

Endpoint function: Authenticate user with username and password

Endpoint: /user/authenticate/default (Not final)

python function: sql_related.authenticate_default(database, data)

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```
{
    "authenticate_default":
    {
        "username": <user username (String)>,
        "password": <user password (String)>
    }
}
```

JSON Response Format (successful login):

```
{
    "authenticate_default_response":
    {
        "id": <user database id (int)>,
        "name_first": <user first name (String)>,
        "name_last": <user last name (String)>,
        "sys_username": <user username (String)>,
        "email_google": <user google email (String)>,
        "email_fb": <user facebook email (String)>,
        "phone_number": <user phone number (String / null)>,
        "sys_user_role":
        {
            "id": <user role database id (int)>,
            "name": <user role name (String)>
        }
    }
}
```

JSON Response Format (failed login):

```
{
    "authenticate_default_response": "invalid login credentials"
}
```

Endpoint function: Authenticate user with email via Google / Facebook

Endpoint:

- /user/authenticate/google (google authentication) (Not final)
- /user/authenticate/facebook (facebook authentication) (Not final)

python function: sql_related.authenticate_email(database, data, google)

- database = main database object (db)
- data = JSON data directly from HTTP request
- google = boolean, true if google login, false if fb login

HTTP Request Type: POST

JSON Request Format:

```
{
    "authenticate_email":
    {
        "email": <email>
    }
}
```

JSON Response Format (successful login):

```
{
    "authenticate_email_response":
    {
        "id": <user database id (int)>,
        "name_first": <user first name (String)>,
        "name_last": <user last name (String)>,
        "sys_username": <user username (String)>,
        "email_google": <user google email (String)>,
        "email_fb": <user facebook email (String)>,
        "phone_number": <user phone number (String / null)>,
        "sys_user_role":
        {
```

```

        "id": <user role database id (int)>,
        "name": <user role name (String)>
    }
}

```

JSON Response Format (failed login):

```

{
    "authenticate_email_response": "invalid login credentials"
}

```

Endpoint function: Request all company_products

Endpoint:

- /company_product/request/all (Not final)

python function: sql_related.request_company_product(database)

- database = main database object (db)

HTTP Request Type: GET

JSON Response Format:

```

{
    "request_company_product_response":
    [
        {
            "id": <product database id (int)>,
            "company":
            {
                "id": <company database id (int)>,
                "name": <company name (String)>,
                "zones":
                [
                    {
                        "id": <zone database id (int)>,
                        "name": <zone name (String)>
                    },
                    ... (Additional zones, same format)
                ]
            }
        ]
    ]
}

```

```

    },
    "name": <product name (String)>,
    "price_buy": <product purchase price (Float)>,
    "price_sell": <product sell price (Float)>,
    "units_per_price": <amount of units per price_buy/sell (int)>,
    "price_sell_per_unit": <product sell price per single unit (Float)>,
    "description": <product description (String)>
  },
  ... (Additional products, same format)
]
}

```

Endpoint function: Request all shop_orders that need to be delivered

Endpoint:

- /shop/request/not_delivered (Not final)

python function: sql_related.request_shop_order_not_delivered(database)

- database = main database object (db)

HTTP Request Type: GET

JSON Response Format:

```

{
  "request_shop_order_not_delivered_response":
  [
    {
      "id": <order database id (int)>,
      "shop":
      {
        "id": <shop database id (int)>,
        "name": <shop name (String)>,
        "email": <shop name (String / null)>,
        "phone_number": <shop phone number (String / null)>,
        "category": (may be null)
        {
          "id": <shop category database id (int)>,
          "type": <shop category type (String)>
        }
      },

```

```

    "zones":
    [
        {
            "id": <zone database id (int)>,
            "name": <zone name (String)>
        },
        ... (Additional zones, same format)
    ],
    "street": <shop street (String)>,
    "city": "<shop city (String)>",
    "providence": <shop providence (String)>,
    "zip_4": <shop zip+4 code (String)>
},
"price_due": <order price (Float)>,
"price_paid": <order was paid (Boolean)>,
"date_ordered": <order placed date (String)>,
"date_delivered_projected": <order expected date (String)>,
"date_delivered": null,
"order_taker":
{
    "id": <user database id (int)>,
    "name_first": <user first name (String)>,
    "name_last": <user last name (String)>,
    "sys_username": <user username (String)>,
    "email_google": <user google email (String)>,
    "email_fb": <user facebook email (String)>,
    "phone_number": <user phone number (String / null)>,
    "sys_user_role":
    {
        "id": <user role database id (int)>,
        "name": <user role name (String)>
    }
},
"order_fulfiller": null,
"completed": false
"shop_order_items":
[
    {
        company_product:
        {

```

```

    "id": <product database id (int)>,
    "company":
    {
        "id": <company database id (int)>,
        "name": <company name (String)>,
        "zones":
        [
            {
                "id": <zone database id
                    (int)>,
                "name": <zone name
                    (String)>
            },
            ... (Additional zones, same
                format)
        ]
    },
    "name": <product name (String)>,
    "price_buy": <product purchase price
        (Float)>,
    "price_sell": <product sell price (Float)>,
    "units_per_price": <amount of units
        per price_buy/sell (int)>,
    "price_sell_per_unit": <product sell price per
        single unit (Float)>,
    "description": <product description (String)>
},
quantity_units: <units of company product ordered (int)>
},
... (Additional order items)
]
},
... (Additional shops, same format)
]
}

```

Endpoint function: Request all shops

Endpoint:

- /shop/request/all (Not final)

python function: sql_related.request_shop(database)

- database = main database object (db)

HTTP Request Type: GET

JSON Response Format:

```
{
  "request_shop_response":
  [
    {
      "id": <shop database id (int)>,
      "name": <shop name (String)>,
      "email": <shop name (String / null)>,
      "phone_number": <shop phone number (String / null)>,
      "category": (may be null)
      {
        "id": <shop category database id (int)>,
        "type": <shop category type (String)>
      },
      "zones":
      [
        {
          "id": <zone database id (int)>,
          "name": <zone name (String)>
        },
        ... (Additional zones, same format)
      ],
      "street": <shop street (String)>,
      "city": <shop city (String)>,
      "providence": <shop providence (String)>,
      "zip_4": <shop zip+4 code (String)>
    },
    ... (Additional shops, same format)
  ]
}
```

Endpoint function: Request all zones

Endpoint:

- /zone/request/all (Not final)

python function: sql_related.request_zone(database)

- database = main database object (db)

HTTP Request Type: GET

JSON Response Format:

```
{
    "request_zone_response":
    [
        {
            "id": <zone database id (int)>,
            "name": <zone name (String)>
        },
        ... (Additional zones, same format)
    ]
}
```

Endpoint function: Request all shop_categories

Endpoint:

- /shop_category/request/all (Not final)

python function: sql_related.request_shop_category(database)

- database = main database object (db)

HTTP Request Type: GET

JSON Response Format:

```
{
    "request_shop_category_response":
    [
        {
            "id": <shop category database id (int)>,
            "type": <shop category type (String)>
        },
        ... (Additional shop categories, same format)
    ]
}
```



```
}
```

Endpoint function: Create a new shop

Endpoint: /shop/create (Not final)

python function: sql_related.create_shop(database, data)

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```
{
  "create_shop":
  {
    "name": <shop name (String)>,
    "email": <shop email (String / null)>,
    "phone_number": <shop phone number(String / null)>,
    "category": <shop category id (int)>,
    "zones":
    [
      {
        "id": <zone id (int)>
      },
      ... (Additional zones, same format)
    ],
    "street": <shop street (String)>,
    "city": <shop city (String)>,
    "providence": <shop providence (String)>,
    "zip_4": <shop zip+4 code (String)>
  }
}
```

JSON Response Format:

```
{
  "create_shop_response":
  {
    "id": <shop database id (int)>,
  }
}
```

```

    "name": <shop name (String)>,
    "email": <shop name (String / null)>,
    "phone_number": <shop phone number (String / null)>,
    "category": (may be null)
  {
    "id": <shop category database id (int)>,
    "type": <shop category type (String)>
  },
  "zones":
  [
    {
      "id": <zone database id (int)>,
      "name": <zone name (String)>
    },
    ... (Additional zones, same format)
  ],
  "street": <shop street (String)>,
  "city": "<shop city(String)>",
  "providence": <shop providence (String)>,
  "zip_4": <shop zip+4 code (String)>
}
}

```

Endpoint function: Create a new zone

Endpoint: /zone/create (Not final)

python function: sql_related.create_zone(database, data)

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```

{
  "create_zone":
  {
    "name": <zone name (String)>
  }
}

```

JSON Response Format:

```
{
  "create_zone_response":
  {
    "id": <zone database id (int)>,
    "name": <zone name (String)>
  }
}
```

Endpoint function: Create a shop_category

Endpoint: /shop_category/create (Not final)

python function: sql_related.create_shop_category(database, data)

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```
{
  "create_shop_category":
  {
    "type": <shop category type (String)>
  }
}
```

JSON Response Format:

```
{
  "create_shop_category_response":
  {
    "id": <shop category database id (int)>,
    "type": <shop category type (String)>
  }
}
```

Endpoint function: Create a shop_order

Endpoint: /shop_order/create (Not final)

python function: `sql_related.create_shop_order(database, data)`

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```
{
  "create_shop_order":
  {
    "shop_id": <shop database id (int)>,
    "price_paid": <if price was paid at time of ordering (Boolean)>,
    "order_taker_id": <order taker user database id (int)>,
    "order_items":
    [
      {
        "id": <company product database id (int)>,
        "quantity_units": <quantity of item ordered (int)>
      },
      ... (Additional order items, same format)
    ]
  }
}
```

JSON Response Format:

```
{
  "create_shop_response":
  {
    "id": <order database id (int)>,
    "shop":
    {
      "id": <shop database id (int)>,
      "name": <shop name (String)>,
      "email": <shop name (String / null)>,
      "phone_number": <shop phone number (String / null)>,
      "category": (may be null)
      {
        "id": <shop category database id (int)>,
```

```

        "type": <shop category type (String)>
    },
    "zones":
    [
        {
            "id": <zone database id (int)>,
            "name": <zone name (String)>
        },
        ... (Additional zones, same format)
    ],
    "street": <shop street (String)>,
    "city": "<shop city(String)>",
    "providence": <shop providence (String)>,
    "zip_4": <shop zip+4 code (String)>
},
"price_due": <order price (Float)>,
"price_paid": <order was paid (Boolean)>,
"date_ordered": <order placed date (String)>,
"date_delivered_projected": <order expected date (String)>,
"date_delivered": null,
"order_taker":
{
    "id": <user database id (int)>,
    "name_first": <user first name (String)>,
    "name_last": <user last name (String)>,
    "sys_username": <user username (String)>,
    "email_google": <user google email (String)>,
    "email_fb": <user facebook email (String)>,
    "phone_number": <user phone number (String / null)>,
    "sys_user_role":
    {
        "id": <user role database id (int)>,
        "name": <user role name (String)>
    }
},
"order_fulfiller": null,
"completed": false
"shop_order_items":
[
    {

```

```

company_product:
{
    "id": <product database id (int)>,
    "company":
    {
        "id": <company database id (int)>,
        "name": <company name (String)>,
        "zones":
        [
            {
                "id": <zone database id (int)>,
                "name": <zone name (String)>
            },
            ... (Additional zones, same format)
        ]
    },
    "name": <product name (String)>,
    "price_buy": <product purchase price (Float)>,
    "price_sell": <product sell price (Float)>,
    "units_per_price": <amount of units per price_buy/sell (int)>,
    "price_sell_per_unit": <product sell price per single unit
        (Float)>,
    "description": <product description (String)>
},
quantity_units: <units of company product ordered (int)>
},
... (Additional order items)
]
}
}

```

Endpoint function: Update a shop_order as delivered

Endpoint: /shop_order/update/delivered (Not final)

python function: sql_related.update_shop_order_delivered(database, data)

- database = main database object (db)
- data = JSON data directly from HTTP request

HTTP Request Type: POST

JSON Request Format:

```
{
  "update_shop_order_delivered":
  {
    "shop_order_id": <shop order database id (int)>,
    "order_fulfiller_id": <order fulfiller user database id (int)>
  }
}
```

JSON Response Format:

```
{
  "update_shop_order_delivered_response":
  {
    "request_payment": <request payment for order (Boolean)>
  }
}
```