

Nastavenia

Zmena nastavení má vplyv na správne fungovanie celej aplikácie. Pred zmenami nastavení je nutné sa oboznámiť s jednotlivými funkciami, ktorých sa nastavenie bude týkať.

- Hlavné nastavenia - systémové hodnoty
- Besteron POS - ECR/CHDU - nastavenia pokladne
- Besteron POI - nastaveia platobného terminálu
- Objednávkové tlačiarne - slip (kuchynské) tlačiarne

Poznámka: táto kapitola je v rekonštrukcii...

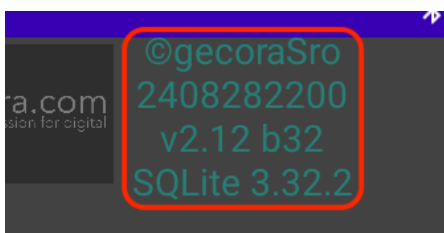
- [Hlavné nastavenia](#)
- [Pomocné procesy](#)
- [Import dokladu predaja](#)

Hlavné nastavenia

Hlavné nastavenia ovládajú kľúčové vlastnosti APK.

Spustiť Hlavné nastavenia

- stlačte dlho informáciu o verzii APK.



Aktivácia prijímania katalógu

- Zaškrtnite spínače pre kategórie a produkty

urlCategories: (<https://zastavka.gecommerce.io/api/v2/categories>)

<https://zastavka.gecommerce.io/api/v2/categories>

☒ activate read

☐ activate send

urlProducts: (<https://zastavka.gecommerce.io/api/v2/products>)

<https://zastavka.gecommerce.io/api/v2/products>

☒ activate read

☐ activate send

Aktivácia prijímania zoznamu partnerov a skladov

- Zaškrtnite spínače pre partnerov a sklady

urlPartners: (https://zastavka.gecommerce.io/api/v2
/partners)

https://zastavka.gecommerce.io/api/v2
/partners

☒ activate read

☐ activate send

urlStores: (https://zastavka.gecommerce.io/api/v2
/stores)

https://zastavka.gecommerce.io/api/v2
/stores

☒ activate read

☐ activate send

Please correctly fill "urlId" and "apiKeySharedSpace"

Ukončiť Hlavné nastavenia

- Potvrďte tlačidlo Ok
- Vráťte sa na úvod

Poznámky

- po aktivácii aktualizácie katalógu sa po stlačení tlačidla "Predaj" v úvode budú zobrazovať informácie o poslednej ich aktualizácii

Pomocné procesy

Tieto funkcie je vhodné využiť pre náhradné, opravné či doplňujúce účely.

Náhrada hotovostnej platby

v otvorenom doklade s produktami pre predaj postupujte:

- Platiť(modré) -> Výdajka -> Tlač => dať bloček zákazníkovi (náhradný doklad) Info -> Tlač
=> nechať si bloček (tí čo majú starý program na PC, tak nie) -> Zahodiť, pretože:
 - momentálne (20241127 verzia 2.23b43) je aj odloženie dokladu je postihnuté
 - info doklady sa doblokuju vo verzii 2.24b44, ktorá už je v Sunmi store a automaticky pôjde cez Besteron store do zariadení

Import dokladu predaja

Komunikácia s programom tretej strany

Ak zariadenie - platobný terminál slúži pre komunikáciu s cudzím systémom, je vhodné použiť formát JSON s tu uvedenou štruktúrou. Je možné požiadať o vytvorenie alternatívnych formátov a ich zloženie.

Princíp komunikácie

Terminál ak je len v režime POI, potom sa gePOS.apk nastaví na mock pre eKasa a doklady budú uložené v zariadení na zvolenú dobu. Ak je terminál zároveň aj eKasa, potom okrem uloženia v lokále budú dáta predaja exportované aj do gecommerce.solutions (sklady).

- Na FTP sa uloží JSON (lokálny server, či WampServer alebo XAMPP)
- gePOS -> Predaj -> Predaj -> Katalóg(podržanie) : dôjde k importu
- ďalej stlačiť Platiť -> Platba kartou : volá sa POI-Besteron a prípadne POS-FiscalCore
- alebo stlačiť platiť -> Platba : ak je zariadenie zároveň eKasa

Príklad JSON

Táto štruktúra (prvý vzor je pradž, druhý vzor je vrátenie) je momentálne zapracovaná - toto je praktický príklad:

```
{
  "head": {
    "date": "20241118",
    "id": 4,
    "note": "myEcr receiptExport",
    "sum": 8.38,
    "user": "1 admin"
  },
  "footer": {
    "note": "Thanks for visit"
  },
  "items": [
    {
      "code": "00020",
      "costpricebase": 0.500,
```

```
"costpricefull": 0.615,
"costpricetax": 0.115,
"customerpricebase": 0.976,
"customerpricefull": 1.200,
"customerpricetax": 0.224,
"hour": "14.20",
"name": "Komodita test 23%",
"note": "",
"orderprinterid1": 0,
"orderprinterid2": 0,
"plu": 1,
"qty": 2,
"receiptdivisor": 0,
"receiptplu": 0,
"referenceid": "",
"selected": true,
"sktype": "",
"sktypezerovat": "",
"stock": 1,
"sumcustomerpricebase": 1.951,
"sumcustomerpricefull": 2.400,
"sumcustomerpricetax": 0.449,
"vatid": 1,
"vatlevel": 23
},
{
  "code": "00010",
  "costpricebase": 0.500,
  "costpricefull": 0.595,
  "costpricetax": 0.095,
  "customerpricebase": 0.924,
  "customerpricefull": 1.100,
  "customerpricetax": 0.176,
  "hour": "14.20",
  "name": "Komodita test 19%",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 2,
  "qty": 2,
```

```
"receiptdivisor": 0,
"receiptplu": 0,
"referenceid": "",
"selected": true,
"sktype": "",
"sktypezerovat": "",
"stock": 1,
"sumcustomerpricebase": 1.849,
"sumcustomerpricefull": 2.200,
"sumcustomerpricetax": 0.351,
"vatid": 2,
"vatlevel": 19
},
{
  "code": "00099",
  "costpricebase": 0.150,
  "costpricefull": 0.150,
  "costpricetax": 0,
  "customerpricebase": 0.150,
  "customerpricefull": 0.150,
  "customerpricetax": 0,
  "hour": "14.20",
  "name": "Komodita test 0% VratObal",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 3,
  "qty": 2,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "",
  "selected": false,
  "sktype": "",
  "sktypezerovat": "VO",
  "stock": 1,
  "sumcustomerpricebase": 0.300,
  "sumcustomerpricefull": 0.300,
  "sumcustomerpricetax": 0,
  "vatid": 3,
  "vatlevel": 0
```

```
},
{
  "code": "muka",
  "costpricebase": 0.500,
  "costpricefull": 0.525,
  "costpricetax": 0.025,
  "customerpricebase": 1,
  "customerpricefull": 1.050,
  "customerpricetax": 0.050,
  "hour": "14.20",
  "name": "Múka 1kg 5%",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 7,
  "qty": 2,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "",
  "selected": true,
  "sktype": "",
  "sktypezerovat": "",
  "stock": 1,
  "sumcustomerpricebase": 2,
  "sumcustomerpricefull": 2.100,
  "sumcustomerpricetax": 0.100,
  "vatid": 4,
  "vatlevel": 5
},
{
  "code": "10",
  "costpricebase": 1.500,
  "costpricefull": 1.845,
  "costpricetax": 0.345,
  "customerpricebase": 2.276,
  "customerpricefull": 2.800,
  "customerpricetax": 0.524,
  "hour": "13.14",
  "name": "Plzen 12° .5",
  "note": "èochtan",
```



```
"orderprinterid1": 0,
"orderprinterid2": 0,
"plu": 5,
"qty": 1,
"receiptdivisor": 2,
"receiptplu": 1,
"referenceid": "",
"selected": false,
"sktype": "",
"sktypezerovat": "",
"stock": 1,
"sumcustomerpricebase": 2.276,
"sumcustomerpricefull": 2.800,
"sumcustomerpricetax": 0.524,
"vatid": 1,
"vatlevel": 23
},
{
  "code": "010",
  "costpricebase": 0.909,
  "costpricefull": 1.118,
  "costpricetax": 0.209,
  "customerpricebase": 1.626,
  "customerpricefull": 2,
  "customerpricetax": 0.374,
  "hour": "13.14",
  "name": "Plzen 12° .3",
  "note": "po èiarku",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 5,
  "qty": 1,
  "receiptdivisor": 3.3,
  "receiptplu": 2,
  "referenceid": "",
  "selected": false,
  "sktype": "",
  "sktypezerovat": "",
  "stock": 1,
  "sumcustomerpricebase": 1.626,
```

```
"sumcustomerpricefull": 2,
"sumcustomerpricetax": 0.374,
"vatid": 1,
"vatlevel": 23
},
{
  "code": "00099",
  "costpricebase": 0.150,
  "costpricefull": 0.150,
  "costpricetax": 0,
  "customerpricebase": 0.150,
  "customerpricefull": 0.150,
  "customerpricetax": 0,
  "hour": "13.14",
  "name": "Komodita test 0% VratObal",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 3,
  "qty": 1,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "",
  "selected": false,
  "sktype": "",
  "sktypezerovat": "VO",
  "stock": 1,
  "sumcustomerpricebase": 0.150,
  "sumcustomerpricefull": 0.150,
  "sumcustomerpricetax": 0,
  "vatid": 3,
  "vatlevel": 0
},
{
  "code": "00099",
  "costpricebase": 0.150,
  "costpricefull": 0.150,
  "costpricetax": 0,
  "customerpricebase": 0.150,
  "customerpricefull": 0.150,
```

```
"customerpricetax": 0,
"hour": "13.15",
"name": "Komodita test 0% VratObal",
"note": "",
"orderprinterid1": 0,
"orderprinterid2": 0,
"plu": 3,
"qty": -2,
"receiptdivisor": 0,
"receiptplu": 0,
"referenceid": "",
"selected": false,
"sktype": "",
"sktypezerovat": "VO",
"stock": 1,
"sumcustomerpricebase": -0.300,
"sumcustomerpricefull": -0.300,
"sumcustomerpricetax": 0,
"vatid": 3,
"vatlevel": 0
},
{
  "code": "9993",
  "costpricebase": 3,
  "costpricefull": 3,
  "costpricetax": 0,
  "customerpricebase": 3,
  "customerpricefull": 3,
  "customerpricetax": 0,
  "hour": "13.15",
  "name": "Voucher 3eur",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 4,
  "qty": -1,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "",
  "selected": false,
```

```
"sktype": "VP",
"sktypezerovat": "N",
"stock": 1,
"sumcustomerpricebase": -3,
"sumcustomerpricefull": -3,
"sumcustomerpricetax": 0,
"vatid": 3,
"vatlevel": 0
},
{
  "code": "sal",
  "costpricebase": 0.100,
  "costpricefull": 0.119,
  "costpricetax": 0.019,
  "customerpricebase": 1,
  "customerpricefull": 1.190,
  "customerpricetax": 0.190,
  "hour": "13.15",
  "name": "Salama",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 8,
  "qty": -0.2300,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "O-1234TEST",
  "selected": false,
  "sktype": "",
  "sktypezerovat": "",
  "stock": 1,
  "sumcustomerpricebase": -0.227,
  "sumcustomerpricefull": -0.270,
  "sumcustomerpricetax": -0.043,
  "vatid": 2,
  "vatlevel": 19
}
]
}
```

vrátenie:

```
{
  "head": {
    "date": "20241223",
    "id": 3,
    "note": "-test",
    "sum": -3.30,
    "user": "1 (hos)"
  },
  "footer": {
    "note": "Thanks for visit"
  },
  "items": [
    {
      "code": "p20",
      "costpricebase": 0.100,
      "costpricefull": 0.120,
      "costpricetax": 0.020,
      "customerpricebase": 1,
      "customerpricefull": 1.200,
      "customerpricetax": 0.200,
      "hour": "17.47",
      "name": "Product 20%",
      "note": "",
      "orderprinterid1": 0,
      "orderprinterid2": 0,
      "plu": 10,
      "qty": -1,
      "receiptdivisor": 0,
      "receiptplu": 0,
      "referenceid": "O-3004E4A90359408084E4A90359E080E2",
      "selected": false,
      "sktype": "",
      "sktypezerovat": "",
      "stock": 1,
      "sumcustomerpricebase": -1,
      "sumcustomerpricefull": -1.200,
      "sumcustomerpricetax": -0.200,
      "vaid": 1,
    }
  ]
}
```

```
"vatlevel": 20
},
{
  "code": "p10",
  "costpricebase": 0.100,
  "costpricefull": 0.110,
  "costpricetax": 0.010,
  "customerpricebase": 1,
  "customerpricefull": 1.100,
  "customerpricetax": 0.100,
  "hour": "17.47",
  "name": "Product 10%",
  "note": "",
  "orderprinterid1": 0,
  "orderprinterid2": 0,
  "plu": 11,
  "qty": -1,
  "receiptdivisor": 0,
  "receiptplu": 0,
  "referenceid": "O-3004E4A90359408084E4A90359E080E2",
  "selected": false,
  "sktype": "",
  "sktypezerovat": "",
  "stock": 1,
  "sumcustomerpricebase": -1,
  "sumcustomerpricefull": -1.100,
  "sumcustomerpricetax": -0.100,
  "vaid": 2,
  "vatlevel": 10
},
{
  "code": "00",
  "costpricebase": 0.100,
  "costpricefull": 0.100,
  "costpricetax": 0,
  "customerpricebase": 1,
  "customerpricefull": 1,
  "customerpricetax": 0,
  "hour": "17.47",
  "name": "Komodita test 0%",
```

```

        "note": "",
        "orderprinterid1": 0,
        "orderprinterid2": 0,
        "plu": 9,
        "qty": -1,
        "receiptdivisor": 0,
        "receiptplu": 0,
        "referenceid": "O-3004E4A90359408084E4A90359E080E2",
        "selected": false,
        "sktype": "",
        "sktypezerovat": "N",
        "stock": 1,
        "sumcustomerpricebase": -1,
        "sumcustomerpricefull": -1,
        "sumcustomerpricetax": 0,
        "vaid": 3,
        "vatlevel": 0
    }
}
]
}

```

data class pre Kotlin:

```

package com.gecora.gepos.data

/**
 * class for JSON import structure from old project MyECR or others
 * note: camel notation is not there because DBASE doesn't know it
 */
data class ReceiptImport(
    val head: Receipt.Head,
    val footer: Receipt.Footer,
    val items: ArrayList<Receipt.Item>
) {

    class Receipt {

        class Head {
            // (KOSH is from MyECR.FoxProProject2024)
            var id: Int? = null // 2, : id of KOSH
            var date: String? = null // "20241119",

```

```

var user: String? = null // "1 (hos💎)"
var note: String? = null // "my test",
var sum: Double? = null // 5.20,
}

```

```

class Footer { // KOSH
    var note: String? = null // "Thanks for visit"
}

```

```

class Item { // KOSP (is from MyECR.FoxProProject2024)
    var plu: Int? = null // 1, : id of KOMODITY->KOSP
    var name: String? = null // "Komodita test 23%"
    var note: String? = null // "description of my product",
    var sktype: String? = null // "", : product type of tax office (SK region)
    // 1="K"(kladnaPol), 2="VO"(vratObaly), 3="V"(vratena), 4="O"(oprava), 5="Z"(zľava),
6="OZ"(odpočZáloha), 7="VP"(výmPouk/voucher)
    var sktypezerovat: String? = null // "", : the reason why there is zero VAT / type of tax office (SK region)
    // "", : "N"(oslOdDane), "P"(prenesDanPov), "K"(cestKanc), "T"(použTov), "U"(umelDiela),
"Z"(zberatPredm/starož)
    var vaticid: Int? = null // 1, 2, 3, 4 // if 0, be assigned from "vatlevel"
    var vatlevel: Double? = null // 23, 19, 0, 5
    var costpricebase: Double? = null // 0.500
    var costpricetax: Double? = null // 0.115
    var costpricefull: Double? = null // 0.615
    var qty: Double? = null // 1
    var customerpricebase: Double? = null // 0.976
    var customerpricetax: Double? = null // 0.224
    var customerpricefull: Double? = null // 1.200
    var sumcustomerpricebase: Double? = null // 1.951
    var sumcustomerpricetax: Double? = null // 0.449
    var sumcustomerpricefull: Double? = null // 2.400
    var hour: String? = null // "14.20"
    var selected: Boolean? = null // false
    var stock: Int? = null // 0 or more
    var receiptplu: Int? = null // 0 or n>0 if tis miniCalculations (parent product have the same id)
    var receiptdivisor: Double? = null // 0 or n>0 if this miniCalculations then: 1 / n
    var orderprinterid1: Int? = null // 0 // slip printer 1 ID
    var orderprinterid2: Int? = null // 0 // slip printer 2 ID
    var code: String? = null // "00020" // barcode or other
    var referenceid: String? = null // uuid from exist ticket if this is refund
}

```



```
}  
}  
}
```

Java class:

```
package com.gecora.gepos.structures.gePosApi;  
  
import java.util.ArrayList;  
  
/**  
 * class for JSON import structure from old project MyECR or others  
 * note: camel notation is not there because DBASE doesn't know it  
 */  
public class MyECR {  
  
    public static class Receipt {  
  
        public Head head;  
        public Footer footer;  
        public ArrayList<Item> items;  
  
        public static class Head { // (KOSH is from MyECR.FoxProProject2024)  
            public Integer id; // 2, : id of KOSH  
            public String date; // "20241119",  
            public String user; // "1 (hos💎)"  
            public String note; // "my test",  
            public Double sum; // 5.20,  
        }  
  
        public static class Footer { // KOSH  
            public String note; // "Thanks for visit"  
        }  
  
        public static class Item { // KOSP (is from MyECR.FoxProProject2024)  
            public Integer plu; // 1, : id of KOMODITY->KOSP  
            public String name; // "Komodita test 23%"  
            public String note; // "description of my product",  
            public String sktype; // "", : product type of tax office (SK region)  
            // 1="K"(kladnaPol), 2="VO"(vratObaly), 3="V"(vratena), 4="O"(oprava), 5="Z"(zľava),
```

6="OZ"(odpočZáloha), 7="VP"(výmPouk/voucher)

public String sktypezerovat; // "", : the reason why there is zero VAT / type of tax office (SK region)

// "", : "N"(oslOdDane), "P"(prenesDanPov), "K"(cestKanc), "T"(použTov), "U"(umelDiela),

"Z"(zberatPredm/starož)

public Integer vatic; // 1, 2, 3, 4 // if 0, be assigned from "vatlevel"

public Double vatlevel; // 23, 19, 0, 5

public Double costpricebase; // 0.500

public Double costpricetax; // 0.115

public Double costpricefull; // 0.615

public Double qty; // 1

public Double customerpricebase; // 0.976

public Double customerpricetax; // 0.224

public Double customerpricefull; // 1.200

public Double sumcustomerpricebase; // 1.951

public Double sumcustomerpricetax; // 0.449

public Double sumcustomerpricefull; // 2.400

public String hour; // "14.20"

public Boolean selected; // false

public Integer stock; // 0 or more

public Integer receiptplu; // 0 or n>0 if tis miniCalculations (parent product have the same id)

public Double receiptdivisor; // 0 or n>0 if this miniCalculations then: 1 / n

public Integer orderprinterid1; // 0 // slip printer 1 ID

public Integer orderprinterid2; // 0 // slip printer 2 ID

public String code; // "00020" // barcode or other

public String referenceid; // uuid from exist ticket if this is refund

}

}

}