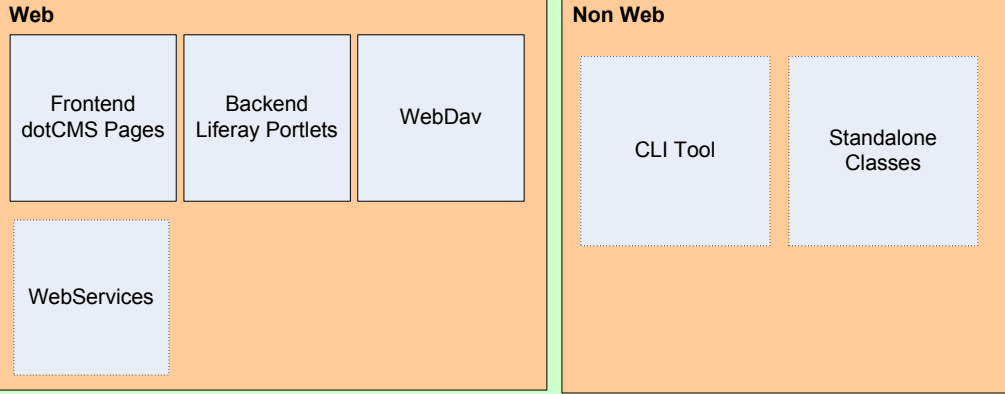


UI/Presentation & Application Layers



Notes

- The business layer is replicated for every entity we have in the cms, E.G.
com.dotmarketing.portlets.category.business
com.dotmarketing.portlets.contentlet.business
- Every Business Layer talks to each other through the APIs, so let say the CategoryFactory needs to retrieve permissions it should do it through the PermissionAPI.

DotCMS Core

Business Layer

(Implemented in the same package named *business* except for the Beans/POJO that should still live in the model package)

Model Layer

POJO/Beans

(Preserve encapsulation - No public attributes, use getters and setters)

com.dotmarketing.portlets.XXX.model
com.dotmarekting.cms.XXX.model

API Layer

API Locator

Used by all class to get an instance any API

API Interaces

All classes will program to the API interface using the API locator to inject the concrete API at runtime.

APIs

(Only instance method should live in these classes and they should inherit from GenericAPI)

com.dotmarketing.portlets.XXX.business
com.dotmarekting.cms.XXX.business

Data Access Layer

Only APIs call the Data Access Layer

Factory Locator

Used by APIs to get an instance of a factory

Factory Interaces

All APIs will program to the factories interface using the factory locator to inject the concrete factory at runtime.

(Methods should only be called by API)

com.dotmarketing.portlets.XXX.business
com.dotmarekting.cms.XXX.business

Cache Locator

Used by factories to get an instance of a cache

Cache Interaces

All factories will program to the cache's interface using the cache locator to inject the concrete cache at runtime.

Caches

(Hidden by the factories, the factories are responsible to maintain caches up to date, the caches cannot call the factories and the APIs should be the only thing that calls the caches)

com.dotmarketing.portlets.XXX.business
com.dotmarekting.cms.XXX.business

Infrastructure Layer

Persistance Services

DotHibernate, HibernateUtil, DotConnect, LuceneUtil

Transactional Services

DotHibernate, HibernateUtil

Mailing Services

Mailer

Logging Services

Logger

Config Services

Config

Liferay Services

PublicUserFactory,
PublicGroupFactory,
PublicRoleFactory,
PublicAddressFactory,
PublicCompanyFactory



DotCMS DB



Lucene Index



External DBs



XML/Text Files



Other Data Sources

Frontend MVC Model

UI Presentation Layer / Struts View

DotCMS Pages

Velocity Code in Templates, Contentlets, Containers, VTL Files, Static VTL Files, JSPs

<<access>>

<<use>>

<<access>>

<<use>>

<<access>>

Application Layer struts-cms.xml

Struts Controllers

Mapped to /dotCMS/XXX
Struts Actions
com.dotmarketing.cms.XXX.action

Struts Model

Struts Forms
com.dotmarketing
.cms.XXX.struts

AJAX Controllers

DWR Objects
com.dotmarketing.cms.XXX.ajax

AJAX Model
Hashmaps

Velocity ViewTools

Toolbox.xml
Standard Velocity Tools like \$date
dotCMS Specific viewtools like
\$categories
com.dotmarketing.viewtools

<<access/use>>

<<access/use>>

<<access/use>>

DotCMS Core (Hiding any component that is not visible to the UI/Presentation and Application Layer)

API Layer

All classes will program to the API's interface and use the API locator to inject the concrete API at runtime.