fireArt platform is a Java/Javascript oriented platform for trip/delivery business solutions.

Database backend developer Guide

Database backend (on MySQL database) support to platform objects, that using for data saving and manipulating : tables, triggers, procedures and functions.

tables objects:

- manufacture, for products manufactures information
- product, for products and services, that can be selling in store
- product_type, for product's types (types of products and services, such as for products: electronics, clothes, shirts and for services: car rental, trucks, transportation)
- product_param, for additional features as parameters of products and services
- product_type_part, links for product to product type
- product_param_part, links for product to product parameters
- product_review, for review product by user
- product_circle, for links product to product (as similar)
- delivery, for products shipping by other transportation companies
- delivery_type, for delivery's types (types of transportation companies, such as: international, in country, in city)
- transport, for cars for transportable, such as cars, trucks, other transport as transportation service provides for trip and delivery
- transport_type, transport's types, such as sedan, minivan or truck, car, etc
- transport_type_part, links for transport to transport type
- transport_product_part, links for transport to product (some transport can support some services as products)
- transport_review, for review transport by user (such as after trip)
- order_status, for order status (for values <0 is failed order, for values >0 is order completing)
- pickup_status, for customer's pickup status if they start a trip from other places for one transport
- order_AB, for order, that can have transportation from point A (as pickup or order's location) to point B (as dropoff or destination location)
- order_AB_product_part, links for order to product

- order_AB_product_param_part, links for order's product to product parameter
- order_AB_user_part, links for user to shared order for transportation (trip)
- order_job, for user which job creates for other user (customer creates job for worker)
- store, for store as stores, restaurants, transportation and taxi services, etc
- store_part, for store's offices, departments names, addresses, phones, locations and other informations
- stock, for links store to product (products availability)
- stock_invoice, for incoming product to stock invoice documents
- stock_invoice_part, links stock invoice document to product
- currency, for currencies support, one main currency is active
- language, for multilanguages support, many languages can be active
- color, for transport color
- attr, for languages translation support
- attr_part, links attr to language, for translate user's screen text fields
- discount, for discount policy provide (by discount type, 1-discount in percent, 2-discount in value add, 3-increase in percent 4-increase in value)
- tax, for VAT calculation support, customer and worker fee (with works by personal account for prepaid amount to use the service)
- purchase, for buy products and services by order
- payment, credit card payment (for web hooks of payment provider's signaled for user payment success finished, such as PayPal)
- user, for admin, customer and worker with prepaid amount (personal account), discount and personal data information
- user_review, for review user by user
- prepaid_card, for prepaid card, that can increase user's personal account by prepaid code (increase prepaid amount)
- message, for user's chat
- sensor, for user's smartphones, tablets and others gps device support for locations define
- sensor_circle, for links sensor to sensor (other user define locations of other user until sensor circle is true)
- sensor_group, for long terms sensor unit for define locations
- sensor_group_part, links for sensor to sensor group
- sensor_place, favorites sensor's places
- track, for sensor's location
- track_type, links for track to track type
- track_part, additional information about user's sensor track
- audit, for system events stores
- preferences, for system information
- geocode, for geocode objects information

triggers object :

for automatic save current data on create_data fields (product_circle,order_AB,delivery,stock_invoice,purchase,discount,payment,user,sensor,sensor_circle,sensor_group,track,message,prepaid_card)

procedure objects :

- add_audit, for audit data

functions objects :

for add, update, remove manipulating data for table object (3) functions has been created such as
for store table object data manipulating (for all connectivity users as fa_client MySQL user, have Grant Select from table objects)
add_store, update_store, remove_store (function objects) created.

functions objects (update) created for picture, activity, order_status fields (such as user, transport, product, order_AB table objects)

WebAPI developer Guide

WebApi (on Apache Tomcat Java web server) support to Database backend objects similar, such as, if Database backend objects has store table object, than WebApi has add_store, update_store, remove_store WebApi objects. WebApi objects provides data manipulating with Database backend objects by url with <name> :

https://localhost/service/start?name = <database_table_object_name>&token=<token>&…

Example :

Database backend table object <store> has database function objects <add_store>, <update_store>, <remove_store>

WebApi template files <get_store>, <add_store>, <update_store>, <remove_store>

works with http queries :

HTTP GET
  http://localhost/service/start?name=json/get_store&…
HTTP POST
  http://localhost/servoce/start?name=json/add_store
  http://localhost/servoce/start?name=json/update_store
  http://localhost/servoce/start?name=json/remove_store
with post data as string of parameters (… & param1 = value 1 & param 2 = value 2 …)

Such as Http GET request and JSON respond:

https://taxi.autozvit.com/service/start?name=json/get_tax&token=<token>&tax_id=%25&language=ENGLISH


```
{"results":
  [
    {"tax_id":1,"tax_attr":
      [
        {"attr_part_id":157,"attr_name":"title","attr_value":"VAT"}
      ],
     "tax_name":"value added
tax","tax_value":"20.00","language":"","last_update":"2017-01-25 17:27:06"},
    {"tax_id":2,"tax_attr":
      [
        {"attr_part_id":158,"attr_name":"title","attr_value":"Single tax"}
      ],
     "tax_name":"single
tax","tax_value":"5.00","language":"","last_update":"2017-01-25 17:27:06"},
    {"tax_id":3,"tax_attr":[],"tax_name":"driver
fee","tax_value":"5.00","language":"","last_update":"2017-07-18 12:12:50"},
    {"tax_id":4,"tax_attr":[],"tax_name":"client
fee","tax_value":"5.00","language":"","last_update":"2017-07-18 12:12:50"}
  ],
 "rows":4,"session_id":8016,"status":"SUCCESS"
}
```


https://taxi.autozvit.com/service/start?name=json/get_transport&token=<token>&transport_id=1&&language=ENGLISH


```
{"results":
  [
    {
      "transport_id":1,
```

```
      "user_id":1,
      "sensor_id":1,
      "sensor_active":false,
      "sensor":
      [
        {
          "sensor_id":1,
          "user_id":1,
          "user":
          [
            {
              "user_id"1,
              "user_type"1,
              "discount_code":"",
              "discount”[],
              ”first_name":"George",
              "last_name":"Washington",
              "call_name":"admin",
              "email":"g.washington@gmail.com",
              "phone":"+380001112233",
              "user_rate":0,
              "active":true
            }
          ],
          "sensor_circle":[],
          "sensor_name":"admin",
          "serial_number":"1111",
          "device_name":"iPhone6",
          "phone":"+380933882180",
          "active":false
        }
      ],
      "type_part":
        [
          {
             "type_id":3,
             "type_name":"Taxi signer",
             "description":"With special sign",
             "type_attr":
               [
                 {"attr_part_id":75,"attr_name":"title","attr_value":"Taxi
signer"},
                 {"attr_part_id":84,"attr_name":"about","attr_value":"With
special sign"}
               ]
          }
        ],
      "transport_name":"Daewoo Lanos",
      "transport_color":"WHITE",
      "license_plate":"AA5142KA",
      "transport_rate":4.00,
      "last_update":"2017-07-18 15:37:53"
    }
  ],
  "rows":1,
  "session_id":8015,
  "status":"SUCCESS"
}
```


For payment providers support (PayPal built) merchant’s API key need for work (See templates files : add_paypal_payment, add_stripe_payment).
Payment Provider web hooks or other payment activities need payment table object of Database backend new row object added results.

Java/Javascript developer Guide

Java servlet with open source codes (and Developer Guide) Downloadable from GitHub and BitBucket (see for details Installation Guide document).

Javascript objects to Database backend objects and WebApi similar, such as, if Database backend objects and WebApi has store object, Javascript library (jfa_lib.js) has store objects list and functions for : get, add, update, remove similar object.

Example :

MySQL

```
CREATE FUNCTION `add_store`(
…
  IF v_user='root@localhost' OR v_user='fa_admin@localhost' OR
     (v_user='fa_client@localhost' AND v_user_type=1) THEN

    INSERT INTO stock (product_id,store_id,count)
    VALUES (p_product_id,p_store_id,p_count);

    CALL add_audit('add_stock',CONCAT('success: code=0 username=',p_user,'
product_id=',p_product_id,' store_id=',p_store_id));

  ELSE
    CALL add_audit('add_stock',CONCAT('error: code=-3 username=',p_user));
    RETURN -3;
  END IF;
…
```

Javascript

```
…
    this.addStore=function(ptr,message_ptr,callback,callback_ptr){
      var request_param={};
      request_param=getStore_RP(ptr,request_param);

postData(this.username,this.password,this.token,this.language,url_add_store,requ
est_param,message_ptr,saveRetVal,this.ret_val_list,callback,callback_ptr);
    };
…
```

Android developer Guide

Android application is for target platform 9 (for manifest file android:minSdkVersion="9"), but third other libraries support from higher Android platform version (PayPal, Stipe payment providers).
See more for details for PayPal provider:
https://developer.paypal.com

See more for details for Stripe provider:
https://stripe.com (Developers menu point)

Android mobile app is a part of fireArt platform and can works in (3) user’s mode:

- Trip style (as from point A to point B trip booking style)
- Delivery style (as selecting point for product's delivery style)
- Easy style (as similar Uber booking style)

User screen forms (Activity and Fragments) for Android application is:

- Driver activity for User profile (client/driver)
- Splash Activity for start screen application form
- Start Activity is the main screen forms, that open works fragments screen forms, buttons clicking processing, menu points clicking processing and others
- Verify Mobile Activity for Cognalys phone number verify provider by incoming calls (see more on http://www.cognalys.com)
- Settings Activity for user's application options, such as map provider (Google, OSM), booking style (Trip, Delivery, Easy modes), hostname (WebAPI server) address, payment provider (PayPal, Stripe), distance (for drivers can take orders), etc.

- Cart Fragment for shopping cart in delivery mode
- Dropoff Fragment for user's drop-off location for Easy booking style
- Error Fragment for errors preview
- Find Fragment for finding addresses (pickup address, destination address, delivery address) for Trip and Delivery booking style, Easy and EasyAB booking style with Google Places support
- Image Fragment for image preview after downloading (not used)
- Info Fragment for information preview
- Inquiry Fragment for details info of client's order, that nearest from driver
- Invoice Fragment for ending trip in Easy booking style for client with transport and driver estimation (rating)
- List Fragment for all application's list preview (orders, products, messages with layout xml files for card preview, such as layout_order, layout_product, etc)
- Map2 Fragment for map preview (main fragment for work)
- OrderStatus Fragment for order status preview for all booking style
- Payment Fragment for start payment provider API (PayPal, Stripe), that have Android version limited (more then Android version 9 for current application)
- Pickup Fragment for client gps location definition (with address by google geocode provider) and start Easy booking style
- Register Fragment for user register with phone number verify and inputs full personal user data
- Ride Fragment for client's ride in Easy booking style
- Signin Fragment for user login after mobile application re-installed
- Start Fragment for user's case to register or signin
- Success Fragment for success information preview
- Taximeter Fragment for driver/deliveryman works after take order in Trip and Delivery booking style(fragment_taximeter.xml) and in Easy booking style(fragment_job.xml)
- Tip Fragment for tips in Trip and Delivery booking style
- Trip Fragment for trip estimation with cost, time duration and distance after google geocoding, after A (pickup) and B (destination) trip points client clicked on map.

- PrepaidAccount Dialog (layout_prepaid_account.xml) for input prepaid code (to add personal account of prepaid amount) if have support with prepaid cards.
- SocialNetwork Dialog (layout_social_network.xml) for social networks links (inputs in social_network.xml file)
- Splash Dialog for splash screen as dialog preview
- TripRaiting Dialog (layout_trip_rating.xml) for estimate trip (as transport review) for client

others dialogs using as text messages with buttons for action.

All dialogs, activities and fragments show functions are in Manager.java class Networks requested to the database backend (with Storage.java class connectivity and responds JSON objects parsing) are in Manager.java class

Third part Key API library files:

- cognalys_api.xml for Cognalys phone number verify provider by call (see details on http://cognalys.com Documentation menu point)
- google_maps_api.xml for Google map and geocode support (see details on https://developers.google.com Google API Console menu point)
- paypal_api.xml for PayPal API key for payments support (see details on https://developer.paypal.com Dashboard menu point)
- stripe_api.xml for Striep API key for payments support (https://stripe.com Developers menu point)

OSM map provider support not required any keys.