**Hướng dẫn sử dụng QSDK – Laravel**

1. Config hệ thống

* Các thiết lập hệ thống giống như Laravel đều nằm trong thư mục config.
* Nếu sử dụng QSDK template project thì không cần config gì. Copy là chạy. Tạm thời do thư viện QSDK bị encrypt bằng Zend Guard nên máy phát triển phải cài Zend Loader mới chạy được. Khuyến khích sử dụng PHP 5.6.10.

1. Định nghĩa widget và dữ liệu

* QSDK là hệ thống widget base có nghĩa là dữ liệu hay bảng dữ liệu được coi là 1 widget. Mỗi widget cho phép nhập, xoá, sửa và hiển thị dữ liệu của bảng và có thể đi kèm với nó là các mối quan hệ 1..N, N…M hoặc N…1.
* Mỗi widget gồm 1 controller, 1 entity, 1 repository, ít nhất một transformer, 1 file config và 1 file route. File config xác định kiểu phân quyền của widget.
* Widget của QSDK đã được implement sẵn phần phân quyền dữ liệu, định dạng dữ liệu, validate dữ liệu. Do đó, khi extend các lớp của QSDK dù là widgetbase hay treebase thì các nội dung trên đã được hỗ trợ sẵn sàng.

1. Gửi dữ liệu login

username:admin@qsoftvietnam.com

password:[password]

grant\_type:password

client\_id:1

client\_secret:123

1. Dữ liệu đầu ra:

* Dữ liệu OK sẽ có định dạng JSON

{

"data": [],

"success": true,

"message": "Done",

"total": 2,

"count": 2,

"per\_page": 20,

"current\_page": 1,

"total\_pages": 1,

"links": []

}

1. Tạo mới module

* Tạo mới module của QSDK sử dụng bộ gen code tại <http://192.168.1.188:86/generator/public> sau đó copy toàn bộ thư mục đã gen trong laravel\_module vào trong thư mục App/V1 và chạy composer dump-autoload
* Copy file migration và seed trong thư mục database của module đã được gen bỏ vào thư mục database tương ứng và chạy php artisan migrate --seed.

1. Thiết lập phân quyền

* Quyền được lưu trong bảng dữ liệu đã được serialize và mã hoá bằng hash.
* Để load lên phân quyền của 1 role gửi: GET:roles/{roleId}/permissions
* Để lấy danh sách các routes hợp lệ trong hệ thống gửi: GET: widget/routes
* Để lấy danh sách các routes hợp lệ của 1 widget gửi: GET: widget/{id}/routes
* Để lấy danh sách các fields của 1 widget gửi: GET: widget/{id}/fields
* Để cập nhật thay đổi quyền của 1 role gửi: PUT: roles/{roleId}/permissions/{WidgetName}
  + Có thể gửi cập nhật từng Permission theo định dạng JSON {routes: [], widget: ‘Customers’} 🡪 để cập nhật cho roleId, widget customer các routes cho phép
  + Có thể gửi cập nhật nhiều Permission cùng lúc theo định dạng JSON [{"routes},"widget":"Cities"],{"routes},"widget":"Customers"]]
  + Các trường dữ liệu có thể cập nhật đối với các widget có check quyền đến mức trường và bản ghi gồm:
    - routes: Array các routes ID cho phép truy cập
    - viewableFields: Array các trường cho phép view
    - viewableRoles: Array các role khác mà role này có thể view
    - viewableByAccounts: Array các account khác mà role này có thể view
    - viewableExceptAccounts: Array các account role này bị cấm view
    - viewableConditions: Array các điều kiện view dữ liệu.
      * 1 Điều kiện: ["where", "roles.name", "ilike", "a"]
      * Nhiều điều kiện: [["where","roles.name","ilike","a"],["whereNotNull","roles.description"]]
      * Nhiều nhóm điều kiện: [["where",[["where","roles.name","ilike","a"],["whereNotNull","roles.description"]]],["where",["where","roles.created\_at",">=","2016-05-01"]]]
    - viewableMaxRecord: Số lượng bản ghi được view tối đa. 0 là view toàn bộ.
    - editableFields
    - editableRoles
    - editableByAccounts
    - editableExceptAccounts
    - editableConditions
    - editableMaxRecord
    - deletableFields
    - deletableRoles
    - deletableByAccounts
    - deletableExceptAccounts
    - deletableConditions
    - deletableMaxRecord
    - exportableFields
    - exportableRoles
    - exportableByAccounts
    - exportableExceptAccounts
    - exportableConditions
    - exportableMaxRecord

1. Tài liệu API: http://192.168.1.188:86/qsdk-laravel/api/