**REQUEST & GIẢI THUẬT PHÂN TRANG (PAGINATION)**

A – LÝ THUYẾT

I – REQUEST

**1. Truyền tham số tĩnh**

* Cú pháp truyền thông qua Router

|  |
| --- |
| /admin/product/:prd\_id |

* Cú pháp nhận

|  |
| --- |
| var prd\_id = req.params.prd\_id |

**2. Truyền tham số động**

* Cú pháp truyền thông qua Router

|  |
| --- |
| /admin/product/\* |

* Cú pháp nhận

|  |
| --- |
| var prd\_id = req.params[key] |

**3. Truyền tham số dạng Query String**

* Cú pháp truyền thông qua Router

|  |
| --- |
| /admin/product?param1=a&param2=b |

* Cú pháp nhận

|  |
| --- |
| var param1 = req.query.param1;  var param2 = req.query.param2 |

II – GIẢI THUẬT PHÂN TRANG (PAGINATION)

Phân trang được chia làm hai phần chính, một là gải thuật phân trang hiển thị dữ liệu, hai là giải thuật tạo ra thanh phân trang

**1. Giải thuật phân trang dữ liệu**

Phân trang dữ liệu chính là cách thức chúng ta lấy ra và hiển thị một khúc dữ liệu bất kỳ trong danh sách tất cả dữ liệu trả về, mà để lấy ra một khúc dữ liệu bất kỳ ta sử dụng cú pháp truy vấn dữ liệu sau trong MongoDB

**Cú pháp:**

|  |
| --- |
| db.CollectionName.find().skip(perRow).limit(rowsPerPage) |

**Trong đó:**

* perRow: Lấy từ Document nào
* rowsPerPage: Số lượng Document muốn lấy tính từ vị trí perRow

Tuy nhiên với công thức trên thì tham số rowsPerPage là tham số do chính lập trình viên quyết định (ví dụ như muốn hiển thị 5 Document hay 10 Document,… trên mỗi trang), do đó bài toán quy về chỉ phải tính toán tham số perRow.

Điều tuyệt vời là công thức tính perRow đã được chứng minh cực kỳ đúng đắn qua công thức chuẩn sau mà chúng ta chỉ cần sử dụng, không cần phải chứng minh lại

**Công thức:**

|  |
| --- |
| perRow = page \* rowsPerPage - rowsPerPage |

**Trong đó:**

* page: Là một tham số mới xuất hiện, nó chính là số thứ tự của trang hiện tại mà người dùng đang muốn xem tất cả các Document trên trang đó (ví dụ: người dùng muốn bấm vào xem nội dung trang 5 thì page = 5)

Cuối cùng page lại được tính theo một công thức hết sức đơn giản sau

**Công thức:**

|  |
| --- |
| if(req.query.param){    page = req.query.param  }  else{    page = 1  } |

**Trong đó:**

* req.query.param: Lấy ra giá trị khi người dùng bấm vào số trang được truyền lên thanh địa chỉ của trình duyệt theo phương thức GET
* page = 1: Là trường hợp khi không có một tham số trang nào được truyền lên thanh địa chỉ của trình duyệt, tức là khi người dùng truy cập vào trang lần đầu tiên (tức là hiển thị trang đầu và như vậy thì page = 1)

**Code full của giải thuật phân trang cho sản phẩm:**

|  |
| --- |
| let rowsPerPage = 5  if(req.query.page){  page = parseInt(req.query.page)  }  else{  page = 1  }  let perRow = page\*rowsPerPage - rowsPerPage  let products = await ProductModel.find().skip(perRow).limit(rowsPerPage).populate("categories")  products = JSON.parse(JSON.stringify(products))  res.render("admin/product", {data:{products:products}}); |

**2. Giải thuật tạo ra thanh phân trang**

Thanh phân trang chính là hiển thị tất cả số trang có được từ tổng số Document trong Collection và tham số rowsPerPage (tổng số Document hiển thị trên mỗi trang)

Như vậy để hiển thị danh sách các trang chúng ta sẽ có các bước tính toán như sau:

**Code JS:**

|  |
| --- |
| let totalRows = db.CollectionName.find()  let totalPages = Math.ceil(totalRows/rowPerPage) |

**Trong đó:**

* totalRows: Tổng số Document trong một Collection
* totalPages: Tổng số trang tính toán được (là một số nguyên dương > 0)
* Math: Đối tượng toán học trong Javacript
* Ceil: Phương thức của đối tượng toán học Math luôn làm tròn lên với tất cả mọi giá trị truyền vào

B – TRAINING DỰ ÁN

I – CÔNG VIỆC TRÊN LỚP

1. Hoàn thiện trang hiển thị danh sách sản phẩm với phân trang sản phẩm

II – CÔNG VIỆC VỀ NHÀ

1. Hoàn chỉnh trang hiển thị danh sách danh mục sản phẩm với phân trang danh mục

2. Hoàn chỉnh trang hiển thị danh sách thành viên với phân trang thành viên