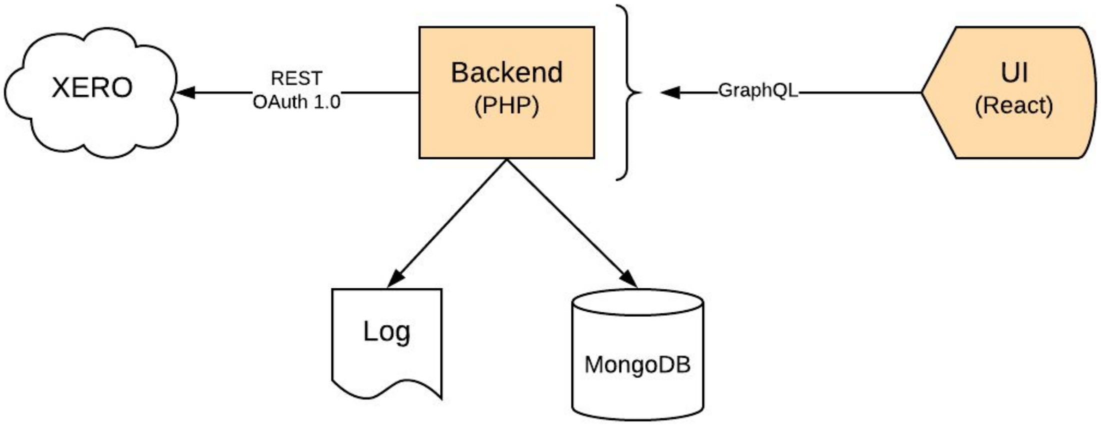
VanHackathon - Beanworks - Jhon Freitez

JJFP Xero-AddOn - Function Description

November, 2019

Solution Architecture

The system has two modules, backend and frontend.



Backend module is responsible of manage the integration with Xero API and database management. This module provides a GraphQL API, which can be used from different applications such-as mobile apps.

Backend module is built in PHP and take advantage of **xero-php** library.

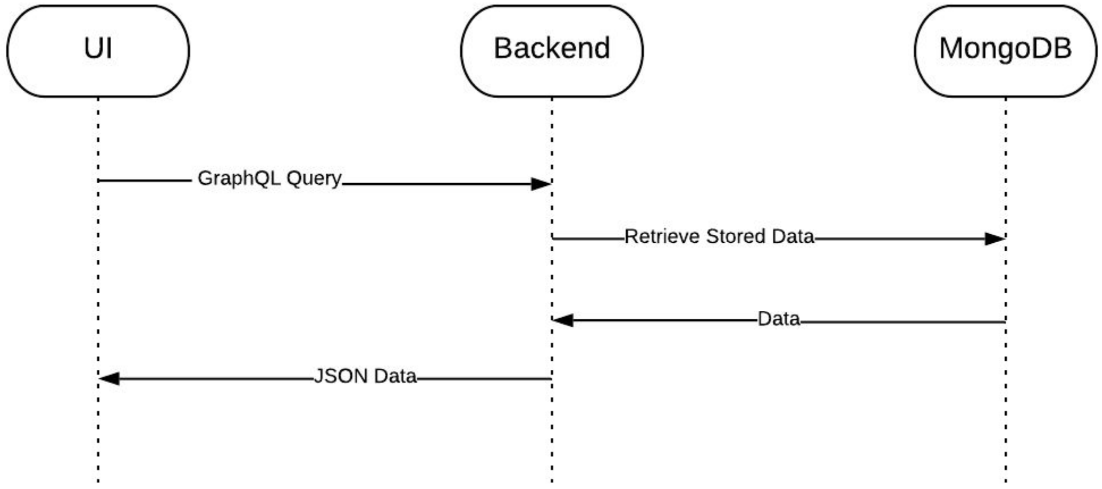
In order to integrate successfully with Xero API, it is necessary to create an Private Application in Xero Developers website.

Frontend module is built with React and Bootstrap and uses Apollo library to connect with GraphQL API.

Main Operations

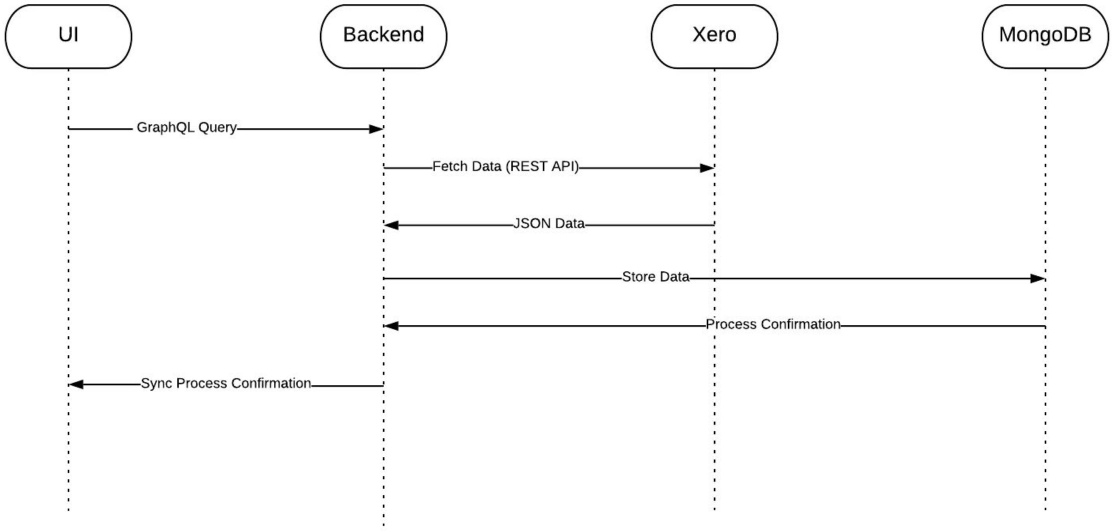
- Query data (Accounts, Vendors, Sync Processes, Sync Processes Log).

  Use to retrieve stored data from and show it in UI. this operations is performed through GraphQL API.



- Perform a Sync Process

  Use to trigger a new Synchronisation Process against Xero.



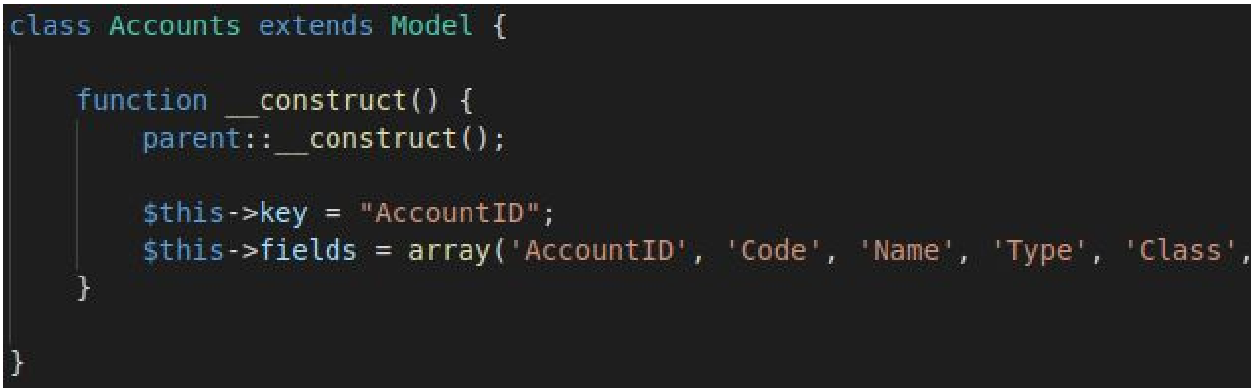
Backend Core Components

Config Manager

Singleton to load app.ini settings.

Model

Manage connection and operations between PHP classes and MongoDB. This is not an abstract class but you have to extends it in order to use it. The child class must have the Mongo Collection name.



```
class Accounts extends Model {

    function __construct() {
        parent::__construct();

        $this->key = "AccountID";
        $this->fields = array('AccountID', 'Code', 'Name', 'Type', 'Class',
    }

}
```

In constructor you have to set two attributes:

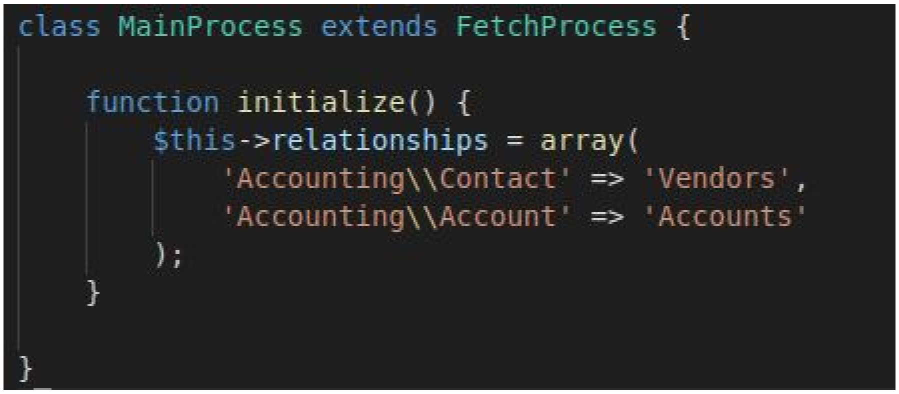
- **key:** With the variable name which contains the Xero ID.
- **fields:** String Array with all variable names to store in MongoDB.

Xero Data Retriever

Uses Xero API to retrieve data.

Sync Fetch Process

Responsible to fetch xero data using **Xero Data Retriever** and store it in MongoDB using **Model classes.** This is an abstract class too, so, you have to extends it.



```
class MainProcess extends FetchProcess {

    function initialize() {
        $this->relationships = array(
            'Accounting\\Contact' => 'Vendors',
            'Accounting\\Account' => 'Accounts'
        );
    }

}
```

You have to override the **initialize** method and just to initialise the **relationship** attribute.

This attribute contains all relationships between **Xero API** and **MongoDB Collection**. For every pair, FetchProcess perform all previous described operations.