**Xây dựng các chức năng thêm – xóa – sửa sử dụng công cụ**

**service builder**

Người viết: Nguyễn Thị Bích Thủy

**Mục Lục**

[1. Giới thiệu công cụ service builder 2](#_Toc294173203)

[2. Thiết kế các đối tượng và ánh xạ xuống cơ sở dữ liệu (file service.xml) 4](#_Toc294173204)

[3. Xây dựng các phương thức cần thiết trong class {Entity}LocalServiceImpl 6](#_Toc294173205)

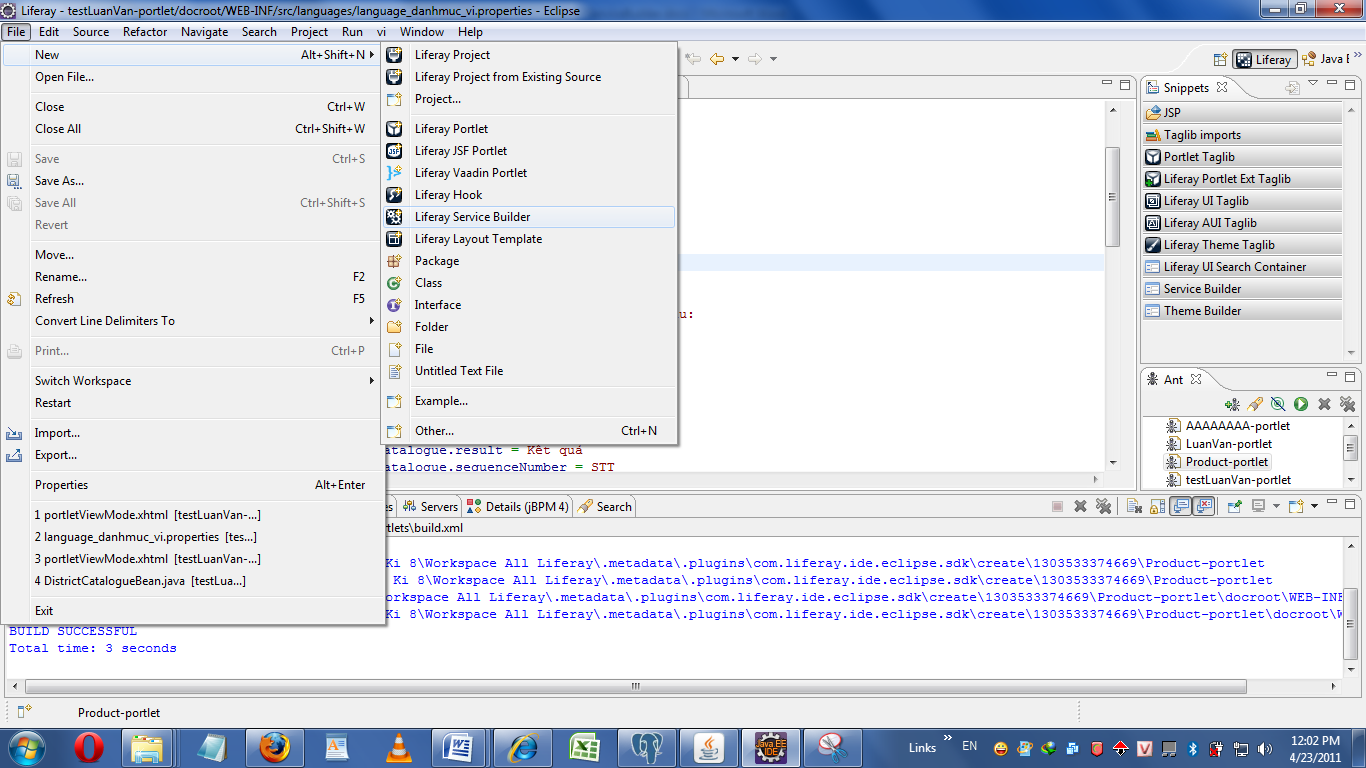
[4. Tạo class {Entity}Finder để xây dựng thêm các phương thức truy vấn CSDL nâng cao 7](#_Toc294173206)

1. Giới thiệu công cụ service builder

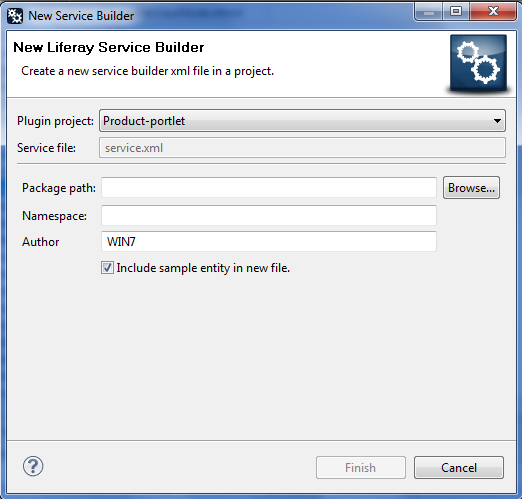
Service builder là một công cụ được xây dựng bởi liferay để tự động hóa việc tạo ra các Interface và Class được sử dụng cho portal hay portlet. Bao gồm code của EJBs, Spring, Persistence, and Model.

Để tạo service builder cho portlet ta thực hiện các bước sau:

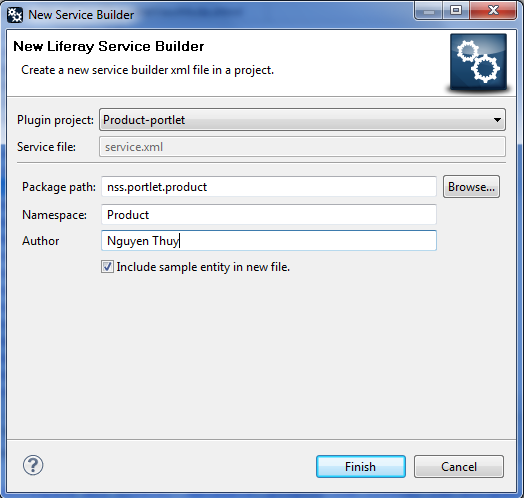
* Bước 1: Chọn File 🡪 New 🡪 Liferay Service Builder.



Giao diện tạo service builder

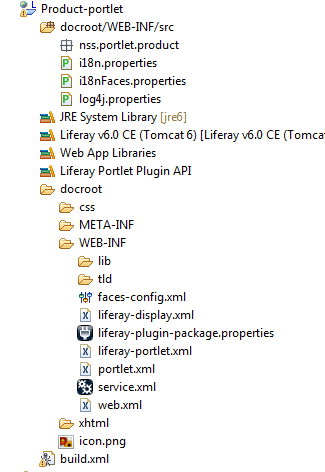


* Plugin project: tên portlet để tạo service.
* Service file: tên file service builder mặc định là service.xml
* Package path: tên package chứa tất cả class và interface trong quá trình tạo build service.
* Namespace: tên của element, để phân biệt 2 nguồn dữ liệu khác nhau trong cùng một file xml
* Author: tên người tạo ra service.
* Bước 2: Nhập đầy đủ thông tin để tạo service.



* Bước 3: Chọn Finish để tạo service.

Cấu trúc portlet sau khi tạo service builder.



1. Thiết kế các đối tượng và ánh xạ xuống cơ sở dữ liệu (file service.xml)

Khai báo package, namespace, author

Bạn có thể khai báo package, namespace, author trong các bước tạo service builder hoặc sau khi tạo service builder vào file service.xml để chỉnh sửa trong các thẻ sau:

|  |
| --- |
| <service-builder package-path=*"nss.portlet.product"*>  <author>Nguyen Thuy</author>  <namespace>Product</namespace> |

Khai báo entity, local-service, remote-service

Cách khai báo entity, local-service, remote-service trong file service.xml

|  |
| --- |
| <entity name=*"Product"* local-service=*"true"*  remote-service=*"true"*> |

* Entity: là thẻ xml được dùng để định nghĩa table trong cơ sở dữ liệu (CSDL).
* Local-service: dịch vụ được sử dụng ở local bỏ qua bước kiểm tra permission.
* Remote-service: dịch vụ được truy xuất từ xa phải thông qua permission.

Khai báo các column

Để khai báo column của table trong service.xml ta dùng thẻ

|  |
| --- |
| <column name=*"productId"* type=*"long"* primary=*"true"* />  <column name=*"productName"* type=*"String"* />  <column name=*"productCode"* type=*"String"* />  <column name=*"description"* type=*"String"* />  <column name=*"price"* type=*"double"* /> |

* Name: tên của column.
* Type: kiểu dữ liệu cho column
* Primary: khai báo khóa chính cho table (primary=”true”).

Khai báo quan hệ giữa các entity

Để khai báo quan hệ giữa các entity ta khai báo như sau:

|  |
| --- |
| <column name=*"categories"* type=*"Collection"* entity=*"Category"*  mapping-table=*"Product\_Category"*/> |

* Mapping-table: khai báo tên table được tạo ra khi table này có quan hệ với table khác.
* Entity: tên table có quan hệ với entity đang tạo.

Khai báo order

Thẻ oder được dùng để sắp xếp giá trị của column theo thứ tự tăng dần hoặc giảm dần.

|  |
| --- |
| <order>  <order-column name=*"productName"* order-by=*"asc"*/>  </order> |

* name : tên column cần sắp xếp.
* order-by: sắp xếp theo tăng dần (asc) hay giảm dần (dsc)

Khai báo các phương thức finder

Thẻ finder được dùng để tìm kiếm dữ liệu từ column.

|  |
| --- |
| <finder return-type=*"Collection"* name=*"ProductName"*>  <finder-column name=*"productName"*/>  </finder> |

* Return-type: kiểu dữ liệu trả về.
* Name: têm column cần tìm dữ liệu.

Khai báo các reference entity

Thẻ reference được dùng để cho phép truy cập đến service của một service.xml khác

|  |
| --- |
| <reference package-path=*"com.liferay.portlet.blogs"* entity=*"BlogsEntry"* /> |

* Package-path: đường dẫn đến package-path của service khác.
* Entity: tên của table cần tham khảo trong service khác.

Ví dụ mẫu file service.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE service-builder PUBLIC "-//Liferay//DTD Service Builder 6.0.0//EN" "http://www.liferay.com/dtd/liferay-service-builder\_6\_0\_0.dtd">

<service-builder package-path=*"nss.portlet.product"*>

<author>Nguyen Thuy</author>

<namespace>Product</namespace>

<entity name=*"Product"* local-service=*"true"* remote-service=*"true"*>

<!-- PK fields -->

<column name=*"productId"* type=*"long"* primary=*"true"* />

<column name=*"productName"* type=*"String"* />

<column name=*"productCode"* type=*"String"* />

<column name=*"description"* type=*"String"* />

<column name=*"price"* type=*"double"* />

<column name=*"categories"* type=*"Collection"* entity=*"Category"* mapping-table=*"Product\_Category"*/>

<order>

<order-column name=*"productName"* order-by=*"asc"*/>

</order>

<finder return-type=*"Collection"* name=*"ProductName"*>

<finder-column name=*"productName"*/>

</finder>

<reference package-path=*"com.liferay.portlet.blogs"* entity=*"BlogsEntry"* />

</entity>

<entity name=*"Category"* local-service=*"true"* remote-service=*"true"*>

<column name=*"categoryId"* type=*"long"* primary=*"true"*/>

<column name=*"categoryName"* type=*"String"*/>

<column name=*"description"* type=*"String"*/>

<column name=*"products"* type=*"Collection"* entity=*"Product"* mapping-table=*"Product\_Category"* />

<order>

<order-column name=*"categoryName"* order-by=*"asc"*/>

</order>

<finder return-type=*"Collection"* name=*"CategoryName"*>

<finder-column name=*"categoryName"*/>

</finder>

</entity>

</service-builder>

1. Xây dựng các phương thức cần thiết trong class {Entity}LocalServiceImpl

* Các phương thức cần thiết trong class LocalServiceImpl

|  |
| --- |
| **public** Product addProduct(String productName, String productCode,  String description, **double** price) **throws** SystemException {  Product product = productPersistence.create(CounterLocalServiceUtil  .*increment*(Product.**class**.getName()));  product.setProductName(productName);  product.setProductCode(productCode);  product.setDescription(description);  product.setPrice(price);  **return** productPersistence.update(product, **false**);  }  **public** Product updateProduct(**long** productId, String productName,  String productCode, String description, **double** price)  **throws** SystemException, PortalException {  Product product = productPersistence.findByPrimaryKey(productId);  product.setProductName(productName);  product.setProductCode(productCode);  product.setDescription(description);  product.setPrice(price);  **return** productPersistence.update(product, **false**);  }  **public** **void** deleteProduct(**long** productId) **throws** NoSuchProductException,  SystemException {  productPersistence.remove(productId);  }  **public** List<Product> getProducts(**int** start, **int** end)  **throws** SystemException {  **return** **super**.getProducts(start, end);  }  **public** List<Product> getProducts(**long** categoryId) **throws** SystemException {  **return** categoryPersistence.getProducts(categoryId);  }  **public** List<Product> getProducts(String productName) **throws** SystemException {  **return** productPersistence.findByProductName(productName);  } |

* Sau khi khai báo cài đặt các phương thức, cần phải build lại service.

1. Tạo class {Entity}Finder để xây dựng thêm các phương thức truy vấn CSDL nâng cao

Các bước để cấu hình custom-sql và hiện thực

* Bước 1: Tạo thư mực custom-sql trong **\docroot\WEB-INF\src**
* Bước 2: Tạo file **default.xml** trong thư mục **custom-sql**, nội dung như sau:

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <custom-sql>  <sql file=*"custom-sql/product.xml"* />  </custom-sql> |

* Bước 3: Tạo file **product.xml** trong cùng thư mục **custom-sql,** nội dung như sau:

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"utf-8"*?>  <custom-sql>  <sql id=*"findBooks"*>  <![CDATA[  SELECT  \*  FROM  Product\_Product  WHERE  (Product\_Product.productName like ?)  ]]>  </sql>  </custom-sql> |

* Bước 4: Tạo file **ProductFinderImpl.java trong thư mục ..\service\persistence có nội dung như sau:**

|  |
| --- |
| **package** nss.portlet.product.service.persistence;  **import** com.liferay.portal.service.persistence.impl.BasePersistenceImpl;  **public** **class** ProductFinderImpl **extends** BasePersistenceImpl  **implements** ProductFinder {  } |

**Build lại service.xml**

* Bước 5: Cài đặt lớp **ProductFinderImpl.java**

|  |
| --- |
| **public** **class** ProductFinderImpl **extends** BasePersistenceImpl **implements**  ProductFinder {  **public** **static** String *FIND\_PRODUCTS* = "findProducts";  **public** List<Product> findProducts(String productName)  **throws** SystemException {  Session session = **null**;  **try** {  session = openSession();  String sql = CustomSQLUtil.*get*(*FIND\_PRODUCTS*);  SQLQuery query = session.createSQLQuery(sql);  query.addEntity("Product", ProductImpl.**class**);  QueryPos qPos = QueryPos.*getInstance*(query);  qPos.add(productName);  **return** (List) query.list();  } **catch** (Exception e) {}  **return** **null**;  }  } |

* **Bước 6: Cập nhật code trong ProductLocalServiceImpl.java thêm vào phương thức** findProducts(String productName)

|  |
| --- |
| **public** List<Product> findProducts(String productName) **throws**  PortalException, SystemException {  **return** ProductFinderUtil.*findProducts*("%" + productName  + "%");  } |

Build lại **service.xml**