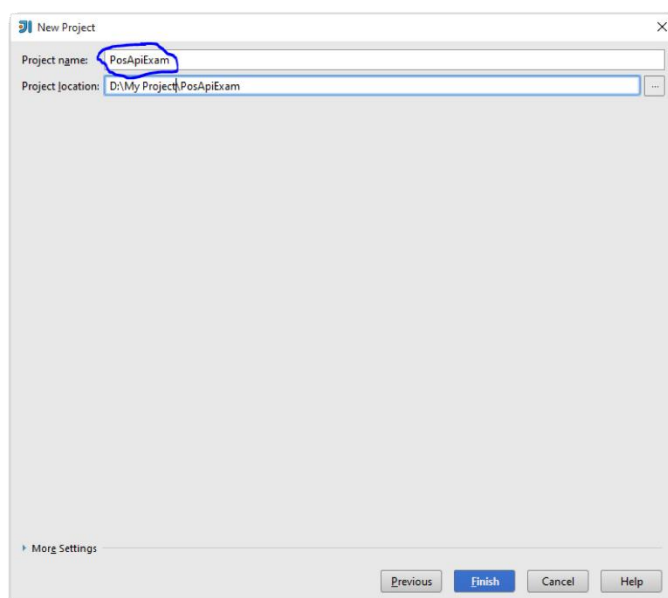
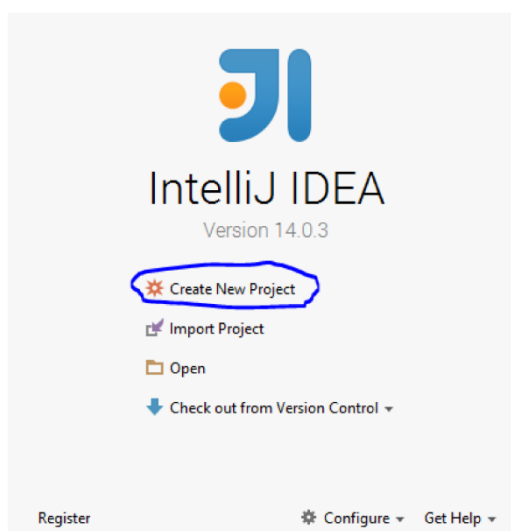
318	Баазтай харьцахад алдаа гарлаа!!!	
320	Баазтай харьцахад алдаа гарлаа!!!	
210	/ {0}/ талбар буруу өгөгдөл оруулсан байна. Шалгасан нөхцөл нь : {1}	Төлбөрийн баримт бүртгэх үед шалгах нөхцөлийг даваагүй бол илрэх алдаа. {0} нь JSON талбарын нэр, {1} шалгасан RegEx нөхцөл.
220	Бэлэн, бэлэн бусын нийлбэр нийт мөнгөн дүнтэй тэнцэхгүй байна.	nonCashAmount + cashAmount == amount гэсэн нөхцөлийг даваагүй байна.
600	Нэвтрэх үйлдэл амжилтгүй боллоо!!! \n	Тухайн PosAPI-г татварын нэгдсэн системээс хорьсон нөхцөлд эсвэл сүлжээнд холбоогүй, доголдолтой нөхцөлд шидэх үйлдлийг эхлүүлсэн бол уг алдаа үүсэнэ
601	Алдаа: [{0}]	Төлбөрийн баримтыг илгээх үед сүлжээнд гарсан доголдолын алдааны тайлбар
2	Тохиргооны файл эвдэрсэн байна.	Тохиргооны мэдээлэл татагдаагүй эсвэл тохиргооны мэдээллийг хүчээр сольсон үед илрэх алдаа
1	Тохиргооны файл эвдэрсэн байна.	
200	JSON хөрвүүлэх явцад алдаа гарлаа.	JSON өгөгдөлийн бүтэц гэмтэлтэй үед илрэх алдаа
201	JSON хөрвүүлэх явцад алдаа гарлаа.	



Java програмчлалын хэлтэй холбох жишээ

Java програмчлалын хэл нь C++ хэлтэй холбогдох бүрэн боломжтой бөгөөд холбогдохын тулд JNI /Java Native Interface/-ийн түслэмжтайгаар холбогдоно. Мөн java хэл дээр хөгжүүлэхийн тулд дээр дурьдсан “Зураг 3” дээрх аргачлалаар холбогдоно. Доорх жишээ нь IntelliJ IDEA болон Visual Studio 2013 дээр хийж гүйцэтгэсэн.

Алхам 1: Project Үүсгэх





Алхам 2: Native method үүсгэх

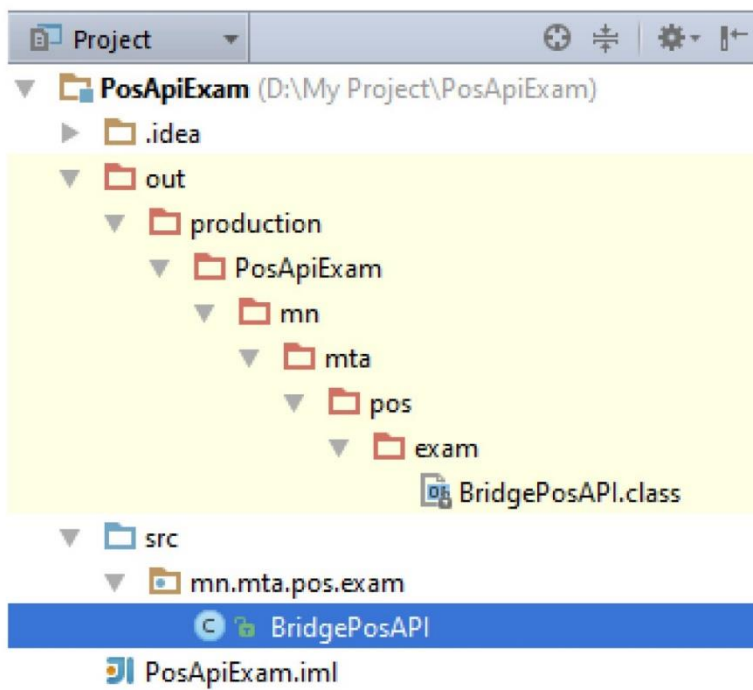
```
package mn.mta.pos.exam;
```

```
public class BridgePosAPI {  
    public static native String put(String data);  
    public static native String returnBill(String data);  
    public static native String sendData();  
    public static native String checkAPI();  
    public static native String getInformation();  
    public static native String callFunction(String funcName, String data);  
}
```

Дээрх method-үүд нь PosAPI::put, PosAPI::returnBill, PosAPI::sendData, PosAPI::checkAPI, PosAPI::getInformation, PosAPI::callFunction, гэсэн C++ сангийн method-уудыг дуудна.

Native class-аас C header файл үүсгэх

C header файл үүсгэхийн тулд эхлээд BridgePosAPI class-ийг compile хийж хөрвүүлнэ. Хөрвүүлэхдээ дээд цэсний Build -> Make Project сонгож хөрвүүлэх эсвэл CTRL+F9 товчлуурын хослолыг дарна. Хөрвүүлэлт амжилттай болсон бол доорх зурганд байгаатай адил үр дүн харагдана.



Дээрх үр дүн гарч ирсэн бол console /CMD/ нээж доорх командуудыг гүйцэтгэнэ.

```
cmd> cd D:\My Project\PosApiExam\out\production\PosApiExam cmd> javah  
mn.mta.pos.exam.BridgePosAPI
```



javah команд нь native java class-аас C-гийн header файл үүсгэх команд юм. Үр дүнд нь “mn_mta_pos_exam_BridgePosAPI.h” гэсэн файл үүссэн байна.

mn_mta_pos_exam_BridgePosAPI.h:

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class mn_mta_pos_exam_BridgePosAPI */

#ifndef _Included_mn_mta_pos_exam_BridgePosAPI
#define _Included_mn_mta_pos_exam_BridgePosAPI
#ifdef __cplusplus
extern "C" {
#endif

    /*
     * Class:      mn_mta_pos_exam_BridgePosAPI
     * Method:     put * Signature: (Ljava/lang/String;)Ljava/lang/String;
     */
    JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_put
        (JNIEnv *, jclass, jstring);

    /*
     * Class:      mn_mta_pos_exam_BridgePosAPI
     * Method:     returnBill
     * Signature: (Ljava/lang/String;)Ljava/lang/String;
     */
    JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_returnBill
        (JNIEnv *, jclass, jstring);

    /*
     * Class:      mn_mta_pos_exam_BridgePosAPI
     * Method:     sendData * Signature: ()Ljava/lang/String;
     */
    JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_sendData
        (JNIEnv *, jclass);

    /*
     * Class:      mn_mta_pos_exam_BridgePosAPI
     * Method:     checkAPI * Signature: ()Ljava/lang/String;
     */
    JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_checkAPI
        (JNIEnv *, jclass);

    /*
     * Class:      mn_mta_pos_exam_BridgePosAPI
     * Method:     getInformation * Signature: ()Ljava/lang/String;
     */
    JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_getInformation
```



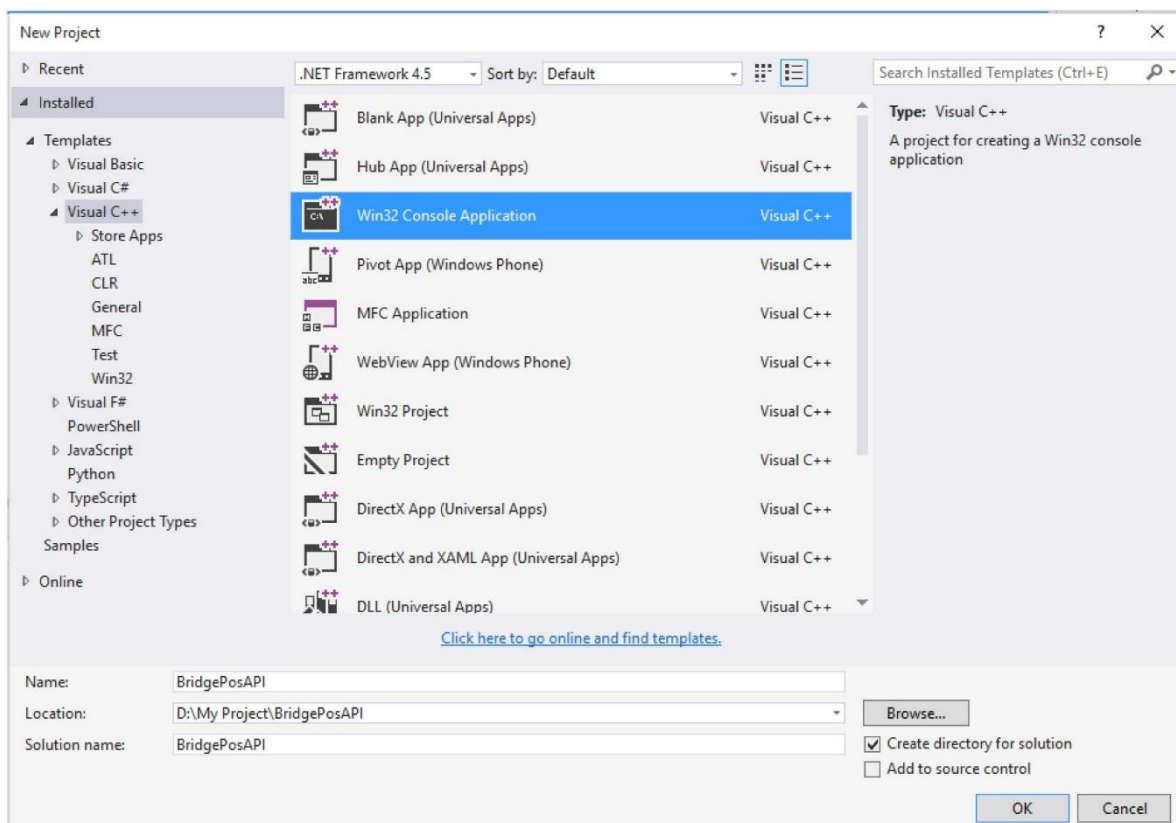
```
(JNIEnv *, jclass);

/*
 * Class:      mn_mta_pos_exam_BridgePosAPI
 * Method:     callInformation      * Signature: ()Ljava/lang/String;
 */
JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_callInformation
(JNIEnv *, jclass, jstring, jstring);

#ifdef cplusplus
}
#endif
#endif
```

Алхам 3: Visual Studio 2013 C++ project үүсгэх

Дээрх үйлдлүүд хийгдсэний дараа Visual Studio 2013 нээж холбох сангаа үүсгэнэ. Project үүсгэхдээ “**New Project -> Win32 ConsoleApplication**” сонгож Next товч дарна.





Дараа нь **Application Type** -> **DLL** сонгож Finish товчийг дарснаар Project үүснэ

Application type:

☐ Windows application

☐ Console application

☒ DLL

☐ Static library

Additional options:

☐ Empty project

☐ Export symbols

☒ Precompiled header

☒ Security Development Lifecycle (SDL) checks

Add common header files for:

☐ ATL

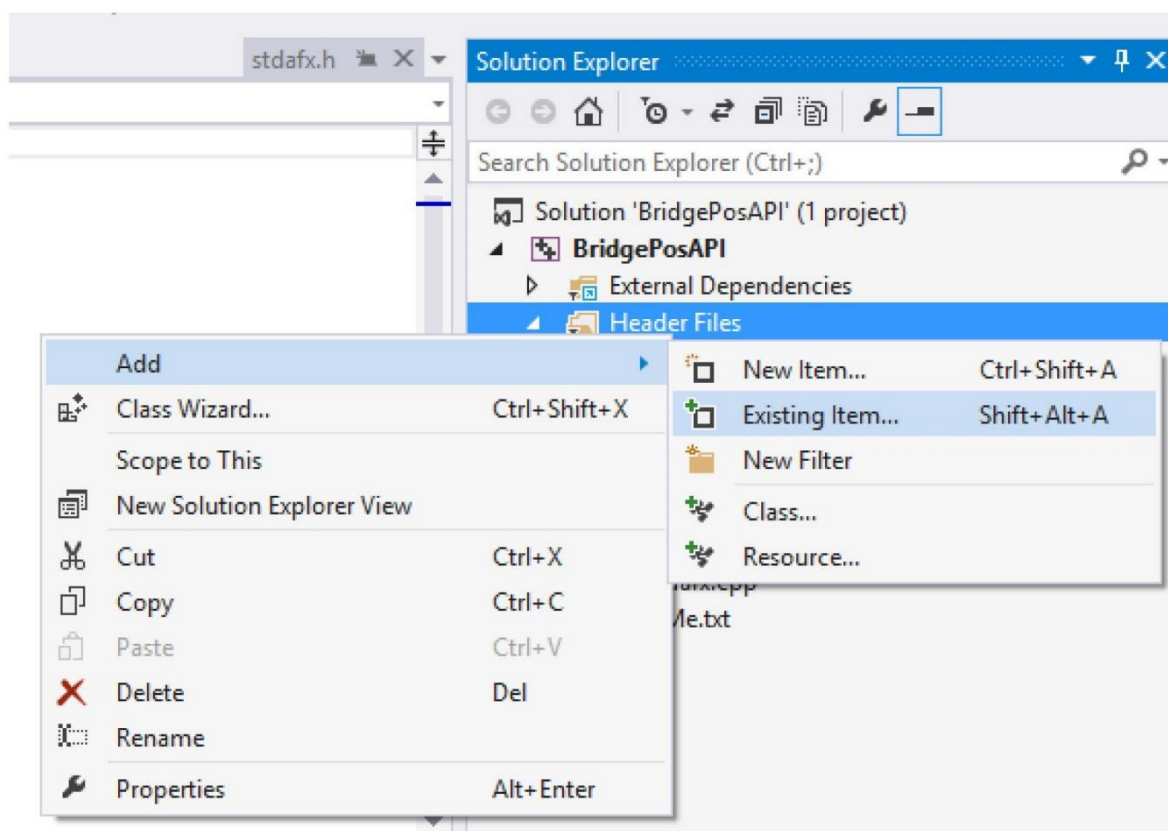
☐ MFC

Алхам 4: mn_mta_pos_exam_BridgePosAPI.h оруулж ирэх

Өмнө нь үүсгэсэн файл болох “mn_mta_pos_exam_BridgePosAPI.h” файлыг сору хийж Visual Studio Project-ийн хавтсанд хуулна.

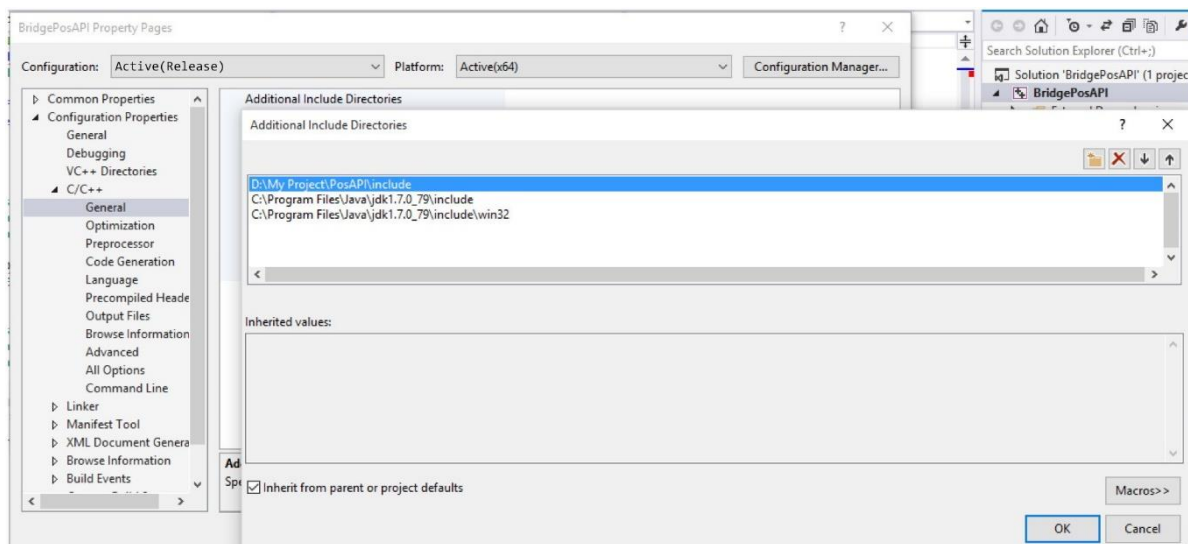
This PC > Local Disk (D:) > My Project > BridgePosAPI > BridgePosAPI > BridgePosAPI				
Name	Date modified	Type	Size	
x64	2015/9/2 18:59	File folder		
BridgePosAPI.cpp	2015/9/2 20:01	C++ Source file	1 KB	
BridgePosAPI.vcxproj	2015/9/2 18:59	VC++ Project	10 KB	
BridgePosAPI.vcxproj.filters	2015/9/2 18:59	VC++ Project Filte...	2 KB	
dllmain.cpp	2015/9/2 17:36	C++ Source file	1 KB	
mn_mta_pos_exam_BridgePosAPI.h	2015/9/2 17:16	C++ Header file	1 KB	
ReadMe.txt	2015/9/2 17:36	Text Document	3 KB	
stdafx.cpp	2015/9/2 17:36	C++ Source file	1 KB	
stdafx.h	2015/9/2 17:36	C++ Header file	1 KB	
targetver.h	2015/9/2 17:36	C++ Header file	1 KB	

Үүний дараа уг файлыг оруулж ирэхийн тулд Solution Explorer -> Header Files дээр хулганы баруу товчыг дарж Add->Existing Item цэсийг сонгож “mn_mta_pos_exam_BridgePosAPI.h” файлыг сонгож оруулж ирнэ



JDK header file болон PosAPI-ийн холболтын тохиргоохийх

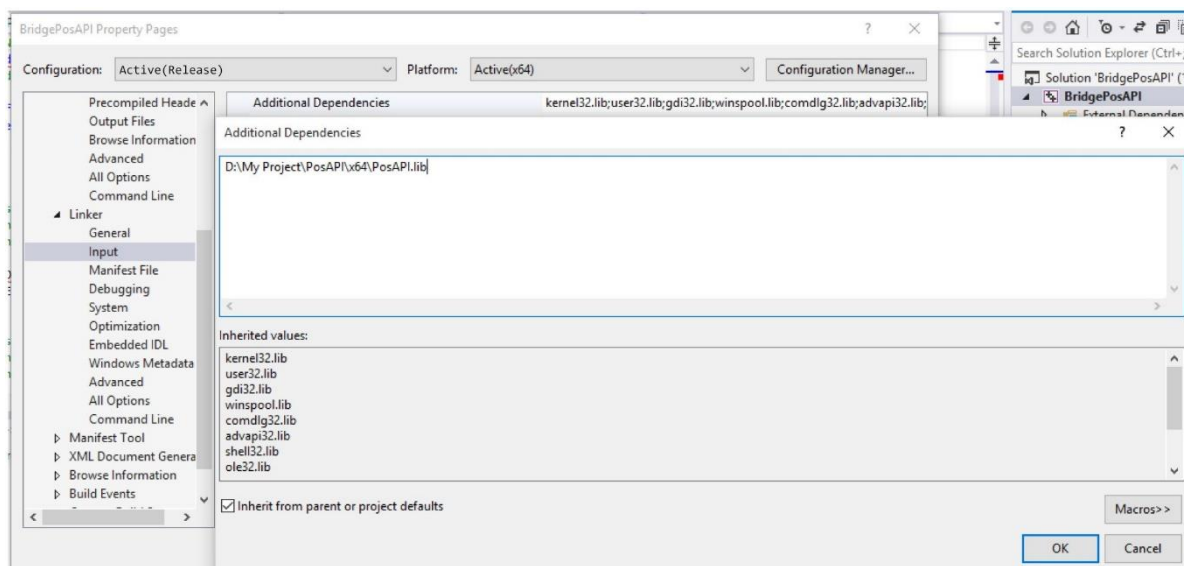
Java хэлтэй холбож буй үед jni.h header файлыг ашигладаг. Уг header файлыг оруулж ирэхийн тулд дараах үйлдлүүдийг хийнэ. Толгойн цэснээс PROJECT -> Property цэсийг сонгоно. **Configuration Properties -> C/C++ -> General -> Additional Include Directories** сонгож дараах замуудыг тодорхойлж өгнө. [JDK PATH]\include болон [JDK PATH]\include\win32 замуудыг тодорхойлж өгнө. Мөн татаж авсан [PosAPI PATH]\include directory-г зааж өгнө.





Дээрх header файлуудыг оруулж ирсэн бол PosAPI.lib файлыг мөн холбоно

Толгойн цэснээс PROJECT -> Property цэсийг сонгоно. **Configuration Properties->Linker->Input->Additional Dependencies** сонгож **[PosAPI PATH]\[Cpu Architecture]\PosAPI.lib** замыг тодорхойлж өгнө.



Алхам 5: BridgePosAPI.cpp файлд холболтын кодыг бичинэ

BridgePosAPI.cpp файлыг нээж доорх кодыг бичнэ.

```
#include "stdafx.h"
#include "mn_mta_pos_exam_BridgePosAPI.h"
#include <PosAPI.h>
#include <codecvt>
#include <locale>
#include <vector>

using namespace vatps;
using namespace std;

/*
 * std::string-ийг std::wstring төрөлрүү хөрвүүлэх method
 */
wstring s2ws(const string& str)
{
    typedef codecvt_utf8<wchar_t> convert_typeX;
    wstring_convert<convert_typeX, wchar_t> converterX;
    return converterX.from_bytes(str);
}

/*
 * std::wstring-ийг std::string төрөлрүү хөрвүүлэх method
```




```

*/
string ws2s(const wstring& wstr)
{
    typedef codecvt_utf8<wchar_t> convert_typeX;
    wstring_convert<convert_typeX, wchar_t> converterX;
    return converterX.to_bytes(wstr);
}

JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_put (JNIEnv
*env, jclass c, jstring param){
    const char* strParam = env->GetStringUTFChars(param, 0);
    UString data = s2ws(string(strParam));
    UString result = PosAPI::put(data);
    return env->NewStringUTF(ws2s(result).c_str());
}

JNIEXPORT jstring JNICALL
Java_mn_mta_pos_exam_BridgePosAPI_returnBill
(JNIEnv *env, jclass c, jstring param){
    const char* strParam = env->GetStringUTFChars(param, 0);
    UString data = s2ws(string(strParam));
    UString result = PosAPI::returnBill(data);
    return env->NewStringUTF(ws2s(result).c_str());
}

JNIEXPORT jstring JNICALL Java_mn_mta_pos_exam_BridgePosAPI_sendData
(JNIEnv *env, jclass c){

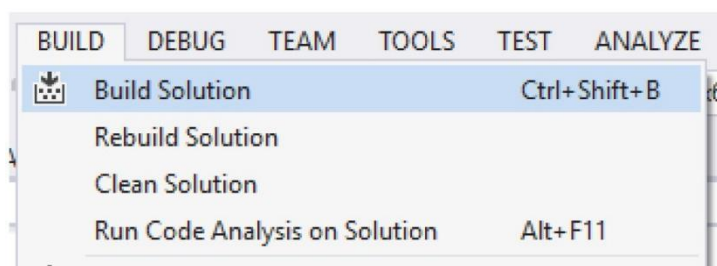
    UString result = PosAPI::sendData();

    return env->NewStringUTF(ws2s(result).c_str());
}

```

Алхам 6: BridgePosAPI project-ийг compile хийж dll санг үүсгэх

Кодыг бичиж дуусгасан бол Build->Build Solution цэсийг сонгон project-ийг release mode ашиглан compile хийнэ.





Хэрэв compile хийх үйлдэл амжилттай болсон бол доорх message харагдана.

```
1>----- Build started: Project: BridgePosAPI, Configuration: Release x64 -----
1> BridgePosAPI.cpp

1> Creating library D:\My Project\BridgePosAPI\BridgePosAPI\x64\Release\Bridge-
PosAPI.lib and object D:\My Project\BridgePosAPI\BridgePosAPI\x64\Release\Bridge-
PosAPI.exp

1> BridgePosAPI.vcxproj -> D:\My Project\BridgePosAPI\BridgePosAPI\x64\Release\
BridgePosAPI.dll

===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

Тодруулсан текст нь compile-ийн үр дүнд үүссэн dll сан хаана байрлаж буйг заана

Алхам 7: Үүсгэсэн санг Java project-той холбож дуудах

Үүсгэсэн *BridgePosAPI.dll*, *PosAPI.dll* сангуудыг бусад туслах сангуудын хамт *C:/Windows/Sun/Java/Bin* хавтсанд хуулж өгнө. Хэрэвүг хавтас үүсээгүй бол гараар үүсгэж өгнө. Мөн уг хавтасанд заавал хуулах шаардлагагүй ба хүсвэл jar файлын дэргэд хуулж өгөх эсвэл *System.getProperty("java.library.path")*/хэвлэж харах/ тодорхойлсон замуудад мөн адил хуулж өгч болно.

Дээрх үйлдлийн дараа java кодноос тухайн сангуудыг дуудаж холбоно.

```
package mn.mta.pos.exam;
public class BridgePosAPI {
    static{
        String os = System.getProperty("os.name"); if
        (os.toUpperCase().contains("WINDOWS")) {
            System.loadLibrary("icudt53");
            System.loadLibrary("icuuc53");
            System.loadLibrary("icuin53");
            System.loadLibrary("Qt5Core");
            System.loadLibrary("Qt5SQL");
            System.loadLibrary("Qt5Network");
            System.loadLibrary("Qt5Gui");
            System.loadLibrary("Qt5Widgets");
            System.loadLibrary("PosAPI");
        }
        System.loadLibrary("BridgePosAPI");
    }

    public static native String put(String data);
    public static native String returnBill(String data);
    public static native String sendData();
    public static void main(String[] args) {
        String result = sendData();
        System.out.println("result = " + result);
    }
}
```



Дээрх кодыг бичсэний дараа main method-ийг дуудахад доорх үр дүн харагдаж байвал таны код амжилттай ажилласан гэж ойлгож болно.

Гаралт:

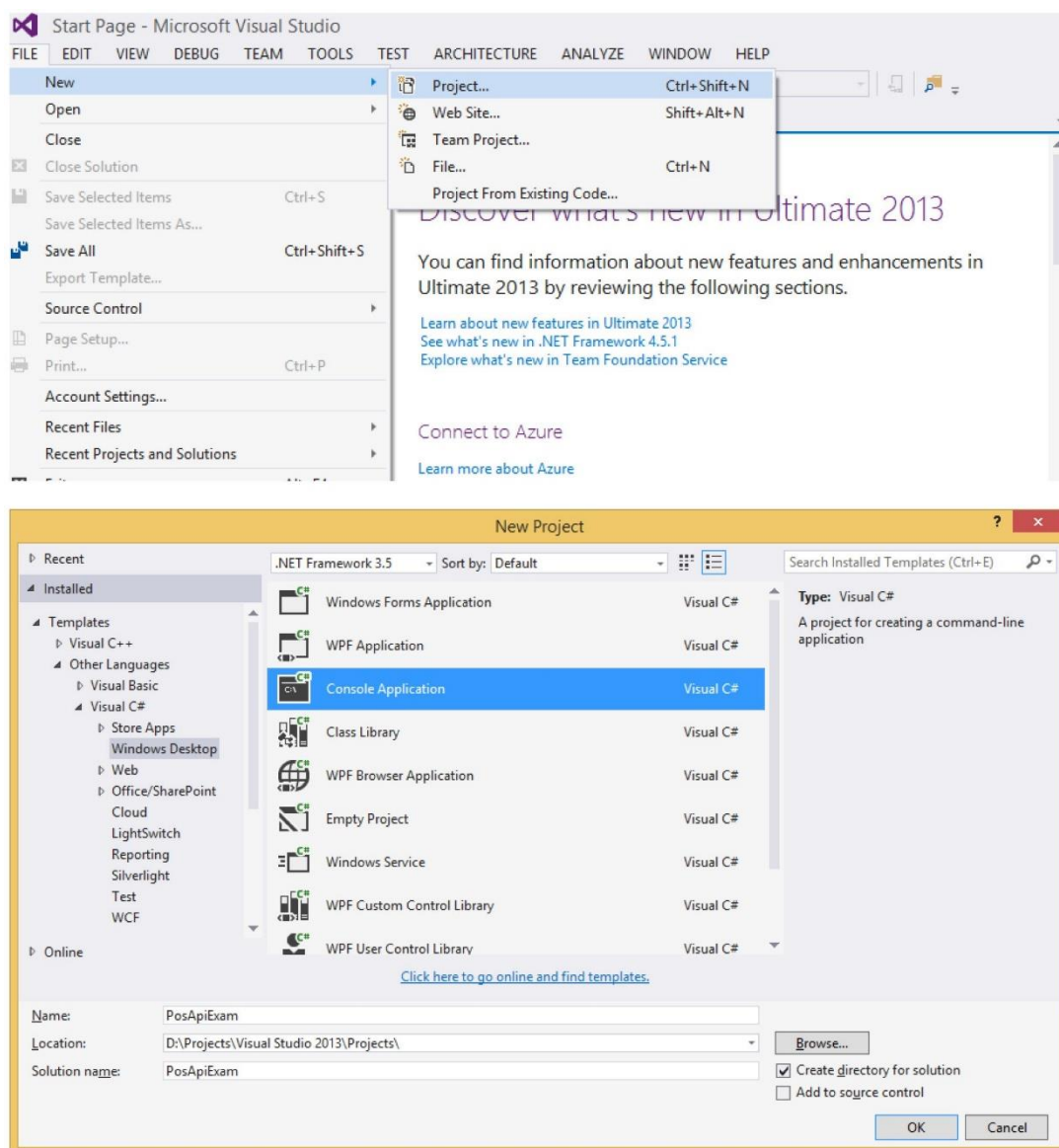
```
result = {  
    "success": true  
}
```



С# .Net програмчлалын хэлтэй холбох жишээ

С# програмчлалын хэл ашиглан С++ хэл ашиглан бичигдсэн сантай холбогдох бүрэн боломжтой. Холбогдохын тулд Unmanaged С++ library-г С# програмдаа доорх зурган дээрх аргачлалаар холбоно. Доорх жишээ нь Visual Studio 2013 дээр хийж гүйцэтгэсэн.

Алхам 1: Project Үүсгэх





Алхам 2: Native method үүсгэх

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;

namespace PosApiExam
{
    class Program
    {
        [DllImport(
            "PosAPI.dll",
            CharSet = CharSet.Unicode, CallingConvention =
            CallingConvention.Cdecl
        )]

        [return: MarshalAs(UnmanagedType.BStr)]
        public static extern string put(String message);

        [DllImport(
            "PosAPI.dll",
            CharSet = CharSet.Unicode, CallingConvention =
            CallingConvention.Cdecl
        )]

        [return: MarshalAs(UnmanagedType.BStr)]
        public static extern string returnBill(String message);

        [DllImport("PosAPI.dll")]
        [return: MarshalAs(UnmanagedType.BStr)]
        public static extern string sendData();

        [STAThread]
        static void Main(string[] args)
        {
            var result = sendData();
            Console.WriteLine("result = " + result);
        }
    }
}
```

Дээрх method-ууд нь PosAPI::put, PosAPI::returnBill, PosAPI::sendData гэсэн C++ сангийн method-уудыг дуудана.

Дээрх кодыг бичсэний дараа main method-ийг дуудахад доох үр дүн харагдаж байвал таны код амжилттай ажилласан гэж ойлгож болно.

Гаралт:

```
result = {
    "success": true
}
```



Танд амжилт хүсье



НЭМЭГДСЭН ӨРТГИЙН
АЛБАН ТАТВАРЫН
УРАМШУУЛЛЫН СИСТЕМ

