1. **API set**
   1. **Driver login**

Method: **PUT**

Path: **/logindriver**

Query params: none

Req. body: **“phone”** – user telephone number in full international format

**“email”** – user email

**“imei”** – an IMEI of the current POS terminal

**“pwd”** – Base62(SHA256(pin))

(either “phone” or “email” may be presented in the request at a time)

Resp. body: **LoginAnswer** model JSON

Errors: **400 Bad request** – in case of any inconsistency in the request body

(missing required params, both “email” and “phone” in the request)

* 1. **Find new shift for today**

This API is called by the application to find an available Shift for current date. The server returns the Shift which is in state “new”. There can be only one Shift in “new” state for a single date. If there’s a Shift with in “accepted” state for current date, this API returns no shift. (A case with both Shifts in “new” and “accepted” state cannot happen according to the backend logic). This API return also current Server time, so the application can verify its local clock.

Method: **GET**

Path: **/todaynewshift**

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **FindNewShiftResponse** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

* 1. **Get all goods with prices**

This API return an array of all good categories with actual prices for the current moment.

Method: **GET**

Path: **/pricedgoods**

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: JSON array of **PricedGood** models

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

* 1. **Accept Shift**

This API accepts a Shift by a Driver. It can only be applied to Shifts in state “new”. Server must validate the User who calls this API.

Method: **PUT**

Path: **/shift/{id}/accept**

Path params: {id} – ID of the requested Shift

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **AcceptShiftResponse** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Shift cannot be accepted

**404 Not found** – in case of the Shift does not exist

* 1. **Reject Shift**

This API rejects a Shift by a Driver. It can only be applied to Shifts in state “new”. Server must validate the User who calls this API.

Method: **PUT**

Path: **/shift/{id}/reject**

Path params: {id} – ID of the requested Shift

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **ShiftEntity** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Shift cannot be rejected

**404 Not found** – in case of the Shift does not exist

* 1. **Get Shift by ID**

Method: **GET**

Path: **/shift/{id}**

Path params: {id} – ID of the requested Shift

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **ShiftEntity** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of this is not my shift

**404 Not found** – in case of the Shift does not exist

* 1. **Close Shift**

This API closes a Shift by a Driver. No Visit can be changed after the Shift has been closed. If **“close\_time”** parameter is presented, it is used as the close timestamp. If **“close\_time”** parameter is NOT presented, server uses its local time at the API call moment as the close timestamp. The server must validate current Shift state (only **“accepted”** is allowed) and the caller Driver is assigned for the Shift.

Method: **PUT**

Path: **/shift/{id}/close**

Path params: {id} – ID of the requested Shift

Query: **“close\_time”** – optional DateTime for closing Shift (see format in 3.1)

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **ShiftEntity** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Shift cannot be closed

**404 Not found** – in case of the Shift does not exist

* 1. **Create Note**

This API creates Note. Server must validate that the Shift for creating note is really for a requester Driver. Server must validate that the Shift for creating note is in **“accepted”** state.

Method: **POST (multipart)**

Path: **/note**

Query: **“creation\_time”** – optional DateTime of this Note creation (see format in 3.1). If provided, the backend must use this value instead of its local timestamp for Note creation.

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Request body part for data (single):

Part name: “note”

Part value: single **NotePostBody** model JSON

Request body parts for images (multiple):

Part name: “image”

Part value: binary data

Resp. body: Single **NoteEntity** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Note cannot be created (see validation)

* 1. **Start Visit**

This API starts the requested Visit by changing its state from **“new”** to **“in\_progress”**. Server must validate Visit’s state (must be **“new”**). Server must validate parent Shift’s state (must be **“accepted”**)

Method: **PUT**

Path: **/visit/{id}/start**

Path params: {id} – ID of the requested Visit

Query: **“delayed\_start\_time”** – optional DateTime to start Vis (see format in 3.1)

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **VisitState** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Visit cannot be started

**404 Not found** – in case of the Visit does not exist

* 1. **Fail Visit**

This API fails the requested Visit by changing its state from **“in\_progress”** to **“failed”**. Server must validate Visit’s state (must be **“in\_progress”**). Server must validate parent Shift’s state (must be **“accepted”**)

Method: **PUT**

Path: **/visit/{id}/fail**

Path params: {id} – ID of the requested Visit

Query: **“delayed\_fail\_time”** – optional DateTime to fail Visit (see format in 3.1)

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Post body: Single **VisitFailPostBody** model JSON

Resp. body: Single **VisitState** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Visit cannot be failed

**404 Not found** – in case of the Visit does not exist

* 1. **Complete Visit**

This API completes the requested Visit by changing its state from **“in\_progress”** to **“completed”** and saved Collected Goods for this visit. Server must validate Visit’s state (must be **“in\_progress”**). Server must validate parent Shift’s state (must be **“accepted”**). Server must set the Visit completion time either from the **“delayed\_complete\_time”** parameter or for from its local current time. Server must calculate total payed amount for all collected Goods. Server must instantiate **“order”** property for the **“CollectedGoods”** table from the body items order

Method: **PUT**

Path: **/visit/{id}/complete**

Path params: {id} – ID of the requested Visit

Query: **“delayed\_complete\_time”** – optional DateTime to complete Vis (see format in 3.1)

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Req. body: Single **VisitResult** model JSON

Resp. body: Single **VisitResponse** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

**403 Forbidden** – in case of the Visit cannot be started

**404 Not found** – in case of the Visit does not exist

* 1. **Get my profile**

Method: **GET**

Path: **/users/me**

Headers: **“User”** – current logged-in User ID

**“Authorization”** – bearer authentication with “access\_token”

Resp. body: Single **UserEntity** model JSON

Errors: **401 Unauthorized** – in case of no authorization or authorization expired

* 1. efguief

1. **API Entities**
   1. **LoginAnswer**

This entity is returned in response of the Login API and contains: result of login, error message, access tokens and a logged it User entity

|  |
| --- |
| Successful login JSON |
| {  "success": true,  "access\_token": "fgkrht94gt94ugrfwefh982g3fg40f",  "expires\_in": 259200,  "refresh\_token": "UYfhaoyh308fhweoifh08hf08whf0hwf0hw",  "user": { <***2.2 UserEntity***> }  } |

|  |
| --- |
| Failed login JSON |
| {  "success": false,  "msg\_error": "Wrong PIN"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| success | Boolean | + | Shows if login was successful.  **false** – the “msg\_error” is included  **true** - "access\_token", "expires\_in", "refresh\_token" and "user" are uncluded |
| msg\_error |  |  | Shows a reason of login failure |
| access\_token | Text |  | The if the Bearer authentication token |
| expires\_in | Integer |  | A lifetime of the Bearer authentication "access\_token" in seconds |
| refresh\_token | Text |  | A token to refresh "access\_token" |
| "user" | UserEntity |  |  |

* 1. **UserEntity**

|  |
| --- |
| JSON |
| {  "user\_id": "6b01ef7e-9bbe-4cab-8efb-820aad98d0b7",  "registration\_date": "1019-06-18",  "first\_name": "Craig",  "surname": "Ihezie",  "photo\_url": "https://instagram.fhrk3-1.fna.fbcdn.net/vp/8916842e96883ade5758b09c77efcf3b/5E2F1413/t51.2885-15/e35/s1080x1080/70211421\_497014251098841\_4023846826284037064\_n.jpg?\_nc\_ht=instagram.fhrk3-1.fna.fbcdn.net&\_nc\_cat=109",  "phone\_number": "+380501234587",  "email": "test@mail.com",  "role": "driver"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| "user\_id" | Text | + | ID of a User |
| "registration\_date" | Text/Date | + | Date of the registration in the system in format **“yyyy-mm-dd”** |
| "first\_name" | Text | + | First name of a user |
| "surname" | Text | + | Family name of a user |
| "photo\_url" | Text/URL |  |  |
| "phone\_number" | Text | + | This property is used as login. Users are tied to devices’ telephone numbers |
| "email" | Text/Email | + | This email is used for PIN recovery |
| "role" | Text/Enum | + | Available values: **“driver”**, **“admin”** |

* 1. **FindNewShiftResponse**

|  |
| --- |
| JSON |
| {  "server\_time": "2016-08-15T23:47:12.123+001",  "shift": { <***2.4 ShiftEntity***> }  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| "server\_time" | Text/DateTime | + | Server time in the ISO8601 format. The example is 2001-07-04T12:08:56.235-0700.  Java SimpleDateFormat template is "yyyy-MM-dd'T'HH:mm:ss.SSSZ" |
| "shift" | ShiftEntity |  | A Shift entity for today in the “new” state, If found |

* 1. **ShiftEntity**

|  |
| --- |
| JSON |
| {  "shift\_id": 482,  "driver\_id": "6b01ef7e-9bbe-4cab-8efb-820aad98d0b7",  "execution\_date": "2001-05-12",  "scheduled\_time\_start": "08:00:00",  "scheduled\_duration": "06:30:00",  "num\_visits": 12,  "num\_visits\_completed": 8,  "num\_visits\_failed": 1,  "actual\_time\_start": "2001-05-12'T'07:20:47.245+001",  "actual\_time\_closed": null,  "state": "accepted"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| "shift\_id" | Integer | + | ID of the shift – primary key |
| "driver\_id" | Text | + | ID of a user this Shift is for |
| "execution\_date" | Text/Date | + | The date this Shift is scheduled for |
| "scheduled\_time\_start" | Text/Time |  | Optional scheduled start time of the Shift |
| "scheduled\_duration" | Text/Time |  | Optional scheduled duration of the Shift |
| "num\_visits" | Integer | + | Number of visits for the shift (metadata) |
| "num\_visits\_completed" | Integer | + | Number of completed visits for the shift (metadata) |
| "num\_visits\_failed" | Integer | + | Number of failed visits for the shift (metadata) |
| "actual\_time\_start" | Text/DateTime |  | Actual time this Shift was accepted by the Driver. |
| "actual\_time\_closed" | Text/DateTime |  | Actual time this Shift was closed by the Driver. |
| "state" | Text/Enum | + | Allowed values: **"new"**, **"accepted"**, **"closed"**, **"cancelled"** |

* 1. **PricedGood**

This is a composite entity which describes Good category and related actual price

|  |
| --- |
| JSON |
| {  "goodcategory\_id": 3,  "name": "Shoes",  "ewc\_code": "20 01 10",  "unit\_of\_measure": "unit",  "price\_id": 2457,  "price\_value": 312,  "price\_scale": 3,  "price\_time\_set": "2001-05-12'T'07:20:47.245+001"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| goodcategory\_id | Integer | + | ID of the Good category – primary key |
| name | Text | + | Name of the good category |
| ewc\_code | Text | + | Related EWC code |
| unit\_of\_measure | Text | + | Enum with values: **“kg”**, **“unit”** |
| price\_id | Integer | + | ID of the actual price at the moment |
| price\_value | Integer | + | Price value |
| price\_scale | Integer | + | Price scale |
| price\_time\_set | Text/DateTime | + | Time this price was set up. |

* 1. **AcceptShiftResponse**

This composite entity is used as the response of the “Accept Shift” API. It contains updated Shift entity, array of Visits this shift consists of and the set of Sites used in the Visits.

|  |
| --- |
| JSON |
| {  "shift": { <***2.4 ShiftEntity***> },  "visits": [ <***2.8 VisitEntity***> ],  "sites": [ <***2.7> SiteEntity***> ]  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| "shift" | ShiftEntity | + | The requested shift with update **“state”** property |
| "visits" | VisitEntity[] | + | Visits included to the requested Shift, sorted by the **“order”** property |
| "sites" | SiteEntity[] | + | Sites related to the Visits included to the answer |

* 1. **SiteEntity**

This entity is directly backed by the “Sites” table on the backend

|  |
| --- |
| JSON |
| {  "site\_id": 41,  "shop\_name": "Cool shop",  "shop\_code": 327,  "address\_line\_1": "<Some address>",  "address\_line\_2": "<Some address>",  "town": "<town>",  "city": "<city>",  "postcode": "<postcode>",  "latitude": 51.088611,  "longitude": 1.558858,  "uuid": "81a04df8-1281-4bc5-8a98-e9a2970312cc"  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| "site\_id" | Integer | + | ID of the Site |
| "shop\_name" | Text | + | Name of the site |
| "shop\_code" | Integer | + | Code of the site |
| "address\_line\_1" | Text |  | Address of the site |
| "address\_line\_2" | Text |  |  |
| "town" | Text |  |  |
| "city" | Text |  |  |
| "postcode" | Text | + |  |
| "latitude" | Double | + |  |
| "longitude" | Double | + |  |
| "uuid" | Text | + | Some parameter which is used for QR-code validation |

* 1. **VisitEntity**

This entity is directly backed by the “Visits” table on the backend

|  |
| --- |
| JSON |
| {  "visit\_id": 5248,  "shift\_id": 482,  "site\_id": 41,  "order": 1,  "suggest\_time": "14:35:00",  "start\_time": "2016-08-15T13:47:12.123+001"  "complete\_time": "2016-08-15T14:03:12.123+001",  "state": "completed",  "customer\_name": "Sally May",  "customer\_signature":"ZGpoZzc4cnQgM3JnNzlydCA5cWdydXcgZ2R2cWcgeTU0dC0wOC1yIDg5cigmJTY3ZQ==",  "total\_payed\_amount": 120715,  "total\_payed\_scale": 2,  "note\_id": 1845  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **Property** | **type** | **required** | **description** |
| visit\_id | Integer | + | ID of the Visit – primary key |
| shift\_id | Integer | + | ID of the parent Shift |
| site\_id | Integer | + | ID of the Site this Visit is for |
| Order | Integer | + | Optional Order of this visit in route |
| suggest\_time | Text/Time |  | Suggested time of this visit |
| start\_time | Text/DateTime |  | Actual time this visit was started – state was changed from “new” to “in\_progress” |
| complete\_time | Text/DateTime |  | Actual time this visit was completed – state was changed from **“in\_progress”** to **“completed”** |
| State | Text/Enum | + | Visit lifecycle tracking property. Allowed values are: **“new”**, **“in\_progress”**, **“accepted”**, **“failed”** |
| customer\_name | Text |  |  |
| customer\_signature | Text/Base64 |  | Base64 encoded byte array of a customer’s signature |
| total\_payed\_amount | Integer |  | Total payed money for the Visit |
| total\_payed\_scale | Integer |  |  |
| note\_id | Integer |  | This is the ID of optional Note for this Visit. May be NULL if there’s no note. Each Visit can have only 1 note. |

* 1. **NotePostBody**

This entity is used to create Notes

|  |
| --- |
| JSON |
| {  "text": " <Some note text> ",  "shift\_id": 482,  "opt\_emergency\_id": 4,  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| shift\_id | Integer | + | ID of a Shift |
| text | Text | - | Text of the Note |
| opt\_emergency\_id | Integer | - | Optionally-attached Emergence category for this Note |

NOTE: Server must validate that either **“text”** of **“opt\_emergency\_id”** is presented.

* 1. **NoteEntity**

|  |
| --- |
| JSON |
| {  "note\_id": 8726,  "shift\_id": 482,  "visit\_id": 5248,  "number": 56,  "author\_id": "6b01ef7e-9bbe-4cab-8efb-820aad98d0b7",  "time\_created": "2016-08-15T13:47:12.123+001",  "text": " <Some note text> ",  "images": [ <Array of image URLs> ],  "opt\_emergency\_id": 4,  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| note\_id | Integer | + | ID of the note – primary key |
| shift\_id | Integer | + | ID of a Shift |
| visit\_id | Integer | - | Optional ID of a Visit after the requesting Shift |
| number | Integer | + | This is server-generated number of the Note. Auto-increment unique within each Driver. |
| author\_id | Text | + | ID of the Driver –author of this note |
| time\_created | Text/DateTime | + | Creation time of this Note |
| text | Text | + | Text of the Note |
| images | Text[]/URL | + | Array of attached images’ URLs. If no images are attached, the empty array must be provided |
| opt\_emergency\_id | Integer | - | Optionally-attached Emergence category for this Note |

* 1. **VisitState**

This entity is used as a response body in the APIs which changes state of a Visit (start, fail).

|  |
| --- |
| JSON |
| {  "visit\_id": 5248,  "state": "in\_progress",  "start\_time": "2016-08-15T13:47:12.123+001"  "complete\_time": null,  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| visit\_id | Integer | + | ID of a Visit – primary key |
| state | Text/Enum | + | State of a Visit |
| start\_time | Text/DateTime |  | Optional time of this Visit was started |
| complete\_time | Text/DateTime |  | Optional time of this Visit was completed |

* 1. **VisitResponse**

This entity is used as response in the APIs which present full visit with dependencies (Complete Visit, Get Visit by ID)

|  |
| --- |
| JSON |
| {  "visit": {  <***2.8 VisitEntity*** >  },  "goods": [  <Array of ***2.13 CollectedGood***>  ],  "note": {  <The ***2.10 NoteEntity*** or null if no Note for the Visit>  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| visit | VisitEntity | + |  |
| goods | CollectedGood[] | + | An empty array must be provided if there are no collected goods for this entity |
| note | NoteEntity | - | Optional single note for a Visit |

* 1. **CollectedGood**

This composite entity represents an item of the set of collected goods with weight and total amount was payed.

|  |
| --- |
| JSON |
| {  "visit\_id": 5248  "name": "Bric-a-Brac",  "weight": 5,  "total\_amount": 270,  "total\_scale": 2  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| visit\_id | Integer | + | ID of the visit this item is for |
| name | Text | + | Name of the Good Category |
| total\_amount | Integer | + | Total payed amount multiplied by 10^total\_scale |
| total\_scale | Integer | + | Scale for amount value |

* 1. **VisitResult**

This Entity is used as a POST body of the API to complete visit

|  |
| --- |
| JSON |
| {  "visit\_id": 5248  "customer\_name": "Sally May",  "customer\_signature":"ZGpoZzc4cnQgM3JnNzlydCA5cWdydXcgZ2R2cWcgeTU0dC0wOC1yIDg5cigmJTY3ZQ==",  "note": {  "text": " <Text of Note> ",  "images": [  <Array of attached image URLs>  ]  },  "goods": [  {  goodcategory\_id: 3,  "price\_id": 2457,  "weight", 127  }  ]  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| visit\_id | Integer | + | ID of a Visit (for validation and debug purpose) |
| customer\_name | Text | + | Name of the Site’s employee, the Driver receipt goods from |
| customer\_signature | Text | + | Base64 encoded byte array of a customer’s signature |
| note | Object | - | Optional Visit-related Note. Nullable |
| note.text | Text | + | Note text. Required within the “note” object |
| note.images | Text[]/URL | + | Attachment images for completion Note of the Visit. Required within the “note” object. Can be empty. |
| goodcategory\_id | Integer | + | ID of the Good category |
| price\_id | Integer | + | ID of the Price this Good item has been receipt for |
| weight | Integer | + | Weight of the receipt item, KG |

* 1. **VisitFailPostBody**

|  |
| --- |
| JSON |
| {  "visit\_id": 5248  "emergency\_id": 4,  "note\_text": " <Some note text> ",  "images": [  <Array of image URLs>  ],  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Properties description** | | | |
| **property** | **type** | **required** | **description** |
| visit\_id | Integer | + | ID of a Visit (for validation and debug purpose) |
| emergency\_id | Integer | + | Reference to an Emergency type which if the reason of Visit failure |
| note\_text | Text | - | Optional Note text describing the reason of the Visit failure |
| images | Text[]/URL | + | Array of attached images’ URLs. If no images are attached, the empty array must be provided |

* 1. fghfghfghfg

1. **Properties data type formats**
   1. **Text/DateTime**

Time in the ISO8601 format. The example is 2001-07-04T12:08:56.235-0700.

Java SimpleDateFormat template is: "yyyy-MM-dd'T'HH:mm:ss.SSSZ"

* 1. **Text/Date**

Local date in the format like “2018-05-18”. Java SimpleDateFormat template is: "yyyy-MM-dd"

* 1. **Text/Time**

Local time in the format like “08:12:37”. Java SimpleDateFormat template is: "HH:mm:ss"

* 1. Dfhgfh
  2. Fgh
  3. Fg
  4. hfgh